Proceedings of the Sixth Annual Conference of the Cognitive Science Society

June 28-30, 1984
Boulder, Colorado

Sponsored by the Institute of Cognitive Science and the University of Colorado, Boulder

TABLE OF CONTENTS

SYMPOSIA:

INVITED ADDRESSES:

SUBMITTED PAPERS:

INVITED ADDRESS: COGNITIVE PRINCIPLES IN THE DESIGN OF COMPUTER TUTORS

John R. Anderson, Carnegie-Mellon University

COGNITIVE PRINCIPLES IN THE DESIGN OF COMPUTER TUTORS

John R. Anderson
C. Franklin Boyle
Robert Farrell
Brian Reiser
Carnegie-Mellon University

This paper will identify and justify a set of principles derived from ACT (Anderson, 1983) for designing intelligent computer tutors (Sleeman & Brown, 1982). In doing this we will be drawing on our studies of high school students learning geometry and college students learning to program in LISP. We have observed four students spend approximately 30 hours studying beginning geometry and three students similarly spending 30 hours learning LISP. We recorded these sessions and have analyzed them to varying degrees. Some of these analyses have been reported in a series of prior publications (Anderson, 1981; Anderson, 1982; Anderson, 1983a; Anderson, Farrell, & Sauers, 1984; Anderson, Pirolli, & Farrell, in press). This data base has served as a rich source of information about the acquisition of problem-solving skill and has heavily influenced our design of computer tutors. We have used this data base to develop tutors both the LISP and geometry. These tutors are described elsewhere (Boyle & Anderson, 1984; Farrell, Anderson, & Reiser, 1984).

Principle 1: Identify the Goal Structure of the Problem Space

According to the ACT theory, and indeed most cognitive theories of problem-solving, the problem solving behavior is organized around a hierarchical representation of the current goals. It is important that this goal structure be communicated to the student and instruction be cast in terms of the goal structure. It is not communicated in typical instruction in courses like geometry.

Proofs in geometry are almost universally in a two-column form. It is basically a linear structure of pairs where each pair is a statement and justification. Typical instruction encourages the belief that the goal structure of the student should mimic this linear structure--that at any point in the proof the student will have generated an initial part of the structure and the current goal is to generate the next line of the structure.

There are two serious flaws with using linear proofs as goal structures. First this practice denies the validity of problem-solving search. It encourages the idea that the correct next line should be obvious, but finding the next line often involves considerable planning and search. Students engage in search but feel bad about themselves because they do. Second, search in such a linear structure is doomed to be hopelessly unguided. If the only constraint is to generate a legal line, the search space for the correct proof is hopelessly large.

We have observed students flail at solving geometry problems because they try to work within this linear goal structure. We have evidence that successful students represent proofs to themselves as hierarchical structures of implications that start with the givens of a problem and end

in the conclusion to be proven. It needs to be emphasized that conventional instruction does not communicate this structure and students hardly find it obvious. This deficit is particularly grievous because the successful student's goal structure is much more closely related to this hierarchical proof structure than it is to the linear structure of a two-column proof. Basically, the successful student engages in a forward search from the givens and a backward search from the to-be-proven statement.

Principle 2: Provide Instruction in the Problem-Solving Context

Students appear to learn information better if that information is presented during problem solving rather than during instruction that is apart from the problem-solving context. There are a number of reasons why this should be so:

First, there is evidence that memories are associated to the features of the context in which they were learned. The probability of retrieving the memories is increased when the context of recall matches the context of study (Tulving, 1983; Tulving and Thomson, 1973). An extreme example of this was shown by Ross (1984) who found that secretaries were more likely to remember a text-editor command learned in the context of a recipe if they were currently editing another recipe.

Second, it is often difficult to properly encode and understand information presented outside of a problem context and so its applicability might not be recognized in a problem context. For instance, students may not realize that a top-level variable is really the same thing as a function argument even though they are obliquely told so. As another example, many students reading the side-angle-side postulate may not know what included angle means and so misapply that postulate.

Third, even if a student can recall the information and apply it correctly, they are often faced with many potentially applicable pieces of information and do not know which one to use. We have frequently observed students painfully trying dozens of theorems and postulates in geometry before finding the right one. The basic problem is that knowledge is taught in the abstract and the student must learn the goals to which that knowledge is applicable. If the knowledge is presented in a problem-solving context its goal-relevance is much more apparent.

Principle 3: Provide Immediate Feedback on Errors

Novices make errors both in selecting wrong solution paths and in incorrectly applying the rules of the domain. Errors are an inevitable part of learning, but the cost of these errors to the learner is often higher than is necessary. They can severely add to the amount of time required for learning. More than half of our subjects' problem-solving sessions were actually spent exploring wrong paths or recovering from erroneous steps. Relatively little is learned while students are trying to get out of the holes they have dug for themselves.

In addition, errors often confuse the picture and make it difficult to determine which steps were right or wrong. The classic example of this is the student who finally stumbles onto the correct code but does not

understand why it works. Students often progress in this trial and error mode with respect to LISP evaluation: they don't know when an element will be treated as a function, a variable, or a literal but play around with parentheses and quotes until they get something to work. It is particularly difficult to learn from errors when the feedback on the errors comes at a delay. We (Lewis & Anderson, submitted) have shown that subjects learn more slowly in a problem-solving situation where they are allowed to go down erroneous paths and are only given feedback at delay. Also, the importance of immediate feedback has been well documented in other learning situations (Bilodeau, 1969; Skinner, 1958).

Another cost of errors is the demoralization of the student. In these problem-solving domains errors can be very frequent and frustrating. We believe that much of the negative attitudes and math phobias derive from the bitter experiences of students with errors.

Principle 4: Minimize Working Memory Load

Solving problems often requires holding a great deal of requisite information in a mental working memory. If some of that requisite information is lost there will be errors. It surprised us to find in our LISP protocols that most of the student errors appear to be due to working memory failures. A frequent and disastrous type of error is "losing a level of complexity". One way this manifests itself is that subjects lose track of one level of parentheses. Another way this occurs is when subjects plan to use function1 within function2 within function3, but forget the intermediate function and write function3 directly within function1.

A good human tutor can recognize errors of working memory and typically provides quick correction (McKendree, Reiser, and Anderson, 1984). Tutors realize that there is little profit in allowing the student to continue after making such errors. However, human tutors really have no means at their disposal to reduce the working memory load. This is one of the ways we think computer tutors can be an improvement over human tutors--one can externalize much of working memory on the computer screen. This involves keeping partial products and goal structures available in windows.

Principle 5: Represent the Student as a Production Set

All of our work on skill acquisition has modelled students' behavior as being generated by a set of productions. There is a fair amount of evidence for this view of human problem-solving (e.g., Anderson, 1983; Newell & Simon, 1972). It is also the case that numerous other efforts in the domain of intelligent tutoring have represented the to-be-tutored skill as a production set (e.g., Brown and Van Lehn, 1980; O'Shea, 1979; Sleeman, 1982).

Productions in ACT represent the knowledge underlying a problem-solving skill as a set of goal-oriented rules. Some representative examples for LISP and geometry are:

IF the goal is to insert an element into a list
THEN plan to use CONS and set as subgoals
1. To code the element
2. To code the list

IF the goal is to code a function that calculates a relation on a list
THEN try to use CDR-recursion and set as subgoals
1. To code the terminating condition
2. To code the recursive condition

IF the goal is to prove $\langle XYZ \overleftarrow{=} \langle UVW$
and $\overline{XY} \overleftarrow{=} \overline{UV}$
and $\overline{YZ} \overleftarrow{=} \overline{VW}$
THEN plan to use side-angle-side and set as a subgoal
1. To prove $\langle XYZ \overleftarrow{=} \langle UVW$

Such rules not only enable the system to follow student problem-solving but they define an appropriate grain size for instruction. Basically, our tutoring systems monitor whether a student uses each rule correctly and corrects any incorrect or missing rules. As emphasized by Brown and Van Lehn, student misconceptions or bugs can be organized as perturbations of correct rules.

Human tutors seem to intuit an appropriate grain size of rules for instruction but often their intuitions are wrong. This is one place where a system based on careful analysis of student problem-solving may be able to outperform the typical human tutor.

Principle 6: Adjust the Grain Size of Instruction According to Learning Principles

One of the reasons human tutors have difficulty with the grain size for instructing students is that the grain size changes as experience is acquired in the domain. According to the ACT learning theory, this change is produced by a knowledge compilation process that collapses a sequence of productions into larger "macro" production rules. Human tutors, being highly skilled in the domain, exemplify a large grain size in their problem-solving and have a considerable difficulty intuiting the appropriate grain size for the student.

An effective computer tutor will have to adjust the grain size of instruction as the student progresses through the material. Using a theory of production learning it will have to predict when the original productions become compiled into macro productions so that it can change the grain size of instruction.

Principle 7: Enable the Student to Approach the Target Skill by Successive Approximation

Students do not become experts in geometry or LISP programming after solving their first problem. They gradually approximate the expert behavior, accumulating separately the various pieces (production rules) of the skill. It is important that a tutor support this learning by approximation. It is very hard to learn in a tutorial situation that requires that the whole solution be correct. The tutor must accept partially correct solutions and shape the student on those aspects of the solution that are weak.

Generally, it is better to have the early approximations occur in problem contexts that are as similar to the final problem context as

possible. Skills learned in one problem context will only partially transfer to a second context. Students learn features from early problems to guide their problem-solving operators. If these features are different from the final problem space the problem-solving operators will be misguided. For instance, early problems in geometry tend to involve algebraic manipulations of measures. Consequently, the student learns to convert segment and angle congruence into equality. Later problems, such as those involving triangle congruence, do not involve converting congruence of sides and angles into equality of measures.

The advantage of a private tutor is that he/she can help the student through problems which are too difficult for the student to solve entirely alone. Thus, it is common to see a sequence of problems where the tutor will solve most of the first problem with the student just filling in a few of the steps, less of the second, etc. until the student is solving the entire problem.

Principle 8: Promote Use of General Problem-Solving Rules Over Analogy

There are two basic methods that we have observed students using to solve the first problems in a domain. One is to use analogies to earlier problems in the text or problems from other domains to help guide the problem solving. The basic strategy is to try to map the structure of a solution of one problem to another problem. Anderson (1981 tech report), Anderson, Farrell, and Sauers (1984), and Anderson, Pirolli, and Farrell (in press) discuss specific examples from our protocols on geometry and LISP.

The other method is to extract general problem-solving operators from the instruction and apply these to the problem. For instance, if the goal is to prove triangles congruent, one can apply postulates about triangle congruence. If the goal is to create a list structure, one can try to apply a function that creates list structures. The problem with such general operators is that in many domains the search space of the combinations of these operators becomes enormous. This is perhaps why only a little additional information tends to be introduced with each new section of a textbook. The student can restrict search to these new potential operations (cf. Van Lehn, 1983).

Another difficulty with the general problem-solving approach is that it is often difficult to encode the needed problem-solving operators. Often the instruction does not contain explicit statements of such operators. Rather the operators have to inferred from the instruction. Even on those occasions in which the operators are directly stated, students have a hard time understanding them because they are stated so abstractly. Students are often only able to encode the operators correctly when they see them applied to an example problem.

Students appear to prefer analogy as a method of solution in both geometry and LISP. The preference is not overwhelming in geometry and there are many episodes of problem solution by general problem-solving operators. In contrast, the preference is overwhelming in novice LISP programming. In almost every case where a student was writing a first instance of a particular type of LISP function, the student relied on analogy to example LISP functions.

Private human tutors differ as to whether they tend to guide the student to solution by analogy or by general problem-solving operators. We claim that solution with general operators would lead to the best long-term gains. This is because the student often successfully generates a solution by analogy but does not understand why the solution works. We have seen students work their way through problems by analogy and not learn anything of permanent value. What they often learn is how to do analogies. If we take away the problems from which to analogize and they are unable to solve problems. Halasz and Moran (1983) have also commented on the negative consequences of problem solving by analogy. They point out that students are prone to incorrect inferences in using the analogy. An analogy is frequently used in place of a deep understanding of the problem domain.

Conclusions

We have stated a number of cognitive principles that seem important to designing intelligent tutors. Our specific geometry and LISP tutors (Boyle and Anderson, 1984; Farrell, Anderson, and Reiser, 1984) can be consulted for successful application of such rules. To the extent that such applications are successful, they are support not only for these cognitive principles of design, but also for the underlying ACT theory of cognition on which they are based.

References

Anderson, J.R. (1981). Tuning of search of the problem space for geometry proofs. In **Proceedings of IJCAI-81** (pp. 165-170).

Anderson, J.R. (1981). **Acquisition of Cognitive Skill.** ONR Technical Report 81-1, Carnegie-Mellon University, Pittsburgh, PA.

Anderson, J.R. (1982). Acquisition of proof skills in geometry. In J.G. Carbonell, R. Michalski & T. Mitchell (Ed.), **Machine Learning, An Artificial Intelligence Approach.**

Anderson, J.R. (1983). **The Architecture of Cognition.** Cambridge, MA: Harvard University Press.

Anderson, J.R., Farrell, R., & Sauers, R. (1982). **Learning to Plan in LISP.** ONR Technical Report ONR-82-2, Carnegie-Mellon University.

Anderson, J.R., Farrell, R., & Sauers, R. (1984). Learning to program in LISP. **Cognitive Science,** in press.

Anderson, J.R., Pirolli, P., & Farrell, R. Learning recursive programming. In forthcoming book edited by Chi, Farr, & Glaser.

Bilodeau, I. McD. (1969). Information feedback. In E.A. Bilodeau (Ed.), **Principles of Skill Acquisition.** New York: Academic Press.

Boyle, C.F., & Anderson, J.R. Acquisition and automated instruction of geometry proof skills. Paper to be presented at the Annual Meeting of the American Educational Research Association.

Brown, J.S., & Van Lehn, K. (1980). Repair theory: A generative theory of bugs in procedural skills. **Cognitive Science**, 4, 379-426.

Cohen, V.B. (April 20, 1982). Computer software found weak. **New York Times**, C4, Summary of a research.

Farrell, R., Anderson, J., & Reiser, B. Interactive Student Modeling in a Computer-Based LISP Tutor, 1984, submitted to **Cognitive Science.**

Halasz, F., & Moran, T.P. (March 15-17, 1982). **Analogy considered harmful.** Technical Report, Proceedings of the Human Factors in Computer Systems Conference, Gaithersburg, MD.

Lewis, M., & Anderson, J.R. The role of feedback in discriminating problem-solving operators.

McKendree, J., Reiser, B.J., & Anderson, J.R. Tutorial goals and strategies in the instruction of programming skills. Paper submitted to the 1984 conference of the Cognitive Science Society.

Newell, A., & Simon, H. (1972). **Human Problem Solving.** Englewood Cliffs, NJ: Prentice-Hall.

O'Shea, T. (January, 1979). A Self-Proving Quadratic Tutor. **International Journal of Man-Machine Studies**, 11(1), 97-124.

Ross, B.H. (1984). Remindings and their effects in learning a cognitive skill. **Cognitive Psychology**, in press.

Skinner, B.F. (1958). Teaching machines. **Science**, 128, 889-977.

Sleeman, D. (1982). Assessing aspects of competence in basic algebra. In D. Sleeman & J.S. Brown (Eds.), **Intelligent Tutoring Systems**, New York: Academic Press.

Sleeman, D., & Brown J.S. (Eds.). (1982). **Intelligent Tutoring Systems.** New York: Academic Press.

Tulving, E. (1983). **Elements of Episodic Memory.** London: Oxford University Press.

Tulving, E., & Ghomson, P.M. (1973). Encoding specificity and retrieval processes in episodic memory. **Psychological Review**, 80, 352-373.

Van Lehn, K. (1983). **Felicity conditions for human skill acquisition: Validating an AI-based theory.** Technical Report CIS-21, Xerox Parc, Palo Alto, CA.

INVITED ADDRESS: A FRAMEWORK FOR A QUALITATIVE PHYSICS

John Seely Brown, Xerox Parc

A FRAMEWORK FOR A QUALITATIVE PHYSICS

John Seely Brown
Johan de Kleer
Xerox Parc

Change is a ubiquitous characteristic of the physical world. But what is it? What causes it? How can it be described? Thousands of years of investigation have produced a rich and diverse physics which provides many answers. Important concepts and distinctions underlying change in physical systems are state, cause, law, equilibrium, oscillation, momentum, quasistatic approximation, contact force, feedback, etc. Notice that these terms are qualitative and can be intuitively understood. Admittedly they are commonly quantitatively defined. The behavior of a physical system can be described by the exact values of its variables (forces, velocities, positions, pressures, etc.) at each time instant. Such a description, although complete, fails to provide much insight into how the system functions. The insightful concepts and distinctions are usually qualitative, but they are embedded within the much more complex framework established by continuous real-valued variables and differential equations. Our long-term goal is to develop an alternate physics in which these same concepts are derived from a far simpler, but nevertheless formal, qualitative basis.

The motivations for developing a qualitative physics stem from outstanding problems in psychology, education, artificial intelligence, and physics. We want to identify the core knowledge that underlies physical intuition. Humans appear to use a qualitative causal calculus in reasoning about the behavior of their physical environment. Judging from the kinds of explanations humans give, this calculus is quite different from the classical physics taught in classrooms. This raises questions as to what this (naive) physics is like, and how it helps one to reason about the physical world.

In classical physics the crucial distinctions for characterizing physical change are defined within a nonmechanistic framework and thus they are difficult to ground in the common-sense knowledge derived from interaction with the world. Qualitative physics provides an alternate and simpler way of arriving at the same conceptions and distinctions and thus provides a simpler pedagogical basis for educating students about physical mechanisms.

Artificial intelligence and (especially) its subfield of expert systems are producing very sophisticated computer programs capable of solving tasks that require extensive human expertise. A commonly recognized failing of such systems is their extremely narrow range of expertise and their inability to recognize when a problem posed to them is outside this range of expertise. In other words, they have no common-sense. In fact, expert systems usually cannot solve simpler versions of the problems they are designed to solve. The missing common-sense can be supplied, in part, by qualitative reasoning.

A qualitative causal physics provides an alternate way of describing physical phenomena. As compared to modern physics, this qualitative physics

is only at its formative stages and does not have new explanatory value. However, the qualitative physics does suggest some promises of novelty, particularly in its explicit treatment of causality - something modern physics provides no formalism for treating.

Our proposal is to reduce the quantitative precision of the behavioral descriptions but retain the crucial distinctions. Instead of continuous real-valued variables, each variable is described qualitatively - taking on only a small number of values, usually +, -, or 0. Our central modeling tool is the qualitative differential equation, called a confluence. For example, the qualitative behavior of a valve is expressed by the confluence $\partial P + \partial A - \partial Q = 0$ where Q is the flow through the valve, P is the pressure across the valve, A is the area available for flow, and ∂Q, ∂A, and ∂P represent changes in Q, A, and P. The confluence represents multiple competing tendencies: the change in area positively influences flow rate and negatively influences pressure, the change in pressure positively influences flow rate, etc. The same variable can appear in many confluences and thus can be influenced in many different ways. In an overall system each confluence must be satisfied individually. Thus if the area is increasing but the flow remains constant, the pressure must decrease no matter what the other influences on the pressure are. A single confluence often cannot characterize the behavior of a component over its entire operating range. In such cases the range must be divided into subregions, each characterized by a different component state in which different confluences apply. For example, the behavior of the valve when it is completely open is quite different from that when it is completely closed. These two concepts, of confluence and of state, form the basis for a qualitative physics, a physics that maintains most of the important distinctions of the usual physics but is far simpler.

ENVISION

One of our central tenets of our methodology for exploring the ideas and techniques is to construct computer systems based on them and to compare their results with our expectations. The program ENVISION has been run successfully on hundreds of examples of various types of devices (electronic, translational, hydraulic, acoustic, etc.). Although we view constructing working programs as an important methodological strategy for doing research, the existence of a working implementation contributes little to the conceptual coherence of the theory - it is an existence proof at best.

Although the algorithms ENVISION uses are not the primary focus of this talk, a brief description of its inputs, outputs, and success criteria clarifies the stage for the following conceptual presentations. ENVISION's basic task is to derive function from structure for an arbitrary device. It relies on a single library of generic components and uses the same model library to analyze each device. The input to ENVISION is a description of a particular situation in terms of (1) a set of components and their allowable paths of interaction (i.e., the device's topology), (2) the input forces applied to the device (if any), and (3) a set of boundary conditions which constrain the device's behavior. ENVISION produces a description of the behavior of the system in terms of its allowable states, the values of the system's variables, and the direction these variables are changing. Most

importantly, it produces complete causal and logical analyses for that behavior. Both of these analyses provide explanations of how the system behaves with the causal analysis also identifying all possible feedback paths.

The success criteria for ENVISION are also important. Our physics is qualitative and hence sometimes underdetermines the behavior of a system. In these cases ENVISION produces a set of behaviors (we call these interpretations). At a minimum, for a prediction to be correct, one of the interpretations must correspond to the actual behavior of the real device. A stronger criterion follows from observing that a structural description, abstracted qualitatively, of a particular device implicitly characterizes a wide class of different physically realizable devices with the same device topology. The stronger criterion requires that for the predictions produced by envisioning to be correct then (1) the behavior of each device in the class is described by one of the interpretations, and (2) every interpretation describes the behavior of some device of the class.

We present a new unifying framework: the qualitative differential equation. This new research builds on the ideas of qualitative value, component models, and envisioning, developed in our earlier investigations. These concepts and our methodology are discussed in more detail in our earlier papers. Some of the important concepts developed more recently within our qualitative framework are:

Quasistatic approximation: Most modeling, whether quantitative or qualitative, makes the approximation that behavior at some small time scale is unimportant. In modern thermodynamics this concept is central to the definition of equilibrium. Until now qualitative physics has treated this modeling issue in both an ad hoc and tacit manner. In our formulation quasistic assumptions play a theoretically motivated and explicit role.

Causality: The behavior of a device is viewed as arising from the interactions of a set of processors, one for each component of the "device." The information-passing interactions of the individual components are the cause-effect interactions between the device's components. Within this framework causal accounts are defined (as interactions that obey certain meta-constraints) and their limitations explored.

Mythical causality and mythical time: Any set of component models make some assumptions about device behavior (i.e., quasistatic assumptions) and hence cannot, in principle, yield causal accounts for the changes that must occur between equilibrium states of a system. In order to handle this problem we have defined new notions of causality and time (i.e., mythical causality and mythical time) cast in terms of information-passing "negotiations" between processors of neighboring components.

Generalized machines: Many physical situations can be viewed as some kind of generalized machine, whose behavior can be described in terms of variable values. These variables include force, velocity, pressure, flow, current, and voltage.

Proof as explanation: Physical laws, viewed as constraints, are acausal. We discuss how a logical proof of the solution of a set of constraints is a kind of acausal explanation of behavior.

Qualitative calculus: Qualitative physics is based on a qualitative calculus, the qualitative analog to the calculus of Newton and Leibniz. We define qualitative versions of value, continuity, differential, and integral.

Episodes: Episodes are used to quantize time into periods within which the device's behavior is significantly different.

Digital physics: Each component of a physical system can be viewed as a simple information processor. The overall behavior of the device is produced by causal interactions between physically adjacent components. Physical laws can then be viewed as emergent properties of the universal "programs" executed by the processors. A new kind of physical law might thus be expressible as constraints on these programs, processors or the information-flow among them.

Structure to Function: We want to be able to infer the behavior of a physical device from a description of its physical structure. The device consists of physically disjoint parts connected together. Each component has a type, whose generic model (i.e., laws governing its behavior) is available in the model library. The structure of a device is described in terms of its components and interconnections. The task is to determine the behavior of a device given its structure and access to the generic model as specified in the model library.

No-function-in-structure: The goal is to draw inferences about the behavior of the composite device solely from laws governing the behaviors of its parts. This view raises a difficult question: where do the laws and the descriptions of the device being studied come from? Unless we place some conditions on the laws and the descriptions, the inferences that can be made may be (implicitly) pre-encoded in the structural description or the model library.

The no-function-in-structure principle is central: the laws of the parts of the device may not presume the functioning of the whole. Take as a simple example a light switch. The model of a switch that states, "if the switch is off, no current flows; and if the switch is on, current flows," violates the no-function-in-structure principle. Although this model correctly describes the behavior of the switches in our offices, it is false as there are many closed switches through which current does not necessarily flow (such as, two switches in series). Current flows in the switch only if it is closed and there is a potential for current flow. One of the reasons why it is surprisingly difficult to create a "context-free" description of a component is that whenever one thinks of how a component behaves, one must, almost by definition, think of it in some type of supporting context. Thus, the properties of how the component functions in that particular supporting context are apt to influence subtly how one models it.

Locality: The principle of locality demands that the laws for a part cannot specifically refer to any other part. A part can only act on or be acted on by its immediate neighbors and its immediate neighbors must be identifiable a priori in the structure. To an extent, locality follows from no-function-in-structure. If a law for part of type A referred to a specific other part of type B it would be making a presupposition that every

device which contained a part of type A also contained a part of type B. The locality principle also plays a crucial role in our definition of causality.

Class-Wide Assumptions: Those assumptions that are idiosyncratic to a particular device must be distinguished from those that are generic to the entire class of devices. For example, the explanation of the pressure regulator's behavior ignored turbulence at the valve seat, Brownian motion of the fluid molecules, and the compressibility of the fluid; these are however all reasonable assumptions to make for a wide class of hydraulic devices. We call such assumptions class-wide assumptions and they form a kind of universal resolution for the "microscope" being used to study the physical device.

Given this definition for class-wide assumptions, the no-function-in-structure principle can be stated more clearly: the laws for the components of a device of a particular class may not make any other assumptions about the behavior of the particular device that are not made about the class in general.

Although as originally phrased, no-function-in-structure is unachievable, its essential idea is preserved through the use of class-wide assumptions. An presupposition behind no-function-in-structure is that it is possible to describe the laws and the parts of a particular device without making any assumptions about the behavior of interest of the device. There is no neutral, objective, assumption-free way of determining the structure of the device and the laws of its components. The no-function-in-structure demands an infinite regress: a complete set of engineering drawings, a geometrical description, and the positions of each of its molecules, all make some unwarranted assumptions for some behavior that is potentially of interest. Thus, we admit that assumptions in general cannot be avoided in the identification of the parts and their laws, which is why class-wide assumptions are crucial.

Class-wide assumptions play two important roles in our qualitative physics. First, they play a definitional role. Formalizing the idealization (i.e., qualitative physics) demands that we be explicit about which assumptions we are making. Second, and as important, they are important for building expert systems. In constructing an expert system to design, operate or troubleshoot complex devices it is critical to clearly state what assumptions are being used in modeling the given device. Thus, when the unexpected situation or causalty occurs, these assumptions can be examined to determine whether the "knowledge base" can be relied on.

The most common kind of class-wide assumption is that behavior of short enough duration can be ignored. Under this assumption the "settling" behavior by which the device reaches equilibrium after a disturbance need not be modeled. As "short enough" is a relative term, this assumption can be made at many levels. This assumption plays a major role in studying the heating and cooling of gases. In classical physics it is called the quasistatic approximation. For example, the lumped circuit formulation of electronics makes the quasistatic assumption that the dimensions of the physical circuit are small compared to the wavelength associated with the highest frequency of interest. Other examples of class-wide assumptions are

that the mean free path of the fluid particles is small compared to the distances over which the pressure changes appreciably and that the rate of change of the fields is not too large.

A sophisticated reasoning strategy concerns when and how to change the class-wide assumptions when reasoning about a particular device. Such concerns are critical for troubleshooting where faults can force devices into fundamentally new modes of operation. However, even a simple analysis can sometimes require departures from the usual set of class-wide assumptions. For example, it is sometimes important to remove a class-wide assumption for some localized part of the device, such as two wires running close together which should be modeled as a transmission line.

The Importance of the Principles: Violating the no-function-in-structure principle has no direct consequences on the representation and inference schemes presented. Although the form of the structure and the laws are chosen to minimize blatant violations of the no-function-in-structure principle, it is possible to represent and draw inferences from arbitrary laws--in fact it is too easy.

Without this principle our proposed naive physics would be nothing but a proposal for an architecture for building hand-crafted (and thus ad-hoc) theories for specific devices and situations. It would provide no systematic way of ensuring that a particular class of laws did or did not already have built into them the answers to the questions that the laws were intended to answer. That is not to say that the hand-crafted theories are uninteresting--quite the reverse, and the architecture proposed in this talk may well be appropriate for this task. This is especially true for constructing an account of the knowledge of any one individual about the given physical situation. We are doing something quite different; we want to develop a physics--not a psychological account--which is capable of supporting inferences about the world.

Another purpose for the principles is to draw a distinction between the "work": our proposed naive physics does and the "work" that must be done (outside of our naive physics) to identify the parts and laws. Only after making such a distinction is evaluation possible. Without making the distinction, a reader could always ask, in response to some complexity in an example, "Why didn't they model it differently?"; or in response to some clever inference in an example "They built this into their models." As the principles define what can and what cannot be assumed within the models, the criticisms implied by these two questions are invalid. Of course, the principles themselves are open to challenge.

Our Basic Strategic Move

The essence of doing physics is modeling a physical situation, solving the resulting equations and then interpreting the results in physical terms. Modeling a physical situation requires a description of its physical structure. Although there does not exist a general methodology for describing the structure of all physical situations, system dynamics fortunately, provides a methodology for describing a large and interesting collection of physical systems. Thus we initially focus our attention on this class of situations and on how behavior arises from structure. This

move combined with our use of causality as an ontological principle results in a very mechanistic world view. Every physical situation is regarded as some type of physical device or machine made up of individual components, each component contributing to the behavior of the overall device.

INVITED ADDRESS: ASPECTS OF COGNITIVE REPRESENTATION

Fred Dretske, University of Wisconsin

ASPECTS OF COGNITIVE REPRESENTATION

Fred Dretske
Department of Philosophy
University of Wisconsin, Madison

An organism's cognitive system is a control mechanism whose function it is to initiate, adjust and, if necessary, suppress behavior in the service of need and (if applicable) desire satisfaction. In order to perform this function the control mechanism must have direct and continuing access to intelligence about the circumstances in which activity is to be carried out. Hence, the cognitive system is that part of the executive mechanism whose contributions to control are themselves under the control of (or at least sensitive to) whatever information is (or has been) available about the theater in which operations are to be performed. It is an information-driven control system.

Our ordinary attributions of perception, knowledge and belief reflect this general picture of cognitive processes. To say what someone sees, knows or believes is to identify particular control states by means of their representational properties--in terms of the kind of information they are themselves under the control of. Such, at least, is the representational theory of cognitive processes as we find it (or as I find it) embodied in ordinary descriptions and explanations of animal and human behavior. What I want to do here is to explore this way of looking at cognitive systems in order to see how much weight it can bear. That is to say, is the general idea of a representational control system fertile enough to support the enormously rich and variegated attributions of semantic content characteristic of our ordinary descriptions of what we see, know and believe? Can it, furthermore, provide us with a way of understanding how such attributions figure, as they are ordinarily thought to do, in explanations of behavior?

By a representational system I shall mean any system whose function it is to indicate, by its various states, how things stand with respect to some other object, condition or magnitude. This, obviously, is not to require much of a representational system. A variety of simple devices qualify. A tachometer, the sort of instrument found on the dashboard of many automobiles, is a representational system according to this characterization. Its various states indicate something about the rate at which the engine is rotating. Hence, it represents the angular velocity of the crankshaft. A doorbell, in virtue of indicating the condition of the doorbutton (depressed or not), thereby represents the position of the button. And the firing of a neural cell, by indicating the presence and orientation of a certain energy gradient on the surface of a photoreceptor, represents the whereabouts and orientation of an "edge" in the optical input.

In speaking of a representational system I shall continue to speak of the information the system carries about the quantities, conditions and objects it represents. I intend nothing subversive in this way of speaking--nothing, I hope, that begs the questions which it is my project to explore. For by information I mean nothing more, and certainly nothing less, than what the particular states of a representational system indicate to be so.

Thus the tachometer's registration of "1,000 rpm" indicates, and thereby carries the information that, the engine is running at 1,000 revolutions per minute. The ringing doorbell carries the information that the doorbutton is being depressed (and therefore the information that someone is at the door) because, presumably, this is what it indicates. Representational systems and information processing systems are, on this way of thinking, two sides of the same coin. Information is what representational systems need in order to represent, and representation is what information processing systems do for the things about which they carry information.

In thinking of a cognitive system as an information-driven control mechanism, therefore, we are thinking of it in representational terms. This may not be all we have to say about a cognitive system in order to distinguish it from other control mechanisms (e.g., those associated with the autonomic nervous system), but it will do as a start. The project is to see how large an oak we can tease out of this tiny acorn, how far one can go in understanding perception, knowledge and belief with these semantically meager resources. If they prove too meager, it will at least tell us something about the special character of cognitive processes.

In thinking about a representational system, there are at least two questions one can ask about its representational capacity. And when the representational system is, in addition, a control system, there are at least three questions that should be asked. One can ask, first, what it is, what quantity, property, object, person or condition, the system is representing. A thermometer represents the temperature, a fuel gauge the amount of gasoline in the tank, a photograph the objects (persons, building, foliage) that the picture was taken of. Secondly, one can ask how what is represented is represented. What does the representation say about what it represents? That it is 95 (in the case of the thermometer)? That the gas tank is almost empty (fuel gauge)? That your niece has let her hair grow long (photograph)? The first question is a question about the reference or denotation of the representation. The second question is about the content of the representation. Topic and comment.

There are, in other words, pictures of black horses and what Nelson Goodman (Languages of Art, Hackett; Indianapolis, 1976, p. 29) has called black-horse pictures. Unless the picture of a black horse is a black-horse picture, it will not represent the black horse as a black horse. Imagine, for example, a picture of a black horse in which the horse is photographed at a great distance in bad light with the camera slightly out of focus. The horse appears as a blurry spot in the distance. This is a picture of a black horse but not what Goodman calls a black-horse picture. When invited to see pictures of your friend's black horse, you expect to see, not only pictures of a black horse, but black-horse pictures--pictures in which the denotation of the picture is identifiably a black horse, pictures in which the black horse is represented as a black horse.

Similarly, the wolf's internal representation of the sick caribou may or may not be a sick-fleeing-caribou representation. But it certainly is a representation of a sick fleeing caribou. How it represents the animal is, to some degree, a matter of speculation, but unless it has some means of representing defenseless caribou, a way of commenting on these creatures which is, for practical purposes, extensionally equivalent to being a

defenseless caribou, its relentless and unerring pursuit of these particular animals is inexplicable. It would be like trying to explain the behavior of a thermostat in controlling the furnace if it had no means of representing the room temperature as above or below the desired setting. There has to be something in there that "tells" the thermostat what it needs to know for it to carry out its function. The same is true of the wolf.

I rehearse these familiar facts about representations only to emphasize that in thinking about a cognitive system as a representational control mechanism, the same questions can be asked about our cognitive states: what do they represent and how do they represent it? What are they getting information about and what information are they getting? But I said that there were three questions that can be asked about a representational system when it functions as a control mechanism. We can ask, not only about its topic (what is represented) and the comments it makes about that topic (the way it is represented), but about which of these comments, if any, has a control function. Which elements of the representation play a causal role in the determination of behavior? A representation may be ever so rich in the comments it makes about what it represents, but if none of this information is, or can be, used to control and direct movements, it is causally inert, hence, functionally irrelevant. Therefore the representation, qua representation, plays no role in the system's cognitive economy. A black-horse picture of a black-horse can be used to paper over a hole in my wall. In this case, the representationally significant aspects of the picture are irrelevant to the way it is functioning (concealing the hole). I could as well have used a picture of a white cow or a picture of nothing at all. And if our internal representations are to qualify as cognitive, they must (potentially at least) make some contribution to the way that mechanism functions in controlling and directing behavior. They must do so, furthermore, by means of their representationally significant properties. Only by so doing will the classification of cognitive states in terms of their content figure in the explanation of the behavior they produce.

Our ordinary attributions of sensory and cognitive states reflect the kind of distinctions just discussed. And this, to my mind at least, supports the idea that our familiar, folk psychological picture of cognition is a picture of a kind of representational control mechanism. We say, for example, that Clyde can see a black horse (in the distance) without (for various reasons having to do either with the great distance, the camouflage, the lighting, or the fact that he doesn't have his glasses on) its looking like a black horse, without its presenting a black-horse appearance. In describing what Clyde sees, we are describing what his sensory representation is a representation of. We are ignoring the kind of comment his perceptual system is making about that topic in order to specify the topic itself about which a comment is being made.

Thinking of the cognitive system as a representational mechanism, a mechanism no more complex than a simple gauge, gives us, therefore, the resources for understanding, not only the propositional attitudes, those attitudes (like knowledge and belief) that take (in their verbal expression) sentential clauses as complement of the verb, but also those attitudes (like seeing and hearing) that take concrete nominals as objects of the verb. S sees a bush, mistakes it for an animal crouching beside the path, and flees

in panic. The description of what S sees is an expression of what his internal representation is a representation of--in this case a bush. What he sees it as, what he takes it to be, or (under optimal conditions) what he can see that it is, is an expression of the way he represents the bush. In this case he has a crouching-animal representation of a bush--the analogue, I submit, of a white-cow picture of a black horse.

There is then, as I see it, no real question about the validity of our ordinary descriptive apparatus for assigning perceptual-cognitive states to organisms. For in describing a creature as seeing, knowing or believing something, we may be (and, I would urge, certainly are) doing no more than what we are already doing with such simple representational devices as gauges, instruments, and detectors. We are saying what and how things are being represented. The only real question about the representational model (aside from the causal efficacy of these representations--a point I will get to later) is whether our ordinary descriptions of what a creature sees, knows and believes are semantically too rich to be supported by the actual physical representational resources of the organism. Are our ordinary assignments of reference and content to an organism's internal representations compatible with--and, if so, are they realizable in, the actual neural machinery available for generating these representations? There may be no real dispute in assigning the reference my gas tank and the content half full to my gauge's representations; for the actual physical construction of the device, and the laws governing its operation, clearly reveal that this is what the device indicates and what it indicates it about. But in saying that the wolf saw a sick caribou on the edge of the herd, recognized it as sick and, therefore, as easy prey and, because of this, pursued the animal while ignoring the thousands of healthy ones nearby, are we assigning reference and content that exceed the representational resources of the animal we are describing? If not, what about our descriptions of Jimmy as seeing his uncle and recognizing him as the man who promised to fix his bicycle?

INVITED ADDRESS: THE MEANING OF VISION-RELATED TERMS TO A BLIND CHILD

Barbara Landau, Columbia University

Lila Gleitman, University of Pennsylvania

THE MEANING OF VISION-RELATED TERMS TO A BLIND CHILD

Barbara Landau, Columbia University
Lila Gleitman, University of Pennsylvania

To construct a realistic language learning theory, it seems necessary to have anchor information of three kinds: what input the child receives, what kinds of theory of language ("interim grammars") he constructs at various stages during the acquisition process, and what kind of theory of language (adult grammar) he finally constructs. Given these kinds of information it would become possible to make inferences about the initial representations in terms of which the child learns, and about the learning procedures that are involved. For some years, we have focussed attention on the first two of these questions. Our assumption, following Chomsky (1965) and Wexler and Culicover (1980), is that the relevant input to the learner is of two kinds: He requires sample strings of formatives from the language to which he is being exposed; and he requires that these strings be paired with interpretable extralinguistic information about the construal of these speech samples. Accordingly, we have been studying the course of language learning under conditions where that input varies.

In initial studies, we examined the sensitivity of learning to naturally occurring differences in maternal speech (Newport, Gleitman and Gleitman, 1977). The general finding was that the learning process is strikingly insensitive to variations in maternal speech style within the normal range of that variation. We take such findings to be a first indicant that learners are endowed with a skeletal framework for natural language, which allows them to override detailed differences in the speech input.

More recently, we have been studying learning in the presence of pathologies that restrict the input to the child much more severely. We first studied a population of deaf children of hearing parents who were not exposed to sign language, i.e., who were deprived of the normal samples of speech (and signing). These individuals construct idiosyncratic manual communication systems which develop in ways that mirror the normal course and timing of language acquisition (Feldman, Goldin-Meadow, and Gleitman, 1977), again supporting the view that learners of normal mental endowment are equipped with rudimentary schemata for language growth that survive environmental deprivation.

In the work summarized here, we examined what seems to be a symmetrical case of environmental deprivation, one in which the opportunity to receive extralinguistic information about the construal of input strings is severely reduced: the case of language learning in congenitally blind children (Landau, 1980, 1981; Landau and Gleitman, forthcoming). We studied three blind children over the whole course of their language learning (roughly, the period from 18 months through 5 years). Very often, these children do not know what is going on in the invisible scene around them, they exhibit extreme confusion about conversations and their intents, and (up to about age 36 months) often behave in ways that seem so bizarre that they are often called "autistic" (Fraiberg, 1977). Thus the evidence that blind children are experientially deprived in relevant ways is very strong. Our question is whether and how such deprivation affects the learning of language.

FINDINGS

The development of spontaneous speech in three blind children was compared to that for sighted children, using all measures now standard in the developmental psycholinguistic literature. After an initial mild delay, there was no discernible pathology in the rate or internal organization of this speech. Still, it is possible to suppose that the blind child's comprehension is restricted or deformed owing to the experiential deprivations (for this conjecture, see Bloom, 1983). To find out, we conducted an inquiry into how certain words, namely those that seem to encode vision and the visual experience, are understood by a young blind child. These include nouns such as photograph, verbs such as see, and adjectives such as green. These are the cases for which blind children's input circumstances seem to be maximally different from the input circumstances of their sighted peers. We restrict attention here to the blind child's meanings of vision-related verbs such as look, see and show.

It is well known that blind youngsters, including our own subjects, utter such words as early (at about age 30 months) and often as sighted youngsters. We conducted comprehension studies with a blind 3-year old and four blindfolded sighted control subjects of the same age, to discover what these verbs, as compared to haptic verbs (e.g., touch), meant to them. Summarizing the method, the subjects were asked to "look at" or "touch" various objects under various conditions, e.g., the child might be told to "Look behind you," or "Show the doll to mommy" or "Touch the doll" or even "Touch the table but don't look at it."

The outcomes are very clear from a variety of such manipulations. Both blind and sighted children distinguish between touch and look. For example, any of these subjects asked to "Touch the table" will reach toward it and contact it, usually by a tap or a bang on that object. Asked to "Look at the table," the blindfolded sighted subjects turn their covered eyes toward the object, i.e., they orient toward it visually. Even if commanded to "Look at the table with your hands, they apparently have no choice but to respond visually: They again turn their covered eyes toward the object, but this time also move their hands in some way, not a sensible way (e.g., they put their palms together, as if praying). In contrast, the blind subject told to "look" moves her hands toward the object--holding the head immobile--and then explores its surfaces by moving the hands all over them (in contrast with her response to "touch," where she only bangs or taps the object). Moreover, if told to "Touch the table but don't look at it" she bangs the table; then told "Now you can look at it," she proceeds to explore all its surfaces.

Such findings are consistent with the following interpretation. For both the blind and sighted subjects, touch means 'contact.' For both the blind and sighted subjects, look means 'explore by means of the dominant modality to determine objects.' In short, the findings lead to the view that blind and sighted 3 year olds mean something sensible, the same thing, by terms such as look. What differs is their "dominant modality"--the eye for sighted children and the hand for blind children. But despite this difference in the exploratory modality that is used to "achieve looking", all the children have extracted a construal of the term that has to do with their perceptual transactions with the world.

A further set of outcomes for the blind learner is worth noting. She evidently formed two construals for sight-related terms, one usable of herself, but another usable as the terms relate to sighted others. That is, we have evidence that the blind learner by age 36 months understands the following properties of "sighted seeing": (1) that it can be performed at a distance (e.g., an object is placed in the mother's hand in response to "Let mommy touch" or to "Give mommy" but is usually displayed at a distance in response to "Let mommy see" and "Show mommy"); (2) that it requires orientation of the line of sight (e.g., the blind child turns her body appropriately in response to "Let mommy see the back of your pants/front of your shirt" regardless of which way she was initially facing as the command was given); and (3) that it is blocked by a barrier (e.g., the blind child hides an object by placing her body between it and the mother when told to "Make it so mommy can't see").

THE ENVIRONMENT OF LEARNING

To determine the input conditions for these acquisitions, we examined 15 hours of videotaped sessions which consisted of the blind child's mother interacting naturally with the child, in the period (up to 36 months) before the learner was using look and see freely and frequently in conversation. We tested the plausible conjecture that the mother used look and see to the blind learner when an object was in the child's hand or at least within arm's reach, available for manual inspection. This hypothesis proved inadequate by itself to explain the learning, because almost all verbs commonly used to the blind child by her mother were uttered when the child had some target object in hand or near to hand. Thus this first description of the context for learning serves to distinguish the visual verbs from a few others (e.g., get and come) but leaves most of the verbs (e.g., look, play, and put) undifferentiated.

However, we were able to show that the syntactic environments of the common verbs, as used by the mother, provide a potentially rich source for further distinctions. For example, only look and see, among the common verbs, take sentential complements: The mother does say "Let's see if Granny's home" and "Look what Barbara brought" but she never says anything like "Let's put if Granny's home" or "Give what Barbara brought," while give and put, but not see, appear in three-term argument structures ("She gave the book to Mary" but not "She saw the book to Mary.") Such syntactic information, taken together with the information from the situational encodings (object "nearby"), is sufficient to distinguish each of the common verbs in the corpus from each of the others.

DISCUSSION

Based on our findings and these analyses, we put forward a description of verb learning that recruits both situational and linguistic-distributional evidence. We argue that syntactic formats for verbs are restricted by their construals, and thus provide a principled basis for the blind child's acquisition of the meaning of look and see. To take a single example, perception is of events as well as of objects, and thus perceptual verbs as used by the mother appear with sentential complements (e.g., "Let's see what kind of cheese you want" or "Look what I did") while such verbs as give never appear in such environments. If a learner is prepared to seek

and analyze for these distinctions of syntactic type as they correlate with available interpretations of the external world circumstance, a basis for learning seems to exist. We argue that an analysis in these terms has important virtues in understanding the development of verb meanings: For one thing, such a procedure accounts for our findings, while a simple analysis of "what was going on in the outside world" does not. Moreover, if syntactic information is recruited by the learning procedure, the required storage of word-to-event pairings is minimized as a factor in the learning process. Most important, the postulated storage of the verb frames is not merely a temporary prop for learning. It represents a relevant outcome of language learning: Every child must and does learn that one never says "Let's see a cracker to mommy" but can and does say "Let's give a cracker to mommy."

Summarizing, we hold that syntactic supports to language learning are useful because they are informative (each verb's formats are correlated with the semantic descriptions it encodes), because they are stable and categorical rather than probabilistic sources of evidence in the data base, and because their acquisition is required as a part of language learning whether or not they are taken to bear causally on that learning. More generally, the emerging competencies of blind children are another indication, supportive of our findings for normal children exposed to different maternal speech styles and for isolate deaf children, that learners have sufficient internal wherewithal to override differences and defects in the ambiant extralinguistic circumstances, to acquire language normally.

INVITED ADDRESS: WORD RECOGNITION: A PARADIGM CASE FOR COMPUTATIONAL (PSYCHO-) LINGUISTICS

Henry Thompson, University of Edinburgh

Word Recognition:

A Paradigm Case for Computational (Psycho-) Linguistics[1]

Henry Thompson
Department of Artificial Intelligence
and
Program in Cognitive Science
University of Edinburgh

1. Introduction

Existing psycholinguistic models of word recognition are incapable of computational realisation as currently formulated. Existing computational linguistic approaches to lexical access, morphology and spelling correction provide a rich repetoire of computational methods which appear to offer attractive possibilities for addressing this problem, but are not as they stand psychologically plausible. The proposals made in this paper come out of the efforts of an interdisciplinary group in the Programme in Cognitive Science of the School of Epistemics at the University of Edinburgh to reconcile these two facts, and arrive at a psycholinguistic model of word recognition in continuous speech which is explicitly and realisably computational.

The focus of this work is on the normal processing of continuous speech, and we take three points to follow necessarily from this:

1. Word boundaries are not unequivocally marked

2. The speech signal is not always in and of itself sufficient for recognition,

   and in particular

3. The initial segments of words are not acoustically reliable

Two possible sources can in principle be identified for the solution to the problem posed to the hearer by the second point above: Linguistic structural knowledge (e.g. the phonotactics, lexicon and grammar of the language involved) and contextual/pragmatic knowledge. Both computional experience (most notably in the ARPA speech understanding research effort in the 70's) and psycholinguistic experiments (e.g. (Pollack and Pickett 1963), (Lieberman 1963)) have given strong support to the position that the information required by these sources to recognise the speech signal sometimes occurs temporally after the under-determined section - the so-called right context effect.

Despite this the model we propose, which is for the moment concerned with only the deployment of lexical and morphophonemic structural knowledge, is strictly left-to-right and bottom-up. In the next sections a sketch of the

---

[1]Many of the ideas presented here were devloped in discussions with Ellen Bard, Gerry Altmann and Anne Johnstone, for whose constructive criticism I am grateful. My debt on the computational linguistics side to Ron Kaplan and Martin Kay (Kaplan and Kay 1982), for their ideas and support, is substantial.

model is presented in sufficient detail to show how it can none-the-less generate the right context effect.

The focus in these sections is on the computational characterisation of the model - owing to space restrictions the psycholinguistic consequences will largely be left implicit.

The basic question to be addressed is how to make effective computational use of the dramatic restriction on the problem of speech recognition which is provided by the fact that the signal to be analysed is constrained to consist (by and large) of sequences of (for the sake of argument) English words. Two knowledge bases, lexical and morphophonemic, are required, plus a computational method for deploying them.

The lexical knowledge base consists of a set of trees in which morphemes (stems or affixes) sharing the same initial phoneme sequence follow the same path until their phonemic representations diverge. Recognising a word then consists of stitching together a morphotactically valid path through these trees, and recognising a whole utterance consists of concatenating a sequence of such paths.

The morphophonemic knowledge base consists of a set of (possibly context-sensative) rewrite rules, which are expressed as a finite-state transducer which is interpolated between the lexicons and the input.

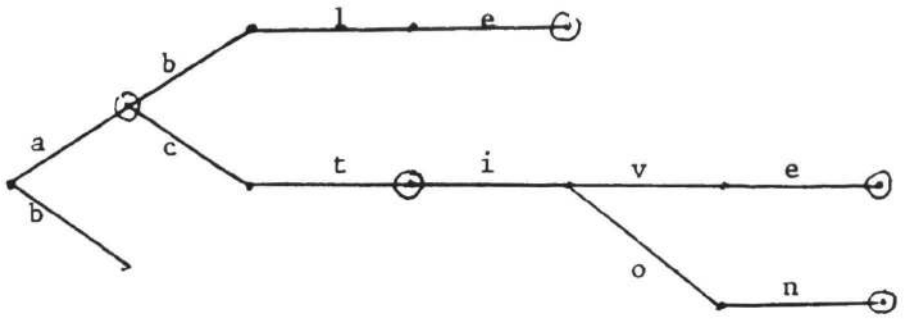
The easiest way to understand this approach is to start with a version appropriate to text, and then suggest how to modify it for speech.

2. The Recogniser - Text version

2.1. The Lexicon

In its simplest form, with only one tree-structured lexicon, looking up a word is straight-forward. One starts at the top of the tree and follows the branch whose label is the first letter of the word in question. From the end of that branch, one follows the branch whose label is the second letter of the word, and so on until one either runs out of letters or branches. In the first case, provided there is an indication in the tree of a word ending at this point, the process has completed satisfactorily. Otherwise, either the dictionary or the candidate word is faulty.

A trivial example will clarify this. Consider the following (extract from a) tree, where circled nodes indicate the presence of words, with lexical entries attached:



If one looks up 'a', 'act', 'able', 'active' or 'action', the process will succeed, but if one tries 'abd', 'abled' or 'acti' the process will fail in one way or another.

Although this approach is clearly much more efficient than one which compares candidate words to entries in a sequential lexicon one after another, it would seem of no interest once we move from text to speech, as it ignores point (1) above, depending as it does on prior knowledge of just that thing we have said is not available in speech - an indication of where words begin. But by elaborating two features already alluded to above, we can then confront this problem in an interesting way.

Firstly, the system can be made to process not words, but strings of characters. Its goal is to find a sequence of paths through the tree which, when laid end to end, will cover the entire target string. To accomplish this it treats spaces and punctuation as alphabetic characters, including them in the lexicon as well.

Secondly, the lexicon consists not of one tree, but of a collection of trees or sub-lexicons, one for each morphological class, including prefixes, stems, noun inflections, verb inflections, derivational suffixes, punctuation and so on. Morphotactic constraints, insofar as they are finite-state, can be expressed by including in each lexical entry an indication of the sub-lexicon(s) where processing should carry on.

The crucial point about this approach is that spaces and other punctuation have now lost their special status with respect to determining word boundaries. It is the act of wrapping around back to the stem (or prefix) sub-lexicon that marks a word boundary.

2.2. Non-determinism and the Chart

This move, as well as other characteristics of the above proposal, introduces a non-deterministic element into the lookup process. A principled approach to this is required, as it will prove crucial once we move to speech. We take the approach of representing all the data in the system, at the character, morph and word level, as edges in a lattice or chart, and adopt the Active Chart Parsing methodology for discharging the non-determinism (Kay 1980), (Thompson and Ritchie 1984). This methodology supports *inter alia* a left-to-right pseudo-parallel process which guarantees that all hypotheses

about possible paths will be pursued, and allows the search strategy through the space of such possibilities to be closely controlled. For example if "r e d" is an initial substring of a text to be analysed, provided the relevant lexical entries are present, three hypotheses will be current after processing those three letters. One will be for the word "red" followed by some as yet unseen inflection and/or punctuation; one for the set of English stems beginning "red", e.g. "rede", "redcoat", "redeem"; and one for the prefix "re-" followed by the first letter of a stem beginning with "d".

The chart-based approach allows ambiguities to be perspicuously and economically recorded. The letters "u n i o n i s e" for instance would be spanned by two edges, one for the stem "union" plus the suffix "-ise", and the other for the prefix "un-", the stem "ion" and the suffix "-ise".

2.3. Morphographemics

The chart also provides a convenient basis for dealing with morphographemics. One can either view morphographemic rules as rewriting rules to be applied by adding edges to the chart (thus y -> ie / _ s can be seen as calling for the addition of a pair of edges with a 'ys' on spanning any 'ies' trio of edges, which then allows e.g. "flies" to be correctly analysed as "fly" plus "-s") (Kay 1977), or as transducers to be interpolated either serially (Kaplan and Kay 1981) or in parallel (Koskenniemi 1983) between the chart and the lexicon.

3. The Recogniser - Speech version

The conversion of the preceding text-based system to handle speech is surprisingly straight-forward. The principal differences stem from the necessity to handle the imperfection of the 'input' acoustic data.

3.1. The Lexicon

Here the change is minimal, simply replacing letter-based branches with phoneme-based branches, and eliminating the punctuation sub-lexicon.

3.2. Non-determinism and the Chart

The chart provides a convenient device for dealing with the indeterminacy of the acoustic data. As in the ARPA speech projects we represent that indeterminacy with a lattice of alternative phonemic analyses, with each alternative analysis of a given segment represented by a distinct edge in the chart.

The Active Chart Parsing methodology insures that all possible matches between phoneme edges and lexicon branches will be tried, but this is not in itself enough. Matching of chart edges against lexicon branches can no longer be considered all or nothing. The imperfections of the input data most be accounted for at this point. For the sake of exposition we will assume that the phoneme edges have some score between 0 and 1 associated with them, representing the degree of certainty assigned to them by the acoustic-phonetic levels of the system.

It follows that hypotheses about the presence of morphs, both complete and partial, will have scores associated with them as well. In the first instance we can think of computing the score incrementally, as a partial hypothesis is extended to cover another phoneme. The score of the new hypothesis will be some function of the score of the prior hypothesis, the score of the newly incorporated phoneme, and the goodness of match between that phoneme and the branch followed in the lexicon. For example if a partial hypothesis existed analysing the initial two segments of a signal as /r e/, with score 0.7, and one of the segments in third place was a /b/ with score 0.8, then we would get among others two new hypotheses, one for /r e b/ with score say 0.75, and one for /r e d/ with score say 0.6.

It follows that there will be a vast increase in the number of partial hypotheses entertained at any given point in the processing. The scoring is what provides a mechanism for keeping this explosion under control. We suppose that at any given point only some number of the highest scoring hypotheses are extended to yield new ones. It is this which produces the appearance of the right context effect. Suppose the actual word uttered in the preceding example had been "reduplicate". In the process of extending the two hypotheses the successors to the one beginning /r e b/ will slowly drop in score, as there is no lexical path which will match well against the input, while the successors to the one beginning /r e d/ will slowly climb in score along the correct path, in effect retroactively confirming the /d/ hypothesis and rejecting the /b/ hypothesis.

An additional level of complexity is required to deal with the issue of word boundaries. A complete word hypothesis is not independent of the subsequent processing it entails. Going back to our example, any hypothesis that the sample in fact begins with the word "red" must be down-graded by the necessarily low score of any hypothesis of a word beginning /oo p l/. Whether this is accomplished by filtering at a higher level, by coupling the scores of successive hypotheses together in some way, or by recording complete hypotheses only after successfully (to some level of score) identifying several words in a row, is a matter of some uncertainty at the moment.

3.3. Morphophonemics

The text-based approach translates easily into a speech-based one, retaining the three options described under that head for the interpretation of the rules.

4. Meta-theoretical Considerations

The model as presented here begs many questions, both down the speech chain, by assuming a phonemic segmentation, albeit imperfect, as input, and up the speech chain, by saying nothing about how syntactic and semantic effects may be felt. We believe the left-to-right, selective interaction approach can work at these levels as well, but are not yet in a position to offer details of how this might be done. The incorporation of many of the ideas presented here in a large scale speech-processing system, now underway, will give substantial impetus to providing such details in these areas as well.

The point we hope to have illustrated with even this brief sketch is that a careful investigation of the computational tools available may reveal a

straight-forward way of providing an account based on a left-to-right process using only selective interaction of a phenomenon usually held to require a right-to-left process using instructive interaction.

The question which must at this point in the discussion be most pressing however is that of computational detail and psychological relevance. Surely a model which is so computationally explicit makes far too many detailed predictions to withstand experimental test. This may prove to be the case, but we feel that if Cognitive Science is to be more than just an indulgent pastime for psycholinguists, then the attempt at fully explicit computational models must be made, so that the consequences in terms of submitting such models to experimental verification can be made in practice, rather than in theory. We furthermore feel that the closer one can get to the automatic periphery of the perceptual system, the better chance one has of making a useful experimental contribution to the development of computational models, and vice versa. In this we see ourselves as in a small way trying to import into the linguistic modality the methodology which was so sucessfully introduced by David Marr for the visual modality.

REFERENCES

Kaplan, R.M. and Kay, M. 1981. Phonological rules as finite state transducers. Presented at the Winter Meeting of the Linguistic Society of America, New York.

Kaplan, R.M. and Kay, M. 1982. Word recognition. Technical Report, Xerox Palo Alto Research Center. To appear.

Kay, M. 1977. Morphological and syntactic analysis. In Zampolli, A., editor, Linguistic Structures Processing. North Holland.

Kay, M. 1980. Algorithm Schemata and Data Structures in Syntactic Processing. In Proceedings of the Symposium on Text Processing. Nobel Academy. To appear. Also CSL-80-12, Xerox PARC, Palo Alto, CA.

Koskenniemi, K. 1983. Two-Level Model for Morphological Analysis. In Bundy, A., editor, Proceedings of the Eighth International Joint Conference on Artificial Intelligence. IJCAI, Los Altos, CA.

Lieberman, P. 1963. Some effects of semantic and grammatical context on the production and perception of speech. Language and Speech 6:172.

Pollack, I. and Pickett, J.M. 1963. The intelligibility of excerpts from conversation. Language and Speech 6:165-171.

Thompson, H.S. and Ritchie, G.D. 1984. Techniques for Parsing Natural Language: Two Examples. In Eisenstadt, M., and O'Shea, T., editors, Artificial Intelligence Skills. Harper and Row, London. Also DAI Research Paper 183, Dept. of Artificial Intelligence, Univ. of Edinburgh.

SYMPOSIUM: NEW PERSPECTIVES ON INTELLIGENT COMPUTER ASSISTED INSTRUCTION

John Black, Yale University, Chair

John Anderson, Carnegie-Mellon University

William J. Clancey, Stanford University

Arthur Graesser, California State University

Derek Sleeman, Stanford University

Elliott Soloway, Yale University

COGNITIVE PSYCHOLOGY AND INTELLIGENT TUTORING

John R. Anderson
Department of Psychology
Carnegie-Mellon University

This paper is an attempt to think through the relationship between intelligent tutoring and cognition. This is a particularly crucial issue to me because I have both worked on the ACT theory of cognitive psychology (Anderson, 1983) and on tutors for geometry and LISP (Boyle & Anderson, 1984; Farrell, Anderson & Reiser, 1984). The paper is in two parts. First, I will start from the goals of intelligent tutoring and reason to how cognitive psychology might serve these goals. Second, I will start from the goals of cognitive psychology and reason to how intelligent tutoring might serve these goals. I found the implications in both directions to be quite surprising.

Implications of Cognitive Psychology for Intelligent Tutoring

Instruction in general, and computer-based instruction in particular, is practiced pretty much as a black art. Students are exposed to various experiences in some vague belief that they will become more capable of dealing with certain vaguely conceived situations in later life. There is both failure to specify what the to-be-learned behavior is and how the experience will affect the learning. While people do learn from such poorly conceived instruction, there is every reason to believe that there is a lot of wasted motion. Clearly, cognitive psychology has a major potential contribution to make in adding precision to the art of education generally, and more specifically to intelligent tutoring. An interesting question concerns which aspects of cognitive psychology are relevant to achieving intelligent tutoring. It turns out that there is a rather interesting demarcation between those aspects of cognitive psychology which are relevant and those aspects which are not.

The first contribution of cognitive psychology would be to provide a well-specified model of the target behavior to be tutored--the goal to which the instruction is directed. In the areas we have thought of tutoring, mathematics and science, this amounts to developing a problem-solving model of the ideal student. Such a model involves a specification of the problem-solving goals, the representation of the relevant knowledge, and the operators that control the transition among goals. Thus the basic ingredients are goal structures, representation, and control. Such student models can be used to represent both the state of the current and the state desired for the student at the end of the instruction.

Creating such models is no mean feat, but given that we had such models, what would we do with them? This requires a theory of the acquisition of problem-solving skills that will specify what the consequences of various experiences will be on the state of the student model. So, we can add a theory of skill acquisition to the list of potential contributions of cognitive psychology to intelligent tutoring.

Exactly how the student model is used in an intelligent tutoring system depends on the learning theory, but our ACT learning theory leads us to

build the tutoring system very directly around the model of the ideal student. Our tutors guide the students through the problems trying to make their steps correspond to those of the ideal student model. At each next step the student's step is compared with the range of acceptable steps and, if it does not correspond, immediate explanation is generated which tells the student what the correct step is and why it is correct. This is a very powerful role for a student model in the tutoring. It effectively structures the whole tutoring interaction. I refer to this mode of tutorial interaction as model-tracing.

Another important consequence of the conjunction of a learning theory and a student model is a prescription for problem sequence. One can look at weaknesses in the student model and construct problems which the learning theory predicts will give the best opportunity to repair the weaknesses.

Another important consequence of the conjunction of a learning theory and a student model is a prescription for problem sequence. One can look at weaknesses in the student model and construct problems which the learning theory predicts will give the best opportunity to repair the weaknesses.

All forms of instruction require that one correctly interpret the student's behavior and successfully impart information to the student. In our model-tracing paradigm this requires that we understand why the student makes moves and that we can correctly describe goals to the student, the operators to apply at these goals, and why these operators should apply. These communication needs place further demands on cognitive psychology:

To the extent that the interaction involves natural language, this brings up issues of natural language processing. These issues of communication turn out to involve issues of knowledge representation. For instance, a major problem turns out to be that of the student acquiring the right representation of domain concepts. In geometry students have to learn the meaning of terms like premise and consequence. In LISP they need to understand the meaning of terms like tail of a list and evaluation of an expression. Part of the tutoring goal becomes teaching of domain concepts. Also one needs to be able to refer to the student's internal goal structures. This requires designing an efficient way of representing the goal structure to the student. For example, in geometry we teach students a graph representation for forward search from the givens and backward search from the to-be-proven statement.

Another issue in communication concerns the serious problems of working memory overload. One has to have an accurate estimate of how much information the student can hold at any one time. It is very easy for instruction to fail because the student cannot process it all.

Requirements from Cognitive Psychology

It is interesting to consider those things that intelligent tutoring needs that cognitive psychology does not offer, those things which it needs that cognitive psychology offers, and those things which it does not need that cognitive psychology offers. It should be recognized that many things go into successful tutoring that have nothing to do with cognitive psychology. This tends to involve the computational aspects of the medium.

In our own work this has included principles of graphics, design of highly modularized systems, efficient implementation of production systems, principles for automatic problem generation, and principles for inducing student models from surface behavior. Each of these is a major concern in our work but a concern that is totally devoid of any psychological content.

Then there are a set of psychological issues that seem key. To review, these were:

1. Theory of goal structures,
2. Theory of control,
3. Theory of knowledge representation,
4. Theory of skill acquisition,
5. Theory of natural language understanding, and,
6. Theory of working memory limitation.

What is most interesting, however, is the large segments of cognitive psychology that seem irrelevant. Specifically, there is no role for a theory of the speed or probability with which knowledge is stored, retrieved, or applied. I cannot think of one timing result from experimental psychology that has implications for tutor design. We would have our tutors engage in the same behavior independent of how long the students studied that information, how long it took them to retrieve it, and whether they could remember it. No matter what, we would test for the knowledge at a later point and provide remedial instruction if the students are wrong. Issues about the exact times and probabilities of performance are irrelevant. We could build into our tutor the ability to make accurate predictions about times and probabilities, but it could not use these predictions to further its tutoring. These are predictions at the wrong level of analysis.

It is useful to make a distinction between two levels of cognitive theory--the level of process specification and the level of process implementation. This can be done in analogy to the distinction between programming language and machine implementation. Although there is some controversy about the matter (e.g., Anderson & Hinton, 1981), it seems that there is a programming language of the mind (compare to Fodor, 1975) in which cognitive processes are specified. Such a language is at a level analogous to the LISP programming language. In our work, we take the ACT production system as this language. The issues listed as relevant to tutoring--goal structure, control, knowledge representation, skill acquisition, and working memory--are all concerned with aspects of this language. The one other relevant issue, natural language understanding, is concerned with a process implemented <u>in</u> this mental language.

Much of a cognitive theory like ACT is concerned with how this mental language is implemented. This is like asking how LISP is implemented on a computer. It is these aspects of implementation which seem not to be relevant to tutoring. This is not to say, of course, that they are uninteresting.

So, the conclusion is that the division between issues of mental language and issues of its implementation corresponds to the division between those aspects of cognitive psychology which are relevant to

intelligent tutoring and those aspects which are not. To the extent that intelligent tutoring work can progress without consideration of the implementation, this is evidence for the distinction between the mental programming language and its implementation.

Implications of Intelligent Tutoring for Cognitive Psychology

It is natural to speculate that the relationship might be symmetrical and that work in intelligent tutoring might have implications only for the mental language level and not for the implementation level. However, this is not so. By looking at the history of latency and success in working with a tutor one may be able to make numerous inferences about cognitive implementation. However, it is hardly the only way to make inferences about cognitive implementation and for many issues, traditional experiments are more effective. On the other hand, I am prepared to argue that the only methodologies that can get at the language level are the tutoring methodology and variants on that methodology.

To make this argument, I will consider an analogy: Suppose a programmer presented to us a fairly complex program, and we wanted to test whether it was a LISP program. In the analogy, LISP corresponds to our theory about mental programming language and the programmer is the learning system that changes the mental program. To make the analogy work we will have to assume we cannot ask the programmer what the language is nor can we physically inspect the program. All we can do is look at the input-output behavior of the program. Now it is well known that any computationally universal programming language can produce any input-output behavior. Similarly, any "reasonable" theory of the mental language could produce any behavior. It is unlikely that implementations in two languages will differ interestingly in their timing behavior, particularly if all we can look at is relative time to solve two problems, not absolute time. Similarly, it is unlikely that we can choose between two theories of mental language by looking at processing times or probabilities of correct responses.

How in fact are decisions made about the mental language? The argument is usually that a particular type of input-output behavior is characteristic of the "programming style" appropriate to a particular mental language. A classic example of this kind of reasoning is the paper by Hayes-Roth & Hayes-Roth (1979) arguing for opportunistic planning. Similarly, one might argue that a program that printed out

```
=>(factorial 4)
1<Enter> factorial (4)
|2<Enter> factorial (3)
| 3<Enter> factorial (2)
| |4<Enter> factorial (1)
| | 5<Enter> factorial (0)
| | 5<EXIT> factorial 1
| |4<EXIT> factorial 1
| 3<EXIT> factorial 2
|2<EXIT> factorial 6
1<EXIT> factorial 24
24
```

was written is LISP. In certain contexts, this might be a reasonable inference. However, in the human case the inference from surface behavior to mental programming language is quite perilous.

Probably a better method would be to ask the programmer to make some addition or modification to the program and see how long it took to accomplish that and with how much difficulty. Certain things are easy to implement in LISP and certain things are not. Analogously, in the case of the mental programming language it is very informative to look at what the effects are of various instructional manipulations. Certain instructional goals should be easy to achieve and certain should be hard.

Thus, the way to test hypotheses about the mental programming language is to perform instructional experiments. Of course, these need not take the form of constructing intelligent tutors, but there are advantages to that specific methodology. First, computer implementation operationalizes very precisely the instructional methodology. Second, interesting instructional manipulations take relatively long times. It is hard to motivate a subject population in such experiments unless one is trying to teach a useful domain. There are serious problems with trying to teach a useful domain in some perverse way to test cognitive theory. It might be informative to study the instruction of geometry without diagrams, for instance, but I could hardly use the scientific information gathered to mollify angry parents whose children were failing high school math. So for these reasons, one is naturally led to conduct instructional tests of cognitive theory as good faith efforts to have computers to teach real topics to real students. Beyond this there is something very informative about finding the optimal tutor for a topic--whether that optimum is just a very local one or a global one. The fact that the tutor is at an optimum places important constraints on the nature of the mental programming language.

Example Tutorial Experiments

It is not hard to think of tutorial manipulations that serve to test alternative theories of the mental system. Consider for instance the contrast between schema systems and production systems. The essence of a schema system is the notion that the critical knowledge structure is an abstract schema which is used for different purposes according to the current goal. Thus, there might be a recursion schema which could be used for coding, debugging, and evaluating. According to schema theory, it is necessary and sufficient to build up a rich schema representation of recursion. From that suitable performance would flow in different tasks. In contrast, production systems theory leads to the conclusion that there are task specific rules that must be acquired. Obviously, such theories lead to very different predictions about the relative merits of general versus task-specific instruction.

To consider a controversy more local to CMU, consider two types of productions system architectures. One (e.g., ACT) holds that information first comes into the system in a declarative form and then is complied into production form with practice. The other holds that all information is procedural, and there is no separate declarative representation. The first viewpoint would hold that there is a point to understanding and memorizing generally relevant facts in advance to actually using them. This would

establish the declarative encoding and permit the procedural compilation right away. The second viewpoint would hold that students might as well be told the specific rules as they are needed in problem solving without advance preparation.s

As a third and final contrast, consider the claim that goal selection is opportunistic and data-driven versus the claim that goal selection is in response to some hierarchical plan. The former would argue for a tutorial strategy that gave the student a lot of flexibility in what to do next whereas the second would promote a rather rigid flow of control through a problem.

Each of the above three contrasts are somewhat caricatures, and people may want to argue for other predictions. Rigorous predictions require careful interfacing of specific proposals with specific learning situations. However, the examples do serve to indicate how tutorial manipulation can go to the heart of fundamental issues about the mental programming language. (Also, it needs to be acknowledged that because these tutors require effective computational principles as well as correct cognitive assumptions, a particular tutor can fail even though the underlying theory is correct.)

Conclusion

So in conclusion, there seems to be a very intimate relationship between a subset of issues in cognitive psychology and intelligent tutoring. It goes beyond the relationship between a science and its applications. Research in intelligent tutoring is the natural methodology for study of the mental programming language. The relationship is this strong because the human mind is more than a naturally occurring object for scientific study; it is an object whose evolution was principly directed so it could be instructed in new problem-solving skills.

References

Anderson, J.R. (1983). The Architecture of Cognition. Cambridge, MA: Harvard University Press.

Anderson, J.A. & Hinton, G.E. (1981). Models of information processing in the brain. In G.E. Hinton & J.A. Anderson (Eds.), Parallel Models of Associative Memory. Hillsdale, NJ: Erlbaum.

Boyle, C.F. & Anderson J.R. Acquisition and automated instruction of geometry proof skills. Paper to be presented at the Annual Meeting of the American Educational Research Association.

Farrell, R., Anderson, J. & Reiser, B. (1984). Interactive Student Modelling in a Computer-Based LISP Tutor. Submitted to Cognitive Science.

Fodor, J.A. (1975). The Language of Thought. New York: Thomas Y. Crowell.

Hayes-Roth, B. & Hayes-Roth, F. (1979). Modelling planning as an incremental, opportunistic process. Proceedings of IJCAI-709, 375-383.

Teaching Classification Problem Solving

William J. Clancey
Heuristic Programming Project
Stanford University
Stanford, CA 94305

A logic-based analysis of heuristic programs suggests a problem solving model that is strongly supported by cognitive science studies of categorization and understanding. This extended abstract describes the model, then outlines epistemologic considerations for constructing instructional programs that can both generate and recognize such problem solving behavior.

A broad range of familiar problems--embracing forms of diagnosis, catalog selection, and skeletal planning--can be characterized in terms of classification [3]. Solutions to these problems have a characteristic inference structure, involving systematic relation of data to a previously known set of solutions by processes of data abstraction, heuristic association to a schema network, and refinement. Previous research has described classification problem solving almost exclusively in terms of identification of an unknown object or phenomenon, what we commonly call "diagnosis." However, a study of the heuristic programs called "expert systems" indicates that reasoning involved in selecting a product or service is characterized by the same inference structure. Moreover, a common kind of problem involves sequential classification problems: first stereotypically characterizing a user's needs or requirements and then heuristically selecting a product or service [11]. Routine software configuration and experiment planning problems are similar: a template solution is found and then refined [6]. Studies of routine physics problem solving 2] show the same process of problem feature abstraction, heuristic association, and refinement.

A computer program called NEOMYCIN implements a form of classification problem solving in a general way [4]. The epistemologic distinctions made in the implementation make it possible to use NEOMYCIN for both generating and recognizing classification problem solving behavior. These distinctions were previously put forth by logic theorists [8, 9]. In our view they are:

- The knowledge to solve a problem is distinct from its implementation in some information processor [10]. Specifically, in AI research representation/interpreter descriptions of computational models have been frequently confused with more abstract descriptions of what the problem solver is doing and what he knows.

- These abstract descriptions of reasoning should make a distinction between inference structure (logic terms and relations) and process structure (rules of inference and strategic operators). Rule-based programs like MYCIN and R1 combine factual knowledge with procedural information about how inferences are to be made.

- There is a distinction between statement of a relation or procedure and its justification or basis. For example, heuristics are justified

by some, generally unstated, model of the world. Making explicit the process structure requires not only stating the procedure in computational terms (such as input, sequence, iteration), but also in terms of underlying assumptions and constraints that justify the procedure, from which it can be derived.

We briefly relate these considerations to NEOMYCIN. The classification problem solving model describes what NEOMYCIN does, independent of its representational scheme (rules, frames, etc.). The program's facts and heuristic are stated separately from the inference procedure. The inference procedure is represented in a special language that allows us to "declaratively" express computational constructs, as well as to annotate some of the underlying constraints that are useful for student modeling. This combined design enables us to use NEOMYCIN to directly solve problems independently, or to "run the program backwards" to predict and recognize behavior that fits its model. The program can also provide a trace of its reasoning, serving as a crude form of explanation [7].

With a general model of problem solving implemented that uses a body of "expert" knowledge, we are now investigating student behavior. Specifically, we can use the program to provide partial interpretations of what students are doing, with differences indicating where the model must be extended (and what kinds of assistance students require). Following from some proposals made by J. S. Brown [1], we are designing a sequence of instructional programs by which we can explore students' reasoning as they explore NEOMYCIN. The first of the series includes GUIDON-WATCH (for learning that classification problem solving has a certain structure), GUIDON-MANAGE (for learning the effects of problem solving operators [5]), and GUIDON-ANNOTATE (for integrating operators with domain knowledge and recognizing patterns of efficient problem solving).

In conclusion, a synthesis of theories of logic, experience in writing expert system programs, and cognitive science studies has enabled us to develop a computational model of problem solving that can be useful for teaching. The ke idea is that epistemologic distinctions--knowledge/processor, inference/process, and relation/justification--are an intricate part of the computational model, providing the basis for not only generating problem solving behavior, but also explaining and recognizing it.

REFERENCES

[1] Brown, J. S. Process versus product--a perspective on tools for communal and informal electronic learning. In Education in the Electronic Age, proceedings of a conference sponsored by the Educational Broadcasting Corporation, WNET/Thirteen. July, 1983.

[2] Chi, M. T. H., Feltovich, P. J., Glaser, R. Categorization and representation of physics problems by experts and novices. Cognitive Science 5:121-152, 1981.

[3] Clancey, W. J. Classification problem solving. Technical Report HPP-84-7, Stanford University, March, 1984. (Submitted for publication in proceedings of AAAI-84).

[4] Clancey, W. J. Acquiring, representing, and evaluating a competence model of diagnostic strategy. Technical Report HPP-84-2, Stanford University, February, 1984. (To appear in Chi, Glaser, and Farr (Eds)., The Nature of Expertise).

[5] Clancey, W. J. The operators of diagnostic strategy. In Cognitive Science Proceedings, pages. Boulder, CO, June, 1984. (To be presented at the Symposium on Knowledge-based Medical Problem Solving).

[6] Friedland, P. E. Knowledge-based experiment design in molecular genetics. Technical Report STAN-CS-79-771, Stanford University, October, 1979.

[7] Hasling, D. W., Clancey, W. J., Rennels, G. R. Strategic explanations for a diagnostic consultation system. The International Journal of Man-Machine Studies :(in press), 1984.

[8] Hayes, P.J. In defense of logic. In Proceedings of the Fifth International Joint Conference on Artificial Intelligence, pages 559-565. August, 1977.

[9] McCarthy, J. and Hayes, P. Some philosophical problems from the standpoint of Artificial Intelligence. In B. Meltzer and D. Michie (editors), Machine Intelligence 4, pages 463-502. Edinburgh University Press, 1969.

[10] Newell, A. The knowledge level. Artificial Intelligence 18(1):87-127, 1982.

[11] Rich, E. User modeling via stereotypes. Cognitive Science 3:355-366, 1979.

How Do Psychologists Think Anyhow?

John Black and Arthur Graesser
Yale University and California State University at Fullerton

What do psychologists know and how do they use that knowledge to answer questions?[1] These are two of the issues we have had to address as part of a project to design an intelligent computer assisted instruction (ICAI) system that will help people learn about psychology. Questions like these are critical for any ICAI system, because central to the design of such systems is determining how knowledge of the domain is to be represented, determining what that knowledge is, and determining how that knowledge is to be used to interact with the students (e.g., to answer student questions).

Several ICAI systems address these issues for rule-based domains like electronics (e.g., Brown and Burton, 1975), arithmetic (Brown and Burton, 1978), algebra (Sleeman, 1982), and computer programming (Soloway, Rubin, Woolf, Bonar and Johnson, 1982); but how to address them is less clear for non-rule-based domains like psychology, history, literary studies, law, and management of organizations. In fact, domains like medical diagnosis are probably less rule-based than current expert systems (e.g., Shortliffe, 1976) make them appear, and are thus more like psychology than algebra. We have approached these issues of how to represent expert knowedge in non-rule-based domains by trying to devise an ICAI system that contains knowledge about psychology in a form that allows it to intelligently answer student questions as a human expert would (or perhaps better).

Psychological Thinking

The kinds of knowledge and reasoning used by psychologists to answer questions is illustrated by the following "think aloud" protocol of a cognitive psychologist (who is a faculty member at a research university) determining how to answer a student question. The psychologist was given the background and student question, then gave the following verbal report of his thoughts (which we have edited slightly) as he figured out how to answer the student:

> ***Background:*** The student asking the question is in a class that has just learned about depth of processing and elaboration theories of memory.
>
> ***Student Question:*** I'm taking Spanish and I was wondering if using imagery is a good way to learn the vocabulary?
>
> ***Psychologist's Reasoning:*** Well, imagery is a deep processing task, so by depth of processing theory, imagery should give good memories... Ah, I'm remembering that there is an experiment about this by Atkinson and Raugh. They found that interacting images is the critical thing. That is, the subjects had to form images of the English words, then form images of the Spanish words' sounds, and then show the images interacting. Just forming images by themselves was not good enough they had to be interacting ... The example I remember (I'm not sure it's right) is "horse -- caballo." Here the image is of a horse kicking an eye because "caballo" sort of sounds like "eye." So I need to tell the student about this study and that they must use interacting images... Ah, I can explain this using elaboration, because the interacting images provide elaborative connections between the words...

So what is going on here? Our psychologist starts out by trying to relate the query to the depth of

---

[1]This research was supported by a grant from the IBM Corporation. However, the views expressed in this paper are those of the authors and are not necessarily endorsed by IBM.

processing theory of memory (probably because he has been told that the student knows that theory), then realizes that he knows of an experiment that directly addresses the strategy of using images to learn vocabulary. He retrieves from memory the details of the relevant experiment and realizes that the results provide more specific information about exactly how to use images to learn vocabulary -- namely, use interacting images. But how is he to explain this to the student? The depth of processing theory does not have a clear way of explaining why interacting images would be better than images alone, so he retrieves what he knows about the elaboration theory of memory (the other theory he has been told the student knows) and realizes that that theory provides an explanation. His eventual answer to the student is to describe the specific experimental result and rationalize it using the elaboration theory of memory.

Note that the crucial information for the answer is provided by the memory for a specific expert, not by the models. Thus the important reasoning here is case-based reasoning rather than rule-based reasoning, because the experiment is a specific case or observation and the needed specific rule is stored as part of the memory for that case. If this had been a rule-based domain, then the expert would have merely derived the answer using the rules stored as part of the knowledge representation of the model -- perhaps citing a case to illustrate the rule. However, such complete models are rare in fields like psychology, so case-based reasoning seems to dominate. Thus experts in fields like psychology seem to have a mental representation composed of cases (memories for specific experiments) that are organized so that they can be accessed directly to answer queries. These specific cases also seem to have links to various explanatory frameworks (e.g., the depth of processing and elaboration theories of memory) that can provide explanations for the results of the experiments.

The *ECALP* System

We have been working on an ICAI system that we call ***ECALP***, for Expert Computer Assisted Learning of Psychology. While it is far from adequate to do the kinds of reasoning shown in our protocols of actual reasoning by psychologists, it does show a few of the needed features in rudimentary form. ***ECALP*** is implemented as a PROLOG program that uses two levels of representation to embody psychological knowledge -- a conceptual packet level and an explanatory packet level.

The conceptual packet level contains interrelated packets of information each using conceptual graph structures (Graesser, 1981) to represent experimental results (e.g., Byrne and Nelson found that the proportion of similar attitudes was more important in causing attraction than was the number of similar attitudes) and general facts (e.g., person X familiar with person Y causes X to be attracted to Y). Each conceptual packet has six information slots: one that contains a pattern describing when the packet will be relevant, another slot that contains a unique identifier, a third slot that is the category of the packet (e.g., event or state), a fourth slot is a list of links from other packets (e.g., is a consequence of, is an implication of), a fifth slot is a list of links to other packets (e.g., causes, implies), and the sixth slot provides alternative ways of expressing the packet information to the student. The first slot provides the mechanism needed to access the conceptual packet directly when relevant queries arise and the fifth slot provides access to other relevant conceptual and explanatory packets.

The explanatory packet level contains interrelated packets of information that organize the information at the conceptual packet level that relates to the various relevant explanatory frameworks. The explanatory packets contain six information slots: one slot containing the names used to describe the explanatory framework (e.g., depth of processing, levels of processing), another slot describing the general phenomena addressed by the framework (e.g., human memory), a third slot describing the type of explanatory framework (e.g., theory, model, hypothesis, or folklore), a fourth slot containing indexing links to the conceptual packets that describe the mechanisms of the framework, a fifth slot containing indexing links to conceptual packets that describe the predictions of the framework, and a sixth slot that provides indexing links to conceptual packets that describe experimental results providing evidence related to the explanatory framework.

Using this two level representation, the question-answering procedures given in Graesser and Murachver (in press), and a simple definite clause grammar (Pereira and Warren, 1980; McCord, 1982), we have been able to get ***ECALP*** to flexibly answer questions of the following kinds:

1. What does X mean?
2. Is the answer to X YES or NO?
3. How does X occur?
4. Why does X occur (or X exist)?
5. What are the consequences of X occuring (or existing)?
6. What is the evidence for X?
7. According to theory T, <question>?

With these capabilities ***ECALP*** can serve as an expert consultant on psychology (to a limited extent at this time) that the student can try out ideas on and use to seek further information. When the student first broaches a topic (e.g., human memory or interpersonal attraction), ***ECALP*** gives the student a few key concepts that the student can then use to guide further queries (e.g., asking what various terms mean) and to derive ideas of their own which ***ECALP*** can evaluate (e.g., Is semantic encoding remembered better than acoustic encoding?). The teaching strategy used in ***ECALP*** is to try to establish a learning environment |n which the students can themselves generate most of the ideas about a topic. The reason for this is that if students can generate ideas themselves, then they will know and remember those ideas beter than if they were merely told the ideas (Jacoby, 1978; Black and McGuigan, 1983; Carroll and Carrithers, 1983). We believe that ICAI systems that exploit the potentially powerful learning mechanisms of such generation effects will be more effective than other ICAI systems, but this is an hypothesis that needs to be empirically evaluated.

Conclusions

In our investigation of psychology, we have found that ICAI systems for such non-rule-based fields need knowledge representations of specific cases in addition to the rules represented in current rule-based ICAI systems. We have implemented a prototype of one such system (***ECALP***) that uses cases and rules to answer student queries about psychology. This system also embodies the teaching strategy that we think is potentially the most powerful: namely, presenting the student with a few key ideas and then having them generate the rest.

References

Black, J.B. and McGuigan, S. The memory strength of inferences in text understanding. Paper presented at the 24th Annual Meeting of the Psychonomic Society. San Diego, CA, 1983.

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

Brown, J.S. and Burton, R.R. Diganostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 1978, **2**, 155-192.

Carroll, J.M. and Carrithers, C. Blocking learner error states in a training environment. Paper presented at the 24th Annual Meeting of the Psychonomic Society. San Diego, CA, 1983.

Graesser, A.C. *Prose comprehension beyond the word.* New York: Springer-Verlag, 1981.

Graesser, A.C. and Murachver, T. Symbolic procedures of question answering. In A.C. Graesser and J.B. Black (Eds.) *The psychology of questions.* Hillsdale, NJ: Erlbaum, in press.

Jacoby, L. On interpreting the effects of repetiton: Solving a problem versus remembering a solution. *Journal of Verbal Learning and Verbal Behavior*, 1978, **17**, 649-668.

McCord, M.C. Using slots and modifiers in logic grammars for natural language. *Artificial Intelligence*, 1982, **18**, 326-367.

Pereira, F.C.N. and Warren, D.H.D. Definite clause grammars for language analysis -- A survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 1980, **30**, 231-278.

Shortliffe, E.H. *Computer-based medical consultation: MYCIN.* New York: American Elsevier, 1976.

Sleeman, D. Assessing competence in basic Algebra. In D. Sleeman and J.S. Brown (Eds.) *Intelligent tutoring systems.* New York: Academic Press, 1982.

Soloway, E., Rubin, E., Woolf, B., Bonar, J. and Johnson, W.L. *MENO-II: An AI-based programming tutor.* Reserch Report #258, Department of Computer Science, Yale University, 1982.

Mis-generalization: an explanation of observed mal-rules

D. Sleeman
Heuristic Programming Project
Department of Computer Science
STANFORD University
California 94305

1. Introduction

Intelligent Tutoring Systems (ITSs) force their implementors to be explicit about domain knowledge, tutoring rules, likely student misunderstandings for a particular domain, etc. Although this explicitness is demanding it does have the advantage that if the system behaves differently than expected, the implementor can determine the reasons for this, modify the suspected rule/knowledge and rerun the system.[1] Further once one has identified misunderstandings which one believes arise pretty consistently in a subject domain, by this or more conventional techniques, then a series of additional investigations are possible. These include:

1. hypothesizing the nature of the processes used by students to solve tasks given the incorrect/buggy/mal-rules.

2. building a remedial subsystem which exploits the inferred student model (this will involve further analysis of teacher-student remedial dialogues).

3. undertaking studies aimed at improving the initial instruction in the domain so as to avoid (some of) the observed difficulties.

In this article I discuss the first of these points in the context of extensive studies undertaken with 14- to 15-year-old algebra students. The Leeds Modelling System, LMS, was implemented and a database of examples, correct- and incorrect, or *mal-* rules had been established which was sufficient to diagnose the majority of difficulties encountered by 15-year-old students, Sleeman [1982]. The same database was then used with 24 14-year-old students and the outcome was very different. A high percentage of the student errors were *not* diagnosed by LMS. The investigator analysed these protocols in some detail and then carried out individual interviews to determine the nature of the students' difficulties, Sleeman [1983a]. The pertinent observations from this latter experiment are:

1. Students appear to regress under cognitive load. That is they are often able to use a particular rule correctly in the context of simple tasks, but make errors with this same rule when the tasks are more complex.[2] See Sleeman [1983a] for examples.

---

[1] The approach used within the Expert Systems paradigm.

[2] This analysis *assumes* that domain rules are independent and one rule does not subsume another.

2. There appears to be a number of clearly identifiable *types* of error, (section 2).

3. Students use a number of alternative "methods" to solve tasks of the same type, (section 3).

2. Observed types of student errors

From the protocols and the interviews I concluded that in this domain errors could be classified as: manipulative, parsing, execution/clerical and random. The first two topics will be dealt with in some detail in the rest of this section; see Sleeman [1983b] for details of the others.

2.1 Manipulative Errors

I define a manipulative mal-rule to be a variant on a correct rule which has one substage either omitted or replaced by an inappropriate or incorrect operation, c.f., Young & O'Shea [1981]. For example, MNTORHS[3] is a mal-rule which captures the movement of a number to the other side of the equation, where the student omits to *change* the sign of the number. MXTOLHS is the mal-rule which corresponds to the analogous X-to-lhs rule. (Most of the errors noted with 15-year-old students were of this form.) Note that this schema would ALSO generate many mal-rules, which we have NOT yet observed; in the next paragraph we give an explanation why some of the possible mal-rules are not observed.

a) Analysis of some manipulative mal-rules: A schema for generating manipulative mal-rules
In a recent experiment we noted three (additional) mal-rules which can be explained by this mechanism. Two of them will be analysed in some detail:

1. A variant on SOLVE. The variant on SOLVE transformed:

   $$4 * X = 6 \text{ to } X = 6$$

   whereas SOLVE would change the same expression to X = 6/4. It is suggested that the student realizes he has a task in which the SOLVE rule should be activated and forgets to apply one of the operations, namely dividing by M. SOLVE has three principal actions: noting down N, the divide symbol and M, and so this mal-rule could be said to be omitting some of the principal steps. Furthermore, it appears that students have an idea about the acceptable FORM of answers and so given the above task we have *not* seen X = 6/ or X = /4.

2. A variant on SIMPLIFY. Examples of the two mal-rules noted here, which have occurred reasonably frequently are:

   $$X = 6/4 \Rightarrow X = 3/4$$

---

[3] MNTORHS is short for *mal-number-to-rhs* rule.

$$X = 6/4 \Rightarrow X = 6/2$$

(The SIMPLIFY rule transforms the same expression to X = 3/2).

Again we argue that the above observations can be explained if we assume that this rule has several principal steps including, calculate the common factor, divide "top" by common factor, divide bottom by common factor, write down the components, and that each of these mal-rules corresponds to one step being omitted.

b) "Grain size" and manipulative mal-rules.

There is a sense in which detailed analyses of manipulative mal-rules allows one to infer the substep processed by students, and this in turn allows one to predict the set of mal-rules that will be encountered in a domain. (Bearing in mind the idea of acceptable form outlined above). Further, one might argue that the representation of the tasks should be at this "lower" level; the justification for the representation chosen, is that this appears to be more consistent with the collected verbal and written protocols for students solving these tasks. The schema discussed above for generating manipulative mal-rules by omitting, or modifying, one substep is thus consistent with Young and O'Shea's modelling of subtraction.

2.2 Incorrect Representation of the Task or Parse Errors

I assert that many of the students whom we interviewed carried out steps of the computations in ways which would not fall within the definition given earlier for manipulative mal-rules. Below, I give typical protocols for two students working the task 6*X = 3*X + 12:

```
I:    6*X = 3*X + 12          II:   6*X = 3*X + 12
      9*X = 12                      X + X = 12 + 3 - 6
      X = 12/9                      2*X = 9
      X = 4/3                       X = 9/2
```

When I pressed the "first" student for an explanation of how the original equation was transformed into the second, i.e., 9*X = 12, the student talked about moving the 3*X term across to the left hand side. Thus the interviewer concluded that this was an instance of a student using a variant of the correct rule, namely a manipulative mal-rule. When the "second" student was pressed he simply asserted that the change from the original equation to the second line "was all done in one step". Hence the interviewer concluded it was a very different type of mal-rule involved and not a simple variant on the correct rule. Thus the interviews provided essential additional information as, of course, the second student's protocol could be explained by the use of MXTOLHS and the mal-rule:

$$M * X \Rightarrow M + X$$

[4]which some people might wish to argue constitutes a manipulative mal-rule (replacing the * operator by the + operator). Even if we did not have the additional experimental evidence, this investigator would maintain that such a transformation belays a profound misunderstanding of algebraic notation and so should be considered as a parsing mal-rule. See Sleeman [1983b] for additional discussion of this issue.

---

[4]Where M stands for an integer, and where in the above example 6*X => 6 + X and 3*X => 3 + X.

3. Bug Migration or Using Alternative Methods

Repair theory gives a neat explanation for the observed phenomena of bug migration in the domain of multi-column arithmetic, Brown & VanLehn [1980], namely that the student will use a related family of mal-rules, and possibly the correct rule, during a single session with one particular task set.

There seems to be an alternative explanation which should also be considered. Although a task-set may have been designed to highlight one particular feature, the student may spot completely different feature(s) and these may dominate his solution.[5] Repair theory accounts for some bugs by hypothesizing that the student had not encountered the appropriate teaching necessary to perform the task. Suppose we make the converse assumption, that the appropriate teaching had been carried out, and further suppose that *some* students[6] do not gain competence in this domain by being told the rules but rather by inferring rules for themselves by noting the transformations which are applied to tasks by the teacher and in texts.[7] It seems reasonable that the student's inference procedure should be guided by his previous knowledge of the domain, in this case the number system, and that the student will normally infer several rules which are consistent with the example, and not just the "correct" rule. Indeed due to some missing knowledge the "correct" rule may not be inferred. (And so the fact that the student never uses the "correct" method along with several "buggy" methods is not evidence that he has NOT encountered the material before). We shall refer to this process as Knowledge Directed Inference of Multiple rules, or mis-generalization for short.

Suppose, the student saw the following stages in an algebraic simplification:

$$3 * X = 6 \quad => \quad X = 6/3$$

Then he might infer

$$X = \text{RHS number/LHS number OR } X = \text{LARGER number/SMALLER number}$$

We will surmise how a student would use such a rule-set. We will suppose that the abler students actively experiment with different "methods", and use their own earlier examples, examples worked by the teacher and in the text to provide discriminatory feedback. From our experiment with 14-year-old students we have direct evidence that some students are aware of having a range of applicable rules and being unsure of when to select a particular method, Sleeman [1983a]. That study did not provided any insights into the rule-selection processes used by these students. We could suggest the common default, i.e., that the process is random. However, studies in cognitive modelling have

---

[5] Earlier Sleeman and Brown [1982] have argued: ".....Perhaps more immediately, it suggests that a Coach must pay attention to the sequence of worked examples, and encountered task states, from which the student is apt to abstract (invent) functional invariances. This suggests that no matter how carefully an instructional designer plans a sequence of examples, he can never know all the intermediate steps and abstracted structures that a student will generate while solving an exercise. Indeed, the student may well produce illegal steps in his solution and from these invent illegal (algebraic) "principles". Implementing a system with this level of sophistication still presents a major challenge to the ITS/Cognitive Science community... "

[6] Note I am *not* claiming that there is a *single* mechanism.

[7] Independently, VanLehn has come to a similar conclusion, the Sierra system described in his thesis relies heavily on inference, VanLehn [1983].

already discredited this explanation many times, so we will postulate that the process is deterministic but currently "undetermined". It is further suggested that tasks which show a rule is inadequate will weaken belief in the rule, but once a (mal) rule is created it may not be completely eliminated - particularly if the "counter-examples" are not presented to the student for some period. Thus given this view point, the phenomena of bug-migration occurs because the (less able) student has inferred a whole range of rules and selects a rule using a "black-box" process.. Given a further task, he again chooses a method and hence selects the same or an alternative algorithm, influenced partly by the relative strengths of the rules. That is if the relative weights are comparable, it is more likely that the student will select a different method for each task. If one weight "dominates" then it is likely that the corresponding method will be selected frequently. Further, if only one (mal) rule is generated by the induction process then this approach predicts that the student will consistently use that rule.

We suggest that many of the bugs encountered in the subtraction domain can be accounted for by this (inference) mechanism. For instance the Smaller-from-Larger bug, where the smaller number is subtracted from the larger independent of whether the larger number is on top or the bottom row, seems one such example, Brown & Burton [1978] and Young & O'Shea [1981]. Brown & VanLehn [1980] report that because borrowing was introduced, with one group of students, using only tasks with 2 columns, these students inferred that whenever borrowing was involved they should borrow from the left-most column, their "Always-Borrow-Left" bug. So it appears important to ensure that the example set includes some examples to counter previously experienced mal-rules. Indeed it seems as if task-sets can be damaging if they are too preprocessed and contain too little "intellectual ruffage"; Michener [1978] puts a similar argument. Additionally, Ginsburg [1977], quotes several instances of young children inferring the name "three-ty" for 30, given the names for "3", "4", "5", "40", "50", "60". So given the wealth of experimental evidence this alternative explanation should be given serious consideration.

Further, I have two philosophical reservations about repair theory. Firstly, that by some mechanism not articulated all students acquire a common set of impasses, and moreover they consistently observe these. Secondly, repair theory which sets out to explain *major* individual differences at the task level, itself proposes a specific mechanism *common* to all students.[8] On the other hand, mis-generalization predicts that the individual's initial knowledge profoundly influences the knowledge which is subsequently inferred, and captures the sense in which learners are active theory builders trying to find patterns, making sense out of observations, forming hypotheses, and testing them out.

4. Summary

Firstly, there are two hypotheses which explain bug-migration the one given by repair theory and the one put forward here, namely mis-generalization. Of course it is possible that each may be applicable in different situations. Secondly, several "algorithms" have been presented for creating student models. I believe these are suggestive about the processes used when a student solves (these) tasks. Repair theory suggests that it can be explained by making "repairs" to incomplete core-procedures, whereas Young and O'Shea suggest that it is adequate to take a correct procedure and merely delete components. The data for the algebra manipulative mal-rules can be adequately explained by either. However, Young and O'Shea's approach seems inadequate to explain the

---

[8] Indeed I am concerned that many theories of (child) development do *not* accept the possibility of there being significant individual differences in development, but merely in the individual's *rate* of progress and the level of his final maturation.

parsing mal-rules. Indeed, we have to extend revised repair theory before the results reported here can be accommodated. This paper claims that there are two very different types of malrules at large with algebra students - namely manipulative and parsing mal-rules. And that this second category of algebra errors, and much of the data collected in other areas, appears to be best explained by a further mechanism, namely mis-generalization. However, once *inferred* I believe rules are additionally *applied* incorrectly, and that the mechanism(s) described in Young & O'Shea, repair theory and section 2.1, are appropriate for this stage.

5. Acknowledgements

To Mr. M. McDermot and students of Abbey Grange School, Leeds, for providing fascinating sets of protocols. To Pat Langley, Kurt VanLehn, Jaime Carbonell, Stellan Ohlsson, Peter Jackson, Alan Bundy and William Bricken for numerous discussions about this work. Additionally, William Bricken made some insightful comments on an earlier draft of this paper.

6. References

J.S. Brown & R.R. Burton (1978). Diagnostic Models for procedural bugs in basic Mathematical Skills, in *Cognitive Science, 2,2*. pp155-192.

J.S. Brown & K. VanLehn (1980). Repair Theory: A generative theory of bugs in procedural skills. *Cognitive Science, 4*. pp 379-426.

H.P. Ginsburg, (1977). *Children's Arithmetic: the Learning Process*. New York: Van Nostrand.

E.R. Michener (1978). Understanding Understanding Mathematics. *Cognitive Science, 2*, pp 361-383.

D. Sleeman, (1982) Assessing competence in basic Algebra. In *Intelligent Tutoring Systems*, edited by D. Sleeman and J.S. Brown, Academic press, pp 186-199.

D. Sleeman & J.S. Brown, (1982) Editorial in *Intelligent Tutoring Systems*, edited by D. Sleeman and J.S. Brown, Academic Press, pp 1-12.

D.H. Sleeman, (1983a). Basic Algebra revisited: a study with 14-year-olds. *Stanford University memo HPP 83-9*. And to be published in *International Journal of Man-Machine Studies*.

D.H. Sleeman (1983b). An attempt to understand pupil's understanding of basic algebra. *Stanford Univ. memo HPP 83-11*.

R. Young & T. O'Shea, (1981). Errors in Children's Subtraction, *Cognitive Science, 5*. pp153-177.

K. VanLehn (1983). Felicity conditions for human skill acquisition: Validating an AI-based theory. *XEROX PARC tech. report CIS-21*.

REMEMBRANCE OF BLUNDERS PAST: A RETROSPECTIVE ON THE DEVELOPMENT OF PROUST[1]

Elliott Soloway
W. Lewis Johnson
Department of Computer Science
Yale University

Three years ago we set out to build an ICAI system that would help novices' learning to program, at the moment when they most need help: when they are sitting one-on-one with the beast. We confidently charged ahead and attempted to build "it". Others too have apparently adopted a similar methodological approach to this problem. While we were initially surprised by the magnitude of our failure, in retrospect we can see how the seeds of our destruction were set in motion: we had only an anecdotal sense of the bugs that real students made in real programs, and we had precious little theory to guide us.

Our confidence only a bit shaken, we set out on a more strategic course: let's see what is out there in the way of buggy and correct programs and let's develop a theory of programming that can guide the system's processing of those student programs. First off, we were stunned by the unbelievable variability that is "out there": if you look at 200 student programs--all attempting to solve the same programming assignment--you will find 200 different programs (unless, or course, there was some collusion). Coping with this variability has been a driving force in the development of our current system, PROUST.

Thus, we now develop theory, build systems, and conduct all manner of empirical studies. This approach appears to be paying off: (1) we have the makings of an empirically supported theory of the knowledge and reasoning strategies that programmers employ in the understanding of programs, and (2) we have built a system, PROUST, that can identify and correctly diagnose real student programs--albeit of a small class--at about 75% accuracy; this level of performance is about as good as human teaching assistant.

[1]This work was co-sponsored by the Personnel and Training Research Groups, Psychological Sciences Division, Office of Naval Research and the Army Research Institute for the Behavioral and Social Sciences, Contract No. N00014-82-K-0714, Contract Authority Identification Number, Nr 154-492. Approved for public release; distribution unlimited. Reproduction in whole or part is permitted for any purpose of the United States Government.

SYMPOSIUM: CONNECTIONISM VERSUS RULES: THE NATURE OF THEORY ON COGNITIVE SCIENCE

Alan Collins, Bolt, Beranek, and Newman, Chair

David Rumelhart, University of California, San Diego

Geoffrey Hinton, Carnegie-Mellon University

Zenon Pylyshyn, University of Western Ontario

Kurt vanLehn, Xerox Parc

The Emergence of Cognitive Phenomena from Sub-symbolic Processes

David E. Rumelhart
Institute for Cognitive Science
University of California, San Diego

Discussion of cognition, especially of language and thought often revolves arouind a discussion of the *rules of language* and the *rules of thought*. The former, namely the rules of language, we often call the grammar of the language – in part I suppose by analogy with the grammar rules we all learn in school. The latter, the rules of thought, we often call natural logic or just plain logic – again I suppose by analogy with the rules of logic we learn in school. These rules that people have in mind are normally expressed as relations among cognitive elements – relationships among NP's and VP's or premises and conclusions. Moreover, in many models of language and thought, these rules play an operative role, that is they are interpreted by some interpreter and decisions are made on the basis of the rules themselves. There are obvious reasons why it is tempting to think of language and thought in just this way – as a set of rules we follow while producing or interpreting what we see or hear in the case of language and as the set "of rules of inference" we employ in reasoning. Language is not haphazard. In English words *must* be used in certain orders, certain inflections *must* be used to signal certain meanings (the s for plural, the -ed for past etc.) and certain words *must* be used for certain meanings. Most importantly, most sentences are *entirely novel* – they are obviously *generated*, not stored and recalled again. On this analysis, explict rules would seem to be a very parsimonious explanation of the facts. This suggests that language learning involves the abstraction of a set of such rules and that language production and comprehension involves the application of these *general rules* to the situation at hand. Similarly, in the case of reasoning, it would appear that the most parsimonious explanation of our ability to produce and interpret arguments is that we have somehow abstracted from our experience a set of rules of inference which are then interpreted in any given situation.

As neat as these accounts have seemed, there are serious problems with them. There are characteristic flaws in our reasoning – sometimes we don't follow the rules. Similarly, language is *full* of exceptions to the rules – cases where the general rule doesn't seem to apply. This includes straight forward cases like the fact that rang is the past of

ring, not ringed as we might have thought to more interesting examples such as the fact that "cold" means something rather different in the phrase "cold person" than it does in the phrase "cold water." Within the framework of the *rule account* we must proliferate rules, differentiate between rule governed and non-rule governed cases or distinguish between competence conditions and performance conditions.

It has seemed to me for some years now that the "explicit rule" account of language and thought was *wrong*. It has seemed that there must be a *unified* account in which the so-called *rule-governed* and *exceptional* cases were dealt with by a unified underlying process -- a process which produces rule-like and rule exception behavior through the application of a single process. On this account, the rules that we analysts discover are more in our analysis than in the heads of our subjects. In short, I have come to believe that the explanation of human cognition by appealing to the interpretation of rules stated at the symbolic level is not, in general, going to work.

I have instead become very interested in a much different conception of human cognitive processing -- a system in which cognitive performance is not produced by the processing of symbolic rules, but one in which both the rule like and non-rule-like behavior is a product of the interaction of a very large number of "sub-symbolic" processes. In this sense the rule-like behavior is seen to "emerge" from these interactions rather than to have the processor in any sense "interpret" the rules at hand. This view has been motivated by two very different concerns -- on the one hand, I have been increasingly disillusioned with attempts to formulate an adequate set of explanations at the rule level. The more I learn about the way language and thought proceeds the less it seems like an application of general abstract rules. On the other hand, I have become increasingly impressed with the power of what we have come to call *Parallel Distributed Processing* (PDP) systems as an alternative to the more conventional accounts.

Parallel distributed processing is my short-hand for *brain-like* or *neurally inspired* processing systems. I am convinced that brains process information in ways fundamentally different from conventional digital computers. Whereas modern computers are capable of carrying out

serial operations in 10s of nanoseconds, brains carry out their operations in times measured in the milliseconds -- brain units seem to process information 100,000 times slower than computers!! Yet, even our best artificial intelligence systems cannot come close to matching brains on simple tasks like recognizing a spoken word or catching a ball.

What then is the brain's advantage? I suspect that this lies in the kind of computation the brain is able to carry out. Primarily, the brain succeeds because is has an enormous number of processing units all working in parallel and cooperatively *settling* into a solution -- rather than *calculating* a solution. Processing is done by cooperating coalitions of independent units each working on the information made available to it. It is as if computation were done by having each little processor carry out its small computation and then *vote* on the answer to the question. Solutions are reached by majority rule, or by reaching a compromise. There is no central processor, rather a highly distributed set of units whose combined activity pushes the whole system toward an action. This is a very different view that that implicit in the symbolic rule oriented processesing systems. There is no interpreter, there is no place in the system where the rules abid. This does not deny the existence and importantce of symbols. Symbols themselves are emergent properties of the interactions of such a set of processing units. Symbols are not, however, processed. The processing occurs at a sub-symbolic level. Regularities at the symbolic level occur and we can write descriptions of those regularities, but whenever we formulate a rule at that level we must recognize that the rule (or law) is not interpreted by the system any more that a ball flying through the air computes the differential equations which describe its behavior. It is simly a description of the system at the symbolic level.

The research program I have been carrying out in conjunction with James McClelland and several other colleagues has been to show how the cooperative interactions among many of these sub-symbolic processing units can account for the regularities which have led to the postulatation of specific rules and which, at the same time, can allow us to account for phenomena which are difficult for an explicitly rule based account. McClelland and I have produced models in two cases which I believe offer a general paradigm for this sort of information processing system. We have

shown how a simple activation model of word perception can behave as if it knows the rules of English orthography and we have shown how a simple associative memory system can mimic the aquisition of past tense verb morphology without explicitly distinguishing between regular and exception verbs. These are two examples of a large number of cases that we have been investigating. In both of these cases the system is generative, but in neither case is their an interpreter or anything that could be construed as an explicit representation of a rule.

LEARNING SEMANTIC FEATURES

Geoffrey E. Hinton
Computer Science Department
Carnegie-Mellon University
&
Terrence J. Sejnowski
Biophysics Department
The Johns Hopkins University

An important idea within cognitive science is that much general knowledge can be represented as constraints between the slot-fillers of a schema. The central idea of "connectionism" is that knowledge is represented by the strengths of the connections in a large network of simple processing elements. The relation between these two ideas is complex.

Several ways of using connectionist networks to implement schemas have been proposed. The obvious, "localist" approach is to identify processing units in the physical network with concepts, and to treat the physical links as if they were direct implementations of the pointers that are conventionally used to represent the filling of a schema-slot by an object (Feldman and Ballard, 1982; Fahlman, 1979). An alternative, "distributed" approach is to allocate a large number of units to each slot of a schema, and to represent the filler of that slot by the pattern of activity of that set of units (Hinton, 1981). The main difference is in how the physical parallelism is used. In the distributed approach, only one instantiation of a particular schema is possible at a time because the units dedicated to each slot can only have one pattern of activity at a time. The physical links are used to implement constraints between slot-fillers. By setting the strengths of the links appropriately, it is possible to make a pattern of activity in one set of units cause (or prohibit) a pattern in another set of units. If each component of a pattern of activity is viewed as a semantic feature of the object represented by that pattern, the physical links between units allow many semantic constraints to be enforced in parallel.

A major difficulty for the distributed approach is this: Someone has to choose what pattern of activity to use to represent a particular slot-filler. If a random pattern is used, it may be hard to represent the constraints between slot-fillers because the underlying semantic features are not explicit. What is needed is an intelligent choice that makes it easy to implement the constraints. If, for example, all male fillers of slot 1 are represented by patterns that have unit 253 turned on, and all male fillers of slot 2 have unit 491 turned on, then the constraint that not both fillers can be male can be implemented by making these two units inhibit each other.

Unfortunately, it is hard to discover useful semantic features automatically. The definition of a useful feature is that it puts relatively strong constraints on the features of objects in other slots, but these other

features also have to be learned and so there is a chicken-and-egg problem. This paper describes a way of learning sets of features that work well together. The learning algorithm uses some rather complicated ideas from statistical mechanics, and it runs very slowly on conventional computers, so the example given is very simple.

A very simple example

Imagine a world in which objects always occur in pairs, and only one pair occurs at a time. Each object can be paired with many but not all of the other objects. One way to characterize the structure of this world would be to simply list all the pairs and their probability of occurence. If the possible pairs were determined randomly, this method might be sensible, but if there are underlying properties of objects that influence the pairings, it will generally be much more efficient to express the probability distribution over the possible pairs by extracting these underlying features and using them to express laws of combination. Moreover, this second method will allow predictions: among the pairs that have never been observed, the ones which satisfy the laws of combination are more likely to occur than the ones which don't. The difficulty in using the second method is that the number of potential features is enormous, even if we restrict ourelves to clearcut binary features. Given n objects there are 2^n ways of picking a subset and hence 2^n potential binary features. Finding just those features which lead to good laws of combination is a formidable problem.

We use this simple example to illustrate a learning algorithm which can discover useful features. The two objects that occur together in a pair are like two slot-fillers. For each slot we have 9 units and 8 possible fillers. The different fillers are represented by turning on exactly one of the first 8 units in a slot, but we do not decide in advance whether or not the 9^{th} unit should be on. It is left to the learning algorithm to decide how to use the 9^{th} unit. The learning algorithm is therefore capable of modifying the representations that are used for the various slot fillers.

Simulations

Figure 1 shows some examples of pairs of objects drawn from a probability distribution over all possible pairs composed of one object from the set {A, B, ... H} and one from the set {S, T, ... Z}. Implicit within this probability distribution is a strong underlying regularity: If the first set is divided into the subsets {A B C D} and {E F G H} and the second set is divided into the subsets {S T U V} and {W X Y Z}, then there is a simple way of expressing the probability distribution: If the first object is in the set {A B C D} the other will be in the set {S T U V} with probability 0.9, and if the first object is in the set {E F G H} the second will be in the set {W X Y Z} with probability 0.9.

Figure 2 shows a network which has been exposed to the probability distribution by clamping the states of some of its units. States that represent a particular pair of objects are clamped with the appropriate probability. The network started with all its connection strengths equal to zero, and after being shown 5000 pairs of objects it has cap-

DT HX HZ DV CS CT EZ BT EY CU BV
EY AS CT FT BT FW GY FX GY GW BX
CV DS AU CT HY BS AT GZ AS FY EZ
HX HX CT DZ AU CV CS FX DU AY EZ . .

Figure 1: A collection of pairs of objects. These pairs were drawn from a probability distribution that can be described relatively simply (see text). The problem is to discover ways of dividing the objects into sets that allow the simple description to be expressed.

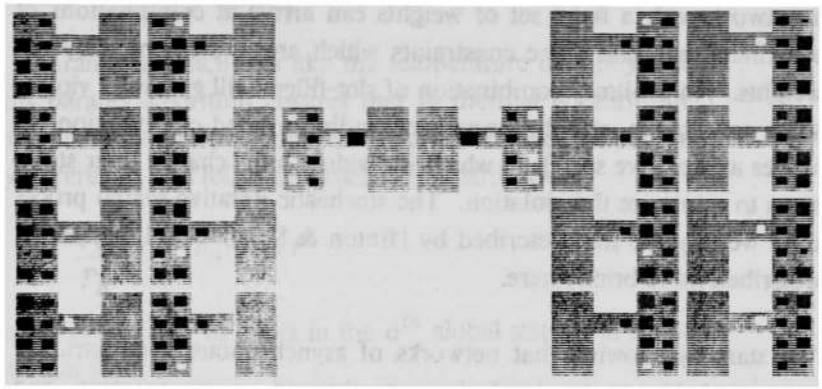


Figure 2: Each unit is represented by a gray "H" shaped region. Within this region, connections to other units are represented by white (positive weight) or black (negative weight) rectangles in the position that corresponds to the location of the other unit in the overall diagram. The size of the white or black box indicates the absolute magnitude of the weight. For example, the white rectangle in the top left hand unit represents an excitatory connection between that unit and the leftmost of the two central units. All connections between units appear twice in the diagram, once in the box for each of the two units being connected. So the white rectangle in the top left-hand corner of the leftmost central unit is the same connection as described above. Units never connect to themselves, so in the position where that connection would be displayed (e. g. the top left-hand corner of the top left-hand unit) we display the threshold using black to mean a positive threshold. The empty gray areas on the right-hand sides of the left-hand group of 8 units show that these units are not directly connected to the right-hand group of 8. Notice that the units in each group of 8 have learned to inhibit each other. This implements the within-slot constraint that only one of them should be on at a time. This constraint follows from our decision to represent each slot filler by a pattern of activity with only one of the 8 units turned on.

tured the regularity by setting its weights so that one of its two central units detects whether the first object is in the set {A B C D}, the other central unit detects whether the second object is in the set {W X Y Z}, and the two units inhibit each other.

It is hard to learn such features because there is no information to suggest them in the fillers of either slot considered separately. All 8 fillers occur equally often and have no intrinsic similarity to each other. The *only* reason for selecting these particular features is that they allow the implicit constraint between slot-fillers to be expressed.

Finding combinations of slot fillers that satisfy existing constraints

Before describing the learning algorithm it is necessary to describe how a network with a fixed set of weights can arrive at combinations of slot-fillers that satisfy the constraints which are implemented by the weights. An arbitrary combination of slot-fillers will generally violate some constraints, and the process of finding a good combination involves an iterative search in which individual units change their states so as to minimize the violation. The stochastic iterative search procedure we use was first described by Hinton & Sejnowski (1983) and is described more briefly here.

We start by showing that networks of asynchronous, symmetrically connected, binary threshold elements obey an energy function, and that repeated iterations are guaranteed to find an energy minimum (Hopfield, 1982). This minimum corresponds to a combination of slot fillers that minimizes the constraint violation. The global potential energy of the system is defined as

$$E = -\sum_{i<j} w_{ij} s_i s_j + \sum_i \theta_i s_i \qquad (1)$$

where w_{ij} is the strength of connection (synaptic weight) from the j^{th} to the i^{th} unit, s_i is a boolean truth value (0 or 1), and θ_i is a threshold.

A simple algorithm for finding a combination of truth values that is a *local* minimum is to switch each hypothesis into whichever of its two states yields the lower total energy given the current states of the other hypotheses. If hardware units make their decisions asynchronously, and if transmission times are negligible, then the system always settles into a local energy minimum. Because the connections are symmetrical, the difference between the energy of the whole system with the k^{th} hypothesis false and its energy with the k^{th} hypothesis true can be determined locally by the k^{th} unit, and is just

$$\Delta E_k = \sum_i w_{ki} s_i \quad - \theta_k \qquad (2)$$

Therefore, the rule for minimizing the energy contributed by a unit is to adopt the true state if its total input from the other units and from outside the system exceeds its threshold. This is the familiar rule for binary threshold units.

Using probabilistic decisions to escape from local minima

The deterministic algorithm suffers from the standard weakness of gradient descent methods: It gets stuck at *local* minima that are not globally optimal. This is an inevitable consequence of only allowing jumps to states of lower energy. If, however, jumps to higher energy states occasionally occur, it is possible to break out of local minima. An algorithm with this property has recently been applied to difficult constraint satisfaction problems by Kirkpatrick, Gelatt & Vecchi (1983). We adopt a form that is suitable for parallel computation: If the energy gap between the true and false states of the k^{th} unit is ΔE_k then regardless of the previous state set $s_k = 1$ with probability

$$p_k = \frac{1}{(1 + e^{-\Delta E_k / T})} \tag{3}$$

where T is a parameter which acts like the temperature of a physical system. This parallel algorithm ensures that in thermal equilibrium the relative probability of two global states is determined solely by their energy difference, and follows a Boltzmann distribution.

$$\frac{P_\alpha}{P_\beta} = e^{-(E_\alpha - E_\beta)/T} \tag{4}$$

where P_α is the probability of being in the α^{th} global state, and E_α is the energy of that state.

At low temperatures there is a strong bias in favor of states with low energy, but the time required to reach equilibrium may be long. At higher temperatures the bias is not so favorable but equilibrium is reached faster. The fastest way to reach equilibrium at a given temperature is to start with a higher temperature and gradually reduce it.

The learning algorithm

When a network is allowed to reach thermal equilibrium using the probabilistic decision rule in Eq. 3, the probability of finding it in any particular global state depends on the energy of that state (Eq 4.). These equations allow us to derive the way in which the probability of a state changes as a weight is changed:

$$\frac{\partial \ln P_\alpha}{\partial w_{ij}} = \frac{1}{T}\left[s_i^\alpha s_j^\alpha - \sum_\beta P_\beta s_i^\beta s_j^\beta \right] \tag{5}$$

where α is a global state of the network and s_i^α is the binary state of the i^{th} unit in the α^{th} global state. Eq. 5 shows that the effect of a weight on the log probability of a global state can be computed from purely local information, because it only involves the behavior of the two units that the weight connects (the second term is just the probability of finding the i^{th} and j^{th} units on together). This makes it easy to manipulate the probabilities of global states provided the desired probabilities are known (see Hinton & Sejnowski, 1983 for details).

Unfortunately, it is normally unreasonable to expect the environment

or a teacher to specify the required probabilities of entire global states of the network. A network typically contains some "visible" units that receive the input or produce the output and it also contains some other units that we call "hidden" because they are not directly involved in representing the input or output. For example, the two central units in figure 2 are hidden units and the rest are visible. The task that the network must perform is defined in terms of the states of the visible units, and so the environment or teacher only has direct access to the states of these units (hence the name visible). The difficult learning problem is to decide how to use the hidden units to help achieve the required behavior of the visible units. A learning rule which assumes that the network is told from outside how to use *all* of its units is of limited interest because it evades the main problem which is to discover appropriate representations for a given task among the hidden units.

In statistical terms, the hidden units can be used to represent the higher-order statistical regularities that are implicit in the ensemble of vectors that the environment causes in the visible units. The learning problem is to decide how best to use the capacity of the weights to capture this higher-order statistical structure. In common-sense terms, the weights should be chosen so that the hidden units represent significant semantic features and the interactions among hidden units capture the important constraints. If we make certain assumptions it is possible to derive a measure of how effectively the weights are being used, and it is also possible to show how the weights should be changed to progressively improve this measure.

We assume that the environment "clamps" a particular vector over the visible units and it keeps it there for long enough for the network to reach thermal equilibrium with this vector as a boundary condition (i.e. to "interpret" it). We also assume (unrealistically) that the there is no structure in the sequential order of the environmentally clamped vectors. This means that the complete structure of the ensemble of environmental vectors can be specified by giving the probability, $P(V_\alpha)$, of each of the 2^v vectors over the v visible units. Notice that the $P(V_\alpha)$ do not depend on the weights in the network because the environment clamps the visible units.

A particular set of weights can be said to constitute a perfect model of the structure of the environment if it leads to exactly the same probability distribution of visible vectors when the network is running freely *with no environmental input*. Because of the stochastic behavior of the units, the network will wander through a variety of states even with no input and it will therefore generate a probability distribution, $P'(V_\alpha)$, over all 2^v visible vectors. This distribution can be compared with the environmental distribution, $P(V_\alpha)$. In general, it will not be possible to exactly match the 2^v environmental probabilities using the weights among the v visible and h hidden units because there are at most 0.5 (v+h-1) (v+h) symmetrical weights and (v+h) thresholds. However, it may be possible to do very well if the environment contains regularities that can be expressed in the weights. An information theoretic measure (Kullback, 1959) of the distance between the en-

vironmental and free-running probability distributions is given by:

$$G = \sum_{\alpha} P(V_\alpha) \ln \frac{P(V_\alpha)}{P'(V_\alpha)} \tag{6}$$

where $P(V_\alpha)$ is the probability of the α^{th} state of the visible units when their states are determined by the environment, and $P'(V_\alpha)$ is the corresponding probability when the network is running freely with no environmental input.

G is never negative and is only zero if the distributions are identical. It is possible to improve the network's model of the structure of its environment by changing the weights so as to reduce G. It can be shown that:

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T}[p_{ij} - p'_{ij}] \tag{7}$$

where p_{ij} is the probability, averaged over all environmental inputs and measured at equilibrium, that the i^{th} and j^{th} units are both on when the network is being driven by the environment, and p'_{ij} is the corresponding probability when the network is free running.

One surprising feature of Eq. 7 is that it does not matter whether the weight is between two visible units, two hidden units, or one of each. The same rule applies for the gradient of G. An even more surprising fact is that the gradient involves only locally available information, even though G is a global property of the whole set of weights and the effect of one weight on G therefore depends on the current values of all the other weights. Fortunately, the other weights affect p_{ij} and p'_{ij} in just the right way to make the dependence locally available.

Parameters for the learning algorithm

The ability to discover the partial derivative of G by observing p_{ij} and p'_{ij} does not completely determine the learning algorithm. It is still necessary to decide how much to change each weight, how long to collect co-occurence statistics before changing the weight, how many weights to change at a time, and what temperature schedule to use during the annealing searches. Reasonable values for these parameters were found by trial and error. Further discussion of the effects of these parameters can be found in Hinton, Sejnowski & Ackley (1984). A "sweep" consisted of annealing 16 times with environmentally determined vectors clamped on the visible units, and 16 times with no clamping. After each sweep, each of the weights was updated with a probability of 0.5. This partially asynchronous updating helps avoid oscillations in the weights. When a weight was updated, it was always increased or decreased by the same fixed amount. The sign of the increment was determined by the sign of $p_{ij} - p'_{ij}$. The magnitude of the weight-step was 0.2.

The annealing schedule started by randomizing the state and then ran for the following times at the following temperatures: 2@2.0, 2@1.5, 2@1.2, 2@1.0, where one unit of time means running the network for long enough so that the expected number of times each unit is picked

is 1. After this annealing, the network was assumed to be at equilibrium at a temperature of 1.0, and was run for a further time of 10 while co-occurence statistics were collected.

Conclusion

One of the major problems with using distributed patterns of activity as representations is to choose the patterns. Some choices work much better than others because they make important underlying features explicit and thus they allow the physical links in the network to capture the constraints that characterize the domain. We have presented a learning algorithm for choosing representations, and shown that it can create semantic features that are useful for expressing the constraints between the fillers of two slots.

Acknowledgements

The research reported here was supported by grants from the System Development Foundation. We thank Dave Ackley, Mark Derthick, Scott Fahlman, Jay McClelland, Dave Rumelhart, and Paul Smolensky for helpful discussions.

References

Fahlman, S. E., *NETL: A system for representing and using real world knowledge.* Cambridge, Mass.: MIT Press, 1979.

Feldman, J.A., & Ballard, D.H. Connectionist models and their properties. *Cognitive Science*, 1982, *6*, 205-254.

Hinton, G. E. Implementing semantic networks in parallel hardware. In G. E. Hinton & J. A. Anderson (Eds.) *Parallel Models of Associative Memory.* Hillsdale, NJ: Lawrence Erlbaum Associates, 1981, 161-187.

Hinton, G.E., & Sejnowski, T.J. Analyzing cooperative computation. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society.* Rochester, NY, May 1983.

Hinton, G.E., Sejnowski, T.J., & Ackley, D. H. Boltzmann Machines: Constraint satisfaction networks that learn. Technical report CMU-CS-84-119, Carnegie-Mellon University, May 1984.

Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA*, 1982, *79* pp 2554-2558.

Kirkpatrick, S. Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science*, 220, 671-680.

Kullback, S. *Information Theory and Statistics.* New York: Wiley, 1959.

WHY "COMPUTING" REQUIRES SYMBOLS

Zenon Pylyshyn
University of Western Ontario

For some years now I have been advocating the view that to understand what is essential about cognition as computing it is mandatory that we preserve a number of distinctions. I have discussed several of these distinctions in my book (Pylyshyn, 1984). For the present purpose I wish to examine one of these distinctions: that between a machine and the symbolically encoded "rules and representations" that the machine uses. (It doesn°t matter here whether by "the machine" one means the device described in the manufacturer°s manual, or what is sometimes called the "virtual machine" consisting of the raw machine plus an interpreter for some higher level programming language. This is just a conceptual distinction in any case since the virtual machine is no less a real physical machine than the one delivered from the manufacturer, only with a different initial state.) Since the distinction between the machine and the symbol structures is one of those distinctions that some people have been trying to do away with (cf., Anderson and Hinton, 1981), I will review one of the fundamental reasons why I believe that the task of providing explanations in cognitive psychology cannot be carried out successfully without it.

The difference between a very complicated device that goes through distinguishable states (but is not characterized as processing symbols) and what I would call a computer in the strict sense (as well as in the usual computer science sense) is exactly the difference between a Turing Machine and any arbitrarily complicated finite state automaton, network, or "connectionist" machine. The main difference, from our perspective, is not that the Turing Machine°s tape is unbounded (though that does have consequences whose relevance to cognitive science is not clear), but that when we do not impose a bound as part of the definition of the machine itself we force a certain kind of qualitative organization on the system. In particular it forces us to distinguish between a strictly finite mechanism (the Turing machine°s finite state "control box") and a finite but unbounded string of symbols. If it were not for that distinction it would not be possible to have a Universal Turing machine. The finite characterization of machines that such a distinction gives us is crucial. Turing machines are individuated by their finite part -- that°s what allows them to be enumerated. A finite part is similarly required for proof theory (the axioms and rules of inference have to be finitely specified).

It is important to see that what is at stake here is the nature of the organization captured in a certain description. An ordinary Von Neumann style computer can clearly be characterized as a finite state automaton. It can also be given a true description at the circuit level. But it°s only when it is described as processing symbols (and in fact only when it°s viewed as processing the particular symbols that are semantically interpreted) that we can explain its input-output behavior in such a way as to capture those regularities that are invariant over certain implementation differences. And what°s even more to the point, it°s only when we describe it at the symbol level that we can explain what it°s doing in semantic terms (e.g., in terms of doing arithmetic, or playing chess, or carrying out inferences, or whatever else the device may be correctly described as

doing). That (at least some) human reasoning (e.g., doing arithmetic, deciding what to have for dinner, planning a trip, deciding on the intended referent of an anaphoric expression, etc.) is correctly characterized in terms of such rules cannot be in dispute. The only arguable point has been whether the behavior described by such rules can be realized by a system that works according to some principles that do not reflect the structure of these rules.

Consider a simple example. A semantically interpretable procedure such as one for adding two numbers cannot be adequately described in terms of state-transition diagrams, such as those used in the description of finite state automata. The reason is that the general rule for adding numbers cannot be finitely stated as a rule for producing a transition from state Sn to state Sn+1 in a computer. Rather, it must be stated as a rule (or a set of rules) for transforming an expression of numerals into a new expression. An interpreted rule, such as the rule for addition, applies to states which have a particular semantic interpretation (say, as certain numbers).

Of course changes in the machine°s state are the result of physical, not number-theoretic, causes. Consequently the way the machine must work in order to be correctly described as following an interpreted (e.g., mathematical) rule, is that on every occasion in which the rule is invoked there must be physical properties of the machine°s state that are capable of serving as physical codes for that semantic interpretation. In other words, for each distinct rule-relevant semantic property there must be a corresponding distinct physical property associated with that state: distinct semantic properties must be preserved by some distinct physical properties -- and in fact they must be the very same physical properties that cause the machine to behave as it does on that occasion. Such articulation of the states into distinct properties must, furthermore, correspond to the articulation of the semantic rule in terms of symbolic expressions. In other words, the articulation of the states must be made explicit if we are to both express the rules that govern the computation and at the same time show how, in principle, such rules might be realized in a physical system.

There are all standard ideas. What they come down to is that in order to finitely express some computational regularity, such as that captured by a mathematical (or other) rule, we have to refer to a structure of symbols, and the structure of the expressions must be preserved by the structure of the states of the system. A characterization of the machinery that does not articulate the states of the system in this way cannot explain how the system can exhibit regularities expressed in the form of such rules as rules of inference. Thus if human behavior can be correctly described as following rules -- if capturing important regularities requires such a formulation -- then it appears that this has implications for the nature of the system that realizes such behavior.

People who object to the conventional view of computation as symbol processing frequently have in mind the implausibility of the mind working like a VAX. I have much sympathy for that view, as I keep saying: that°s why it°s so important in cognitive science to find out what the functional architecture of the mind is. I would not be the least surprised to find that it is so very different from a Von Neumann machine that it may scarcely

be recognizeable as a computer by examining its command set. It will, no doubt have massive parallelism. Many people think that having a lot of parallelism will make a fundamental difference to what we count as computing. But the issue is not whether the mind is a serial or a highly parallel computer. The issue is whether it processes symbols: whether it has rules and representations. A highly parallel system can process symbols in at least two ways. One is that it may be parallel only in the way it implements its primitive functions, i.e., the functional architecture may be neurally implemented in a highly parallel way. But, of course, that much is true of the Von Neumann computer. Its random access memory mechanism requires a great deal of simultaneous activity in every part of the memory. The other way that it may be parallel is that the primitive operations need not form a total ordering in time. But even in an architecture as radically nonlinear as one based on populations of ACTORS, there is no conflict with the sense of computing that I claim must be going on in the mind, so long as each actor processes semantically interpreted symbols, as opposed to just sending activations that have no semantic interpretations in the domain of our perceptions, thoughts, and the like.

The point of all this is to suggest that so long as cognition (human or otherwise) involves semantic regularities, such as knowledge based decisions and inferences, and so long as we view it as computing in any sense, we will need to view it as computing over symbols. No connectionist device, however complex, will do. Nor will any analog computer, but that is a topic for another occasion (for example, see Pylyshyn, 1984).

References

Anderson, J.A., and Hinton, G.E. "Models of Information Processing in the Brain," in G.E. Hinton & J.A. Anderson (Ed) Parallel Models of Information Processing in the Brain. Hillsdale, NJ: Erlbaum, 1981.

Pylyshyn, Z.W. Computation and Cognition: Toward a Foundation for Cognitive Science. Cambridge, Mass: M.I.T. Press (Bradford), 1984.

A critique of the connectionist hypothesis that recognition uses templates, and not rules

Kurt VanLehn*

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

Connectionist models of cognition feature a network of nodes, whose topology is assumed to be relatively permanent. Computation (i.e., thinking) is represented by fluctuations of the activation levels of nodes and by transmission of excitation and inhibition along connections. More elaborate formulations equip nodes with small state registers instead of activations, and connections pass small messages instead of an excitatory or inhibitory quantities. The main architectural principles are (1) information transmission along connections happens in parallel, (2) there is little, if any, global control (i.e., no central processor), and most importantly, (3) a cognitive model may use as many nodes and connections as it needs, but there are severe limitations on the amount of information stored in nodes or transmitted by connections.

Historically, connectionism is analogous to the production system movement. Both schools are revisions of earlier movements. Both schools rose to recent prominence in psychology when extraordinarily good pieces of research were done within their respective paradigms. Newell and Simon's (1972) study of problem solving kicked off the production system movement. Studies by Rumelhart and his colleagues of reading, typing and speech kicked off connectionism (McClelland & Rumelhart, 1981; Rumelhart & McClelland, 1982; Rumelhart & Norman, 1982; Elman & McClelland, 1983). Both production systems and connection systems have attracted the help of computer scientists interested in them for non-psychological reasons. Connection architectures like Fahlman's NETL or Hinton's Boltzmann Machine (Fahlman, Hinton & Sejnowski, 1983) are the analogs of production system languages like OPS and ACT. Unlike production systems, connection systems have attracted hardware designers who are building massively parallel computers for rapid execution of connection systems. Connection systems are as hot today, or even hotter, than production systems were a decade ago.

If the analogy between production systems and connection systems can be trusted, psychology will soon enjoy the fruit of a new formalism. It is good to have a wealth of technical notations and distinctions. Although today's cognitive scientist may not like production systems, she or he still knows what the left-hand side of a rule is, and how important conflict resolutions strategies are. Such widely-shared conceptual tools enrich and empower the field by making it easier to communicate complex ideas. Perhaps they even make it easier to generate those ideas in the first place.

---

* This work was supported by the Personnel and Training Research Programs, Psychological Sciences Division, Office of Naval Research, under contract number N00014-82C-0067, contract authority number NR 667-477. Thanks to David Christman, Danny Bobrow and Johan de Kleer.

However, the analogy issues a warning as well as heralding a benefit. Although the research that kicked off the production system movement was outstanding psychology, some later works claimed psychological validity *solely because they used production systems to express their models.* Since one can easily express absurd cognitive models in production systems, psychologists must do much more than notate their models as a production system before they are entitled to even suggest that the model is psychologically plausible. I sincerely hope that this methodological error will not plague connectionism. A connectionist model is not a psychological plausible model just because it uses the same connection system that, say, McClelland and Rumelhart used. Even if one wired up the model with squid neurons, there would be no reason to believe it had anything to do with the mind. One can write rubbish in any representation language. It takes hard work to uncover the principles that are fundamental to a particular model, and even more hard work to show that those principles are psychologically valid. This extra work, which is over and beyond the work needed to implement the model as a connection or production system, is just exactly what yields theory (VanLehn, Brown & Greeno, 1982). Without it, one has just another program that behaves with an amusing, superficial similarity to humans. It has no more scientific merit than the "robots" hired by shopping malls.

Enough methodology! Let's move on to substantive psychological issues.

Connectionism makes an important hypothesis: For some tasks, the best models are those that achieve a rule-like behavior without rules by using a large, finite store of templates. Perhaps the most impressive demonstration of this hypothesis is Rumelhart and McClelland's interactive activation model of word recognition. It has a store of the 1179 most common four-letter English words, and it has no orthographic or phonological rules. Yet it accounts for a host of rule-like human behavior.

The experimental task goes as follows: The subject is shown a four letter string for a short time, then it is replaced by a mask (e.g., a string of "#" signs). The subject is tested on a single letter in the stimulus, using a forced choice between two letters. The subject guesses which letter occurred in that position. Three main effects are observed in such experiments. *Word advantage:* When the stimuli are English words, the subjects' guesses are correct about 17% more often than they are if the stimuli are non-words such as QXRL or ACUU. *Pseudoword advantage:* When the stimuli are pseudowords (i.e., orthographically regular, such as MAVE or SPET, but not English

words), the subjects' guesses are correct about 15% more often than they are with non-words. *Wordlike consonant strings advantage:* When the stimuli are consonant strings (and hence orthographically and phonologically irregular) that are constructed by replacing a word's vowel with a consonant (e.g., SPAT becomes SPCT), then subjects' guesses are about 15% more accurate than with non-word stimuli. This third finding tends to refute any theory of word recognition based on stored orthographic or phonological rules.

To account for these three findings, the interactive activation model stores all common four-letter English words. This is the key feature. One can get adequate empirical accuracy, I contend, without a connection system as long as there is a word store and it is used in certain ways. That is, the credit for explaining the main effects belongs to the hypothesis that people recognize words with templates instead of rules. The success of the explanation does not depend on the representation language, which is good. To demonstrate this point, a simplified version of the McClelland/Rumelhart model is presented. Let the function Friends(S,L,P,I) return the set of all words in the store that share I letters with the stimulus S and have letter L at position P. Of course, $1 \leq I \leq 4$ and $1 \leq P \leq 4$. If the stimulus were the pseudoword "MAVE", then

Friends("MAVE","H",1,3) = {"HAVE"}
Friends("MAVE","M",1,4) = {}
Friends("MAVE","A",2,3) = {"HAVE","SAVE","MALE",...}

Given this function to access the word store, the percentage of correct guesses is predicted using the following formuli:

$$\text{Activation}(S,L,P) = \sum_i a_i |\text{Friends}(S,L,P,i)|$$

$$\%\text{Correct}(S,P) = \frac{\text{Activation}(S,P,R)}{\text{Activation}(S,P,R) + \text{Activation}(S,P,W)}$$

where R is the right letter choice, W is the wrong one, the a_i are task parameters, and |X| is the cardinality of set X.

Let's see how this model behaves with each of the stimuli kinds. If the stimulus S is a non-word, then both R and W will have few 2-, 3- or 4-letter friends (i.e., Friends(S,R,P,I) $\approx$ Friends(S,W,P,I) $\approx$ {} for all $I \neq 1$, for all P). They will both have many 1-letter friends. So Activation(S,P,R) $\approx$ Activation(S,P,W), and %Correct is roughly 50%. If S is a word, then R will have exactly one 4-letter friend (i.e., S) and W won't have any. Because W is chosen by the experimenter so that it forms a word when substituted into S, W has exactly one 3-letter friend. On the other hand, R usually has many 3-letter friends. Similarly, R will generally have more 2-letter friends than W. Since |Friends(S,R,P,I)| > |Friends(S,W,P,I)| for all $I>1$, Activation(S,R,P) > Activation(S,W,P), and hence %Correct is greater than 50%. The case for pseudowords and wordlike consonant strings is just like the case for words, except that R will have no 4-letter friends. Hence, the %Correct will be a tad lower than the %Correct for words, but it is still greater than the 50% correct of nonwords. These predictions are qualitatively similar to the main findings. To get quantitative accuracy would require fitting the a_i parameters. Parameter a_4 controls the relative advantage of words over pseudowords and wordlike consonants. Parameters a_3 and a_2 control the advantage of

pseudowords and wordlike consonants over nonwords. Interestingly, if subjects are not instructed to expect pseudoword stimuli, then the pseudoword advantage disappears. Under these conditions, $a_3 = a_2 = 0$.

The above model is a simplification of the one actually used by Rumelhart and McClelland. To compete with theirs, it would need an input/output model wrapped around it in order to account for phenomena involving the duration and image quality of the stimuli, the kind of masking, serial position effects, and so on. The interactive activation model can explain some of these effects, although some extra, non-connectionist mechanisms must added (e.g., a clock and a gated response buffer) in order to do so. Extensibility of a model is important, but it is not as important as accounting for the main effects. I take it that Rumelhart and McClelland have convincingly demonstrated that the main findings are best explained by the hypothesis of word storage rather orthographic or phonological rule storage. Moreover, it doesn't matter whether one expresses the hypothesis with connections, as in the interactive activation model, or with a simple additive model, as above.

To return to the production/connection analogy, this "templates, not rules" hypothesis is analogous to Newell's problem space hypothesis (Newell, 1980). Neither of the two hypotheses mention the representation language (which is good), both are quite general (which is excellent), and both can be tested in specific cases (which is best of all). As it turns out, both are controversial, and that's good too. Contention over hypotheses like these will advance cognitive science, but fights about connections versus rules are mere religious squabbles.

In the interest of controversy, I'll try to indicate some problems that the templates-not-rules hypothesis might have in accounting for other kinds of recognition than word recognition. I undertake this with some reluctance. I prefer to evaluate specific hypotheses against specific empirical facts. However, I can find no problems with the way that the templates-not-rules hypothesis accounts for the word recognition phenomena, so I have no alternative but to attack it with general observations.

First, a few quick shots. Any straightforward realization of the templates-not-rules hypothesis means that the recognizer outputs templates from its finite store. A word recognizer outputs words. A word sense disambiguator outputs word senses. But there are plenty of cognitive domains where there isn't a finite set of classes to recognize. Take natural language understanding. If there were a finite number of things that an utterance could mean, then they could be the templates in the store, and the interactive activation model might be a perfectly adequate explanation of natural language understanding. But clearly there are not a finite number of meanings. It's likely that meanings, and maybe even utterances as well, are not countable. Where does one stop and another begin? This suggests that the templates-not-rules hypothesis has significant trouble with domains that don't admit of finite classifications.

Here's another quick shot. A big advantage of template systems is that new knowledge is easy to acquire. To learn a new word, one just adds it to the word store. That's a bit too simple, of course, because it is very rare for a recognition stimulus to *exactly* match any of the training instances. There is a certain amount of abstraction,

differentiation and noise removal that has to go on in getting from a training example to a recognition stimulus. A rule-based system does most of this work during learning. A system that stores training instances does most of the work during recognition. Which do people do? A general observation about human behavior is that learning something is much harder than recognizing it. It might take several seconds of rehearsal to add a new word to one's vocabulary, but once that word is learned, recognition takes mere milliseconds. This observation tends to refute the templates-not-rules hypothesis.

I'll finish with a longer criticism of the hypothesis, which may ultimately be more interesting to computer scientists than to psychologists. When Elman and McClelland (1983) applied the interactive activation model to speech perception, they had a little problem, which turns out to be symptomatic of a nearly fatal flaw in the connectionist approach. They assumed that the template store held words, represented as sequences of phonemes. They assumed that the input (stimulus) was a mixture of phonemes, where the strength of each phoneme varied with each tick of the clock. (Actually, they used phonological features as input, but that's irrelevant to the current account.) The model had two difficulties. (1) It depended too much on finding a clear occurrence of the initial phoneme of a word in the input stream. (2) The model had difficulty recognizing words spoken more slowly or more rapidly than usual. Both difficulties can be traced to the same underlying problem: the word's phonemes must be brought into registration with the phonemes in the input sequence. For the word recognition task described above, registration is not a problem. There are always exactly four stimulus letters, which can be matched in only one way against the stored four-letter words. If instead the stimulus were, say, a 14 letter string with a four letter word buried somewhere inside it, then there would be 10 possible ways to match the input with a stored word. This would be a 10-way ambiguous registration problem. The speech recognition task has this registration problem, and more. It has a second source of ambiguity because the phonemes are not guaranteed to be certain cannonical lengths, nor must they be adjacent. If a word is spoken slowly, noises may intervene. These are not problems for just the Elman/McClelland model. They are inherent in the speech recognition task. All speech recognizers must deal with them.

In fact, all recognizers of any kind must deal with the registration problem. In vision, for instance, it is not enough to store an image of an object. The system must be able to recognize the object under translation, rotation, scaling and possibly other kinds of transformation. Perhaps the clearest cases of the registration problem occur in non-metric tasks, such as story recognition. Suppose a stimulus story has N actors, and an old, stored story has M actors. There are N^M possible ways to map stimulus actors to the stored actors. The number of possible registrations decreases if one adds the constraint that two stimulus actors can't fill the same role in the stored story. The number of registrations decreases even more in tasks with more constrained topologies, such as reading, speech or vision. As illustrated a moment ago, there are only $N - M$ registrations to check in order to find an M-long word in an N-long stimulus string.

Not only is the registration problem universal, its intransigent. The typical recognition procedure for a serial machine has two nested loops:

```
(For each template in the store do
   (For each legal registration of the parts of the template
      with the parts of the stimulus do
      (... some matching of the template to the stimulus
         under the mapping of the registration ...)))
```

Connection machines basically eliminate the outer loop by doing all instantiations of its body in parallel. That is, each template is associated with a distinct group of nodes. The stimulus is broadcast all at once to each group, which then trys to recognize its template in the stimulus. Each node group has the connectionist equivalent of the inner loop, which does registration. If the registration problem is trivial, then the node group can be small. For instance, each group consists of a single node in the Rumelhart/McClelland model because there is only one way to register a four-letter stimulus against a four-letter template. On the other hand, if the registration problem is complex, then node groups are large. For instance, the general case of registering an N-part stimulus to an M-part template could use a tree of nodes that is M levels deep with a uniform branching factor of N.* There are $O(N^M)$ nodes per node group. Replacing the inner loop by a parallel scheme merely replaces time complexity by space complexity. The registration problem is inherently complex as well as universal.

A common approach to coping with the registration problem is to reduce M, the number of parts in the template to be matched. Instead of M parts, a template has J subtemplates as its parts where $J \ll M$, and each subtemplate has its J parts, which in turn may have parts, and so on. An old, flat template becomes a part-whole hierarchy. As it stands, this doesn't reduce the combinatorics of registration. Consider a flat template with 6 parts. When it matches an input with N objects, their are N^6 registrations to check. Suppose the new template has two parts, A and B, each of which is a three-part subtemplate. There are N^3 possible matches for A, and N^3 matches for B. If there are no constraints on A and B that are independent of the main template, then the main template has to check all N^3 bindings for A against all N^3 bindings for B. Since $N^3N^3 = N^6$, the registration problem in no easier. Simply dividing flat templates into subtemplates doesn't reduce the combinatorics at all. Combinatorics only begin to decrease when constraints can be added at the subtemplate level. Sharing subtemplates among templates also helps.

---

* The details: Each node has N descendents. The root node passes a message to its first daughter that pairs the first template part to the first of the N stimulus parts. The second daughter gets a message pairing the first template part to the second stimulus part, and so on for all N daughters. So the first level (= root node) takes care of pairing the first template part to each of the N stimulus parts. The second level (= the daughters of the root node) pairs off the second template part in similar fashion. In order to bind all M template parts, the tree has M levels. So it has $(N^{M+1} - 1)/(N - 1)$ nodes.

Here is a template-subtemplate hierarchy that shares subtemplates and has lots of constraints among subtemplates. I think you will find the notation familiar.

S → NP VP
NP → Determiner NBAR
NBAR → Adjective* Noun
VP → Auxiliary VBAR
VBAR → Verb (NP) (NP) PP*
PP → Preposition NP

A templates-not-rules recognizer that "optimizes" its performance by adopting a part-whole hierarchy with constraints is no longer a templates-not-rules system. It is a rule-based system.

The conclusion is that templates-not-rules systems are infeasible for any recognition problems that require non-trivial, non-metric registration. Neither serial computers nor connection machines can run them fast enough. On the other hand, rule-based systems are feasible.

A psychologist can draw one of two conclusions from this. Either, (1) people are subject to the same "laws of information processing" as machines, therefore they must use rule-based recognizers, and therefore the templates-not-rules hypothesis is generally false, or (2) people have templates-not-rules recognizers, but they run them (so to speak) on some as yet undiscovered information processing architecture that somehow solves registration problems very quickly.

References

Elman, J.L. & McClelland, J.L. Speech perception as a cognitive process: The interactive activation model. To appear in N.Lass (Ed.) *Speech and Language, Vol. 10.* New York: Academic Press. Currently available as ICS Report No 8302, University of California, San Diego, 1983.

Fahlman, S.E., Hinton, G.E. & Sejnowski, T.J. Massively parallel architectures for AI: NETL, Thistle, and Boltzmann Machines. In *Proceedings of 1984 AAAI Conference*, Los Altos, CA: Kaufman, 1983.

McClelland, J.L. & Rumelhart, D.E. An interactive activation model of context effects in letter perception: Partl. An account of basic findings. *Psychological Review*, 1981, *88*, 375–407.

Newell, A. Reasoning, problem solving and decision processes: The problem space as a fundamental category. In R. Nickerson (Ed.) *Attention and Performance VIII*, New York: Erlbaum, 1980.

Newell A. & Simon H.A., *Human Problem Solving*, Englewood Cliffs, New Jersey: Prentice Hall, 1972.

Rumelhart, D.E. & McClelland, J.L. An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extensions of the model. *Psychological Review*, *89*, 60–94.

Rumelhart, D.E. & Norman, D.A. Simulating a skilled typist; A study of skilled cognitive-motor performance. *Cognitive Science*, 1982, *6*, 1–36.

VanLehn, K., Brown, J.S. & Greeno, J.G. Competitive argumentation in computational theories of cognition. In W. Kinsch, J. Miller & P. Polson (Eds.) *Methods and Tactics in Cognitive Science.* New York: Erlbaum, forthcoming. Currently available as Xerox Parc report CIS-14, Palo Alto, CA, 1982.

SYMPOSIUM: CROSS-DISCIPLINARY APPROACHES TO LANGUAGE PROCESSING

Morton Ann Gernsbacher, University of Oregon, Chair

Talmy Givon, University of Oregon

Wendy Kellogg, University of Oregon

Penny Yee, University of Oregon

Frances J. Friedrich, VA Medical Center

Russell S. Tomlin, University of Oregon

Peter W. Jusczyk, University of Oregon

Sixth Annual Conference of the Cognitive Science Society
Boulder, Colorado
June 28-30, 1984

SYMPOSIUM: Cross-Disciplinary Approaches to Language Processing

Chair: Morton Ann Gernsbacher, Department of Psychology and Cognitive Science Program, University of Oregon

This symposium presents work recently completed and still in progress, conducted within the Cognitive Science Program at the University of Oregon and the Cognitive Neuropsychology Laboratory at Good Samaritan Hospital, Portland, Oregon. This body of work reflects a collective interest in a variety of issues related to identifying and pursuing areas of fruitful interaction between linguistics, cognitive psychology and cognitive neuropsychology. The focus of the research is the cognitive operations underlying language processing, and their relationship to syntactic devices proposed by various linguists.

Givon, Kellogg, Posner and Yee will describe their use of mental chronometrics for testing the psychological reality of a linguistic theory of referentiality. Friedrich will discuss what the semantic and syntactic processing capabilities of a conduction aphasic with a phonological coding deficit suggest about normal sentence processing. Gernsbacher will describe some cognitive responses (attentional, integrative and memorial) that occur after a linguistic topic change. Tomlin will describe his use of laboratory manipulations (priming and cueing) to investigate the role of attention in thematic organization and its accompanying syntactic manifestations. Following the four paper presentations, Jusczyk will comment on some of the benefits and caveats of cross-disciplinary approaches to language, and lead an open discussion of the work presented.

The cross-disciplinary efforts represented by these studies yield at least three benefits to an understanding of language and language processing.

First, each discipline offers novel methodological tools to the others, opening up new avenues for empirical investigations of linguistic problems. Second, such efforts help theoretically to develop theories of language by considering how cognitive processes influence and constrain linguistic performance. And, third, cross disciplinary efforts prove intellectually stimulating and provide a crucial opportunity to reconsider discipline dependent thinking about language processing and cognition.

Time dynamics of referential processes in discourse

T. Givon, Wendy Kellogg, Michael I. Posner, Penny Yee
Depts. of Linguistics and Psychology and Cognitive Science Program
University of Oregon, Eugene

Thoughts exhibit continuity over time. It is the ability to integrate separate elements into a coherent theme that is an essential aspect of cognition. Yet it has proven difficult to find experimental methods to investigate the processes providing the basis of such continuity. One of the attractive parts of the study of discourse is that discourse is defined in terms of the existence of a coherent theme that allows us to relate varied elements. In order to continue a theme over time it is important to refer to concepts that have been mentioned previously but are not now present. Language provides a number of specific devices for referring the listener or reader back in time to the concepts that are relevant to the current information. Pronouns, repetitions of a noun phrase used previously, or closely related noun phrases that serve to reinstate the previous concept are all devices that can be used to reestablish in the reader's mind a previous topic.

Givon has proposed that languages use differing syntactic devices to signal the reader or listener how far back the information which is being referenced occurs. If a topic has been absent for some time, a noun phrase will be used to refer to it. Since readers or listeners never know how long the discourse will be, it appears likely that they retrieve the topic on-line while they continue to listen to or read new information. It would be useful to be able to observe the time dynamics of the processes which are used to reinstate such topics.

The present studies employed chronometric methods to examine the time dynamics of the processes used to reinstate concepts central to the discourse topic. Subjects pressed a bar to receive each successive word of discourse while monitoring the CRT screen for a visual probe, to which a speeded detection response was required. Dependent measures were the time to respond to the probe and the bar-press time on a referring device and its surrounding words. These measures were examined in two experiments as a function of the degree of continuity of the discourse and the type of referring device (noun phrase or pronoun).

Results showed differences in maximal interference for noun phrases vs. pronouns. Pronouns show an earlier interference effect than noun phrases. A careful examination of the time ocurse data from Experiments I and II also reveals some evidence in favor of prolonged processing in cases where the syntactic device is not appropriate. When a pronoun is used to reinstate a topic which has been absent for several clauses, interference is prolonged relative to the case where a pronoun refers to a continuously present topic. In contrast, interference for noun phrases does not differ as a function of the referential distance. The results show that on-line psychological methods are sensitive to differences in reference between noun phrases and pronouns, and may provide data useful for evaluating linguistic theories of reference.

Consequences of a phonological coding deficit on sentence processing

Frances J. Friedrich
VA Medical Center
Fresno, California

It is often difficult to determine the role of specific cognitive codes or processes in complex language skills, such as sentence comprehension, because of the number of operations and the flexibility that individuals display in combining these operations. Neuropsychological data from patients with brain injuries can be useful in constructing models of normal language processing by demonstrating how language breaks down when a specific code or operation is impaired. The work reported here examined the sentence processing abilities of a patient with a specific impairment in phonological coding. The pattern of abilities and deficits that emerges can be used to clarify the role that the phonological code plays in normal sentence processing.

The data were collected from E.A., a conduction aphasic with the primary repetition disorder and the good spontaneous speech and comprehension that characterize the syndrome. E.A.'s digit span was severely impaired (auditory = 1.5 items; visual = 2.4 items), but various tests suggested that the memory deficit was secondary to an inability to represent information in a phonological code. The main concern in this study, however, was with the consequences of a phonological code/short-term memory impairment on sentence comprehension and production. Comprehension tests of reversible active and passive sentences revealed that E.A. made few errors on the active sentences (5% overall) but frequently made subject-object reversal errors for the passive sentences with both auditory and visual presentation (33% and 19%, respectively). Thus, she appeared to use an S-V-O mapping strategy when assigning grammatical roles to the constituent noun phrases.

Production tests, including story completion and picture description tasks, demonstrated that E.A. was able to generate a variety of sentence constructions and was generally sensitive to the contextual cues that dictate what constructions are appropriate, although she did show some tendency to produce sentences in active voice. Many of her errors, however, involved the omission of grammatical morphemes from her written production and did not occur in spontaneous speech or in oral responses to the production task.

Overall, the data collected from E.A. help define the role of the phonological code and short-term memory in normal language processing. Specifically, these data suggest that a phonological code is the primary means by which important syntactic markers are represented and that the ability of intact adults to recompute grammatical roles when the S-V-O mapping is violated may be heavily dependent on a phonological representation.

Cognitive responses to (linguistic) topic change in discourse

Morton Ann Gernsbacher
Department of Psychology and Cognitive Science Program
University of Oregon, Eugene

Imagine the following scenario: I am at a conference having a conversation with one of my colleagues. We are discussing some of the papers we heard during the day's meeting. In the middle of this discussion, I nonchalantly mention: "Watermelons grow well in the South." This utterance would lead to several consequences. With the exception of my colleague concluding that I was a bit odd, most of these consequences would be at his or her expense.

Subjectively, the feeling one has when the discourse topic changes abruptly is something like a mental double-take. We stop for a puzzled moment. Mentally we question the speaker or author: Wait a minute, weren't you just talking about Cognitive Science? What in the world does that have to do with watermelons? However, this overt response occurs only to the most blatant topic changes. Those that occur in natural discourse are much more subtle. I suggest that what occurs there is a covert response, a cognitive Processing Shift.

The notion of processing shifts is drawn from a framework recently sketched (Gernsbacher, 1984). According to this framework, the goal of comprehension is to build up a coherent mental representation of the complete stimulus and this representation is built in the following way. During comprehension memory "cells" are activated by incoming information. (Memory cells contain previously stored mental representations or traces). Initial activation of memory cells and their transmission of processing information (enhancement and suppression) lays a foundation. Once laid, congruent (similar or related) stimulus information simply adds on to the developing structure. This is because the more overlapping the incoming information is with that previously received, the more likely it will be to activate the same or connected cells. However, the less congruent the incoming information is, the less likely it will be to activate the same or connected cells; hence, the less likely it will be to build on to the developing structure. In this case, a different set of cells will be activated. Because this second set of cells has not been recently activated, a relatively new foundation begins to be laid. This shift from actively building one structure, really a sub-structure, to initiating another is a processing shift.

Presumably processing shifts should be manifested in cognitive behavior. Likely consequences are that processing shifts consume attention, retard integration, and reduce memorial retention (at least of the pre-topic change material). There is some evidence to substantiate each of these assumed responses. But this evidence has been collected from comprehension situations where the topic changes are of the gross semantic type -- similar to my introductory example. I was interested in whether indeed we are also sensitive to more subtle indicants of topic change. In particular, I wanted to know what cognitive responses might result from processing linguistic devices that signal topic change. Three such devices are: Renomalization, Object-Subject Conversion, and Adverbial Leads. These devices are best described by example; moreover, illustrating them will give me a chance to explain some of the study's methodological nuts and bolts.

I constructed several pairs of seven-sentence paragraphs. Each pair differed only by their fourth and fifth sentence. In one member of a pair, the linguistic device appeared in the fourth sentence, while in its mate paragraph, the device appeared in the fifth sentence. I will refer to these members of a pair as versions A and B, respectively. So, for example, a pair of paragraphs using Renomalization began with these three sentences: (1) *Joe was studying hard for his math test.*

(2) *He had read two chapters in the book.* (3) *He was now trying to work the problems.* In version A, the fourth sentence restated the main character's name; (4A) *Joe could tell he would be up all night.* In version B, this renomalization did not occur in the fourth sentence: (4B) *He could tell he would be up all night.* Rather, it occurred in the fifth sentence: (5B) *Joe knew he should have begun earlier.* In contrast, the fifth sentence of version A resumed pronomalization; (5A) *He knew he should have begun earlier.* Both pairs ended identically; (6) *But earlier he had a big softball game.* (7) *And he much preferred softball to math.*

A pair of paragraphs using Object-Subject conversion began with these three sentences: (1) *One afternoon Beth was feeling bored.* (2) *She was tired of watching television.* (3) *She picked up the phone and called Jack.* Then, in version A, the object of the previous sentence was converted to the subject; (4A) *Jack asked her what she had been doing.* In version B, it was not; (4B) *She asked Jack what he had been doing.* Rather, the object-subject conversion appeared in the fifth sentence: (5B) *Jack really didn't have much to say.* The fifth sentence in version A maintained the conversion: (5A) *He really didn't have too much to say.* And again the two pairs ended identically; (6) *He just told her about his boring day.* (7) *All he'd been doing was watching T.V.*

Finally, a pair of paragraphs using an Adverbial Lead began with these three sentences: (1) *The man was sitting on his front porch.* (2) *He was alone and reading a newspaper.* (3) *He was enjoying the peace and quiet.* The fourth sentence of version A began with an adverb: (4A) *Suddenly, he heard someone call his name.* In version B, it did not: (4B) *He heard someone start calling out his name.* Rather, the adverb fronting occurred in the fifth sentence: (5B) *Suddenly, he ran inside to find out why.* And again, the two versions ended identically: (6) *He found his wife trying to open a jar.* (7) *It appeared that she needed his help.* (Note also that despite these manipulations, the critical fourth and fifth sentences always contained approximately the same number of characters in the A or B version).

In each of these pairs, comparing the cognitive responses occurring after the critical fourth vs. fifth sentences should indicate whether the manipulated linguistic device effected a processing shift (presuming that processing shifts are indeed manifested into cognitive behaviors). In a first experiment, I looked at what might be considered attentional or integrative responses. Subjects were presented with these paragraphs to read line-by-line. Their self-paced reading times were recorded. Presumably, the more attention demanding or difficult the integration process, the longer the reading time. In a second experiment, a more direct measure of attentional capacity was taken: while subjects were comprehending these paragraphs they simultaneously monitored for a stimulus external to the sentences (a probe). Presumably, the more attention consumed, the slower the response to the probe. In a third experiment, memorial availability was observed. Immediately after comprehending the fourth or fifth sentence, subjects were asked to make a speeded judgment as to whether a test sentence was identical to a previously presented one. Presumably, the less available a memory representation, the slower and less accurate the judgment.

There were several possibilities. First, it was possible that we are only sensitive to one or two of these linguistic coding devices. Second, perhaps only one or two of the investigated cognitive responses are manifestations of the proposed processing shifts. Third and least optimistic, it could be that none of the linguistic coding devices noticeably lead to any of the cognitive responses, at least as I operationalized all these concepts. Now, speaking of watermelons ...

Thematic organization and attention in discourse production

Russell S. Tomlin
Department of Linguistics and Cognitive Science Program
University of Oregon, Eugene

On-line oral descriptions of brief computer animated films show that the organization of discourse production and its accompanying syntactic manifestations can be changed under differential priming of visual attention before and during viewing.

In recent linguistic literature, there has been considerable interest in the role played by syntax in coding different kinds of discourse information (cf. recent collections by Chafe (1980) or Givon (1983)). Numerous studies have examined the syntactic coding of thematic information, information of relatively greater importance to the overall discourse, considering both clause-level theme or topic (thematic arguments) and foreground-background information (thematic propositions). In each of these kinds of cases differences in the relative importance or salience of arguments or propositions to the overall discourse correlate with differences in their syntactic treatment (e.g. thematic arguments are coded by syntactic subject; thematic propositions by independent clauses).

In all such studies of thematic information and syntax there exists a fundamental weakness in argumentation for the functional correlation proposed. This weakness is the problem of identifying the thematic units of interest in actual discourse data clearly and explicitly and with syntax-independent criteria. Current identification strategies rely either on syntactic evidence (resulting in circular argumentation) or on individual introspection.

Looking to research in cognitive psychology suggests that the selective focusing on information according to its importance in discourse may be the linguistic reflection of the more general processes of selectively focusing attention during information processing. If a connection can be established between cognitive attention and linguistic attention, empirically more sound methods of identifying thematic information in natural discourse data may be developed.

Two groups of subjects were asked to produce oral on-line descriptions of animated films composed of multiple characters interacting in a variety of ways. Two versions of the computer animated film were made. Identical target segments in each film were differentially primed to increase or decrease the overall importance of individual characters and the target segment event to the overall film. Visual primes consisted either of scenes focusing on different characters preceding the target segment (external priming) or highlighting characters within the target segment (internal priming).

It is expected that particular characters and events will take on different levels of importance, or thematicity, in response to differential priming. This should result in subjects producing protocols which differ significantly in the assignment of arguments to subject position and in the use of referring expressions.

SYMPOSIUM: MULTIPLE LEARNING MECHANISMS: PSYCHOLOGICAL, NEUROPSYCHOLOGICAL AND NEUROBIOLOGICAL EVIDENCE

Richard Granger, University of California, Irvine, Chair

Larry Squire, VA Hospital and University of California, San Diego

Mortimer Mishkin, National Institute of Mental Health

Gary Lynch, University of California, Irvine

SYMPOSIUM: KNOWLEDGE BASED APPROACHES TO THE STUDY OF MEDICAL PROBLEM SOLVING

Guy J. Groen, McGill University, Chair

Paul J. Feltovich, Southern Illinois University

Allan Lesgold, University of Pittsburgh

Harriet Rubinson, University of Pittsburgh

Dale Klopfer, University of Pittsburgh

Robert Glaser, University of Pittsburgh

William J. Clancey, Stanford University

Vimla Lodhia-Patel, McGill University

GENERAL SUMMARY OF SYMPOSIUM

This symposium is concerned with the nature of competent performance by expert physicians. While this issue has been studied from the point of view of cognitive science for a number of years, the earlier work tended to concentrate on general problem solving processes. No effort was made to characterize the knowledge base or investigate processes highly dependent on it. In terms of Newell's well known distinction, the focus was on weak rather than strong methods.

More recently, however, a number of approaches have evolved that attempt to take directly into account the relationship between the knowledge base and the processes utilized. They stem from four areas, each of which is represented in this symposium: problem solving (Feltovich), knowledge engineering (Clancey), propositional analysis (Patel et al) and the psychology of perception and memory (Lesgold et al).

These approaches all have in common the fact that they tend to go beyond the standard theory and empirical methodology of problem solving. Comprehension processes play a crucial role. The models are more oriented towards frames and schemas. The empirical techniques make far more extensive use of probes.

Medical problem solving is almost a prototype of a "messy", realistic domain. It deals with extremely complex stimuli and verbally rich protocols. Performance is highly dependent on an elaborate but only partially well defined knowledge base. Because of this, the approaches described by the participants of this symposium may be relevant to other areas where similar issues of complexity must be faced.

Knowledge-based Components of Expertise in Medical Diagnosis

Paul J. Feltovich
Southern Illinois University School of Medicine

Studies of clinical reasoning prior to this one had established the general form of clinical reasoning as hypothesis testing; cues in patient data suggest interpretive hypotheses which direct further interrogation of a case. However, parameters of this process reflecting timing and number of hypotheses did not discriminate expert from novice reasoning. As also indicated by other psychological studies emerging at the time (e.g., chess), expertise appeared to reside in the quality of diagnostic outcomes (intermediate and ultimate) and in the knowledge base supporting reasoning. The present study in pediatric cardiology set out to explore the roots of quality in the knowledge base of clinical practitioners.

Hypotheses about the disease knowledge base and its development with experience were proposed, involving: (1) density, the clustering of similar diseases into categories, (2) precision, the tuning of clinical expectations in a disease to their naturally occurring variability, and (3) non-classicality, the elaboration of disease prototypes into many less representative variations.

Medical subjects (students through highly experienced professionals) diagnosed clinical cases while thinking aloud. Each case was designed to test one of the aspects of disease knowledge. This was aided by the use of a "garden path" methodology by which subjects were led initially to an erroneous hypothesis and had to adjust in particular ways if they were to be successful in the case. The focus on quality in the study was addressed through analysis of subjects' use of "logical competitor sets" (LCS), hypotheses sharing a "deep" physiological commonality and defined in advance to be plausible explanations for each case.

Results were generally consistent with the hypotheses concerning knowledge base development. Experts generated and considered LCS diseases in groups, branched to subtle disease variations where appropriate, and evaluated alternatives appropriately to make discriminations. Novices considered subsets of the LCS more in isolation, often focusing on the classic prototype, and made systematic errors of evaluation, reflecting imprecision.

Quality in clinical reasoning was attributed partly to a dense interactive knowledge base for diseases organized by pathophysiological "deep structure," many variations on each disease theme, and well-tuned clinical expectations for diseases.

Radiological Expertise and Its Acquisition

Alan Lesgold
Robert Glaser
Harriet Rubinson
Dale Klopfer

Data will be presented from studies of radiology residents and experts showing differences in the qualitative characteristics of the diagnostic process at journeyman and master levels. These differences are generally consistent with other data on expertise but have a unique qualitative character because of the heavy perceptual loading of the radiological diagnosis task in addition to its cognitive side. Expertise plays a heavy role even in *where* particular features are seen. Preliminary results of longitudinal tracking of attended features over the course of learning will be used to illustrate effects of deep conceptual knowledge on diagnosis.

Some of the assertions supported by the results to date include:

- The ability to construct detailed mental representations of the physical situations which gave rise to an image is a critical aspect of expertise in perceptual diagnostic skills.

- The knowledge which underlies radiological expertise includes the mental representation of relevant body structures, a theory of the perturbation of those structures under pathology, and the projection of those structures into the domain of diagnostic images.

- Experts have more automated perceptual processes that are sensitive to a variety of anatomical variations. These automated capabilities are the product of more elaborated and flexible schemata that are slowly built up over the course of training.

- Novices have difficulty constructing representations in which the same stimulus feature is mapped onto two different objects.

- The precise, rapid recognition skill that characterizes expertise involves interactions between higher and lower levels of representation. It is not purely top-down or bottom-up.

- Experts recognize constraints on problem solution early but defer decisions until they are necessary. Experts are opportunistic planners.

- Novices are more likely to maintain bad film interpretations in the face of discrepant evidence from the patient's clinical history.

- Novice schemata are classic and less tunable

The Operators of Diagnostic Strategy

William J. Clancey

Stanford University

NEOMYCIN is a computer program that models one physician's diagnostic reasoning within a limited area of medicine. NEOMYCIN'S diagnostic procedure is represented in a well-structured way, separately from the domain knowledge it operates upon. We are testing the hypothesis that such a procedure can be used to simulate both expert problem-solving behavior and a good teacher's explanations of reasoning.

The model is **acquired** by protocol analysis, using a framework that separates an expert's causal explanations of evidence from his descriptions of knowledge organization and strategies. The model is **represented** by a procedural network of goals and rules that are stated in terms of the effect the problem solver is trying to have on his evolving model of the world. The model is **evaluated for sufficiency** by testing it in different settings requiring expertise, such as providing advice and teaching. The model is **evaluated for plausibility** by arguing that the constraints implicit in the diagnostic procedure are imposed by the task domain and human computational capability.

This paper discusses NEOMYCIN'S diagnostic procedure in detail, viewing it as a memory aid, as a set of operators, as proceduralized constraints, and as a grammar. This study provides new perspectives on the nature of "knowledge compilation" and how an expert-teacher's explanations relate to a working program.

PROPOSITIONAL REPRESENTATION IN THE ANALYSIS OF CLINICAL PROBLEM SOLVING

Vimla L. Patel
Guy J. Groen

Centre for Medical Education,
McGill University

We consider a model of clinical reasoning based on theories of cognitive processes in text comprehension which views expert diagnostic reasoning as driven by case comprehension. Our principal means for identifying the cognitive processes underlying case comprehension is through the application of the propositional analysis techniques of C.H. Frederiksen. In an analysis of inferences that subjects produced in recalling case information, Patel and Frederiksen have shown that there is a difference in case representation and interpretation by physicians and medical students.

In this paper we focus on procedures which look directly at problem solving and the relationship of problem solving to case comprehension. Here, problem solving is viewed as a sequence of frame transformations, the final transformation being the one that generates the diagnosis. We investigate the physicians' ability to shift between various frames, using specific questions as probes. The task involves presenting a physician with a clinical problem text (infectious endocarditis) to read and then recall. Next, the physician is asked to summarize the pathophysiological aspects of the clinical problem. Finally, the physician is required to make a diagnosis. The frame construction during a problem solving task revealed in the subjects' responses to specific probes provides the "baseline" information against which all question-induced shifts in frame processing are assessed.

This procedure directly manipulates the frame construction (or transformation) processes through the use of probes that require processing a problem solving text according to a particular frame structure, and we use this procedure to study the extent to which there is a frame shift with respect to specific probes in physicians' clinical reasoning process.

SYMPOSIUM: THE BIOLOGICAL CONSTRAINT

James L. McClelland, Carnegie-Mellon University, Chair

Neal Cohen, Johns Hopkins University

Jerry Feldman, University of Rochester

Paul Rozin, University of Pennsylvania

Terry Sejnowski, Johns Hopkins University

The Biological Constraint

A Symposium

Organizer: James L. McClelland
Department of Psychology
Carnegie-Mellon University

Many biologists see human cognition as a supreme example of the power of biological mechanisms. But cognitive scientists often forget that cognition is a biological process at all.

In the early days of cognitive science and its predecessors, a shift away from biology was important and justified. Our first task was to show that ideas, memories, thoughts and perceptions were important explanatory constructs, independent of their possible implementation in the brain. Now, it occurs to more and more of us that what we know about the brain and its evolution might help us to understand cognition more deeply. To some, the standard tools of the cognitive psychologist seem insufficient to answer fundamental questions. To others, the struggle to implement intelligence in the von Neumann computer has yielded increased respect for the computational power of the brain. These and other feelings have motivated a search for clues to cognitive function in the biological substrate.

As we have been moving in these directions, there has been considerable development around us. A number of researchers trained in neurophysiology have begun to suggest how what we know about the brain can be used to enhance our models of cognitive processes. Neuropsychologists have been employing the methodological and theoretical tools of the cognitive scientist in their analyses of neurological dysfunction, with results which are sure to influence theorizing about normal cognition. And some psychobiologists have suggested how an evolutionary perspective might help us understand some aspects of mental organization.

This symposium is intended to celebrate and reinforce these trends. Four speakers, each representing a different viewpoint on the biological constraint on cognition, will give us brief glimpses of their work. Neal Cohen describes some striking evidence of preserved ability to acquire cognitive skills in patients who show profound deficits in the ability to remember facts and events. This evidence has profound implications for theories of learning and memory. Paul Rozin explains how an evolutionary approach can shed light on the roles of specialized modules and generalizable systems in cognition. Terry Sejnowski describes some of the properties of the cerebral cortex and points to some of the implications of these properties for biological computation. And Jerry Feldman tells us some of the ways we must change our thinking about cognitive processes if we wish to develop models that respect the strengths and weaknesses of the brain.

Neuropsychological Constraints on the Organization of Memory

Neal J. Cohen
Department of Psychology
Johns Hopkins University

Neuropsychological investigation of patients with cognitive impairments following brain insult has long been directed at understanding the the organization of mental processes at an anatomical level. Recently, however, considerable attention has been focused on the potential contributions of neuropsychological work to an understanding of organization at a functional or cognitive level. Damage to discrete regions of the brain can produce surprisingly circumscribed deficits that are specific to particular cognitive domains, and, indeed, specific to particular component processes or systems within a given cognitive domain. The selectivity of these deficits indicates constraints on cognitive organization imposed by the architecture of the brain. Identification and characterization of the nature of these selective cognitive deficits has permitted new, principled inferences to be drawn about the organization of normal cognitive systems, and it has provided confirmation of the biological validity of aspects of models derived from work with normal subjects or computer simulations.

This paper will focus on one example of the way in which cognitive neuropsychological research identifies important biological constraints on cognition. In brief, recent studies have revealed that amnesic patients with severe and pervasive disorders in learning and memory can nonetheless acquire new cognitive skills at a rate and in a manner comparable to normal control subjects. These subjects retain the capacity to acquire information that guides performance while at the same time showing a profound disability in acquiring memory traces accessible to verbal report or other explicit memory retrieval processes. For example, such patients can learn the optimal solution to the Tower of Hanoi problem, even though they may be incapable of remembering that they have done the problem before, or that they know how to solve it. They may also be incapable of explaining either the solutions they produce or the constraints they observe in producing a legal and optimal series of moves.

Future models of normal learning and memory must account for this striking dissociation. Such accounts may be built around the assumption that there are fundamental differences between the memory processes or systems mediating skilled performance and those mediating the remembering of facts and events.

Adaptive Evolution Applied to Cognition and the Concept of Accessibility

Paul Rozin
Department of Psychology
University of Pennsylvania

Cognitive processes, like other biological phenomena, are subject to adaptive and evolutionary explanations as well as to explanation in terms of mechanisms. In addition to providing a different type of explanation, adaptive (functional) and evolutionary explanations can enlighten the search for mechanism in a number of ways. For example, principles of behavior or cognition can be arrived at by a study of function (e.g., optimization). In addition, functional considerations may provide clues to mechanism and focus research on phenomena with ecological validity.

The adaptive and evolutionary approach emphasises diversity, both within the organism and across species. Some processes -- and certainly, some processing principles -- may be domain general. But most cognitive and behavioral adaptations are seen as solutions to specific problems and hence are expected to show specific adaptation to the peculiar properties of these problems. Adaptive specializations of this sort are illustrated by highly sophisticated perceptual and navigational abilities in some species. These specializations are available only to a restricted set of inputs and outputs, and are therefore described as inaccessible.

It is proposed that cognitive capacities can be arrayed on a dimension of accessibility. At one extreme these capacities are inaccessible or domain limited. At the other extreme they are utilized in many systems and perhaps available to consciousness. Part of the evolution of intelligence might involve increasing accessibility of originally tightly wired or inaccessible programs. Thus, while Pavlovian Conditioning may have originally appeared in a very specific context, through evolution it becomes a generally accessible system. The major evolutionary mechanism responsible for this is probably preadaptation, in which a structure originally evolved for one purpose is used for another. This can happen in two ways: By sharing of the actual circuitry or by reduplication of the circuitry through use of genetic blueprints.

Accessibility also describes some basic features of development. Piagetian decollages are examples of increasing accessibility in which the same program becomes operative in wider and wider domains. Similarly, some aspects of education involve getting access to what one already knows. Thus, in learning alphabets the critical insight has to do with phonemic segmentation. This is carried out by a specialized part of the brain involved in speech perception and tied into the ear-mouth input output system. But in learning to read, one must access this system through a new route, the visual system. On the other hand, many aspects of neuropsychology can be described as loss of access. This is essentially what Geschwind describes as a disconnection syndrome.

In general, the concept of accessibility allows for different degrees of domain specificity and for systematic changes usually in the direction of greater accessibility in evolution and development, and in the direction of decreased accessibility in aging and neuropathology.

Cortical Computation

Terrence J. Sejnowski
Biophysics Department
Johns Hopkins University

Computation by digital computers is the only thoroughly studied model of symbolic information processing that we have, so it is not too surprising that computation is often defined within the conceptual framework of von Neumann machines. It is difficult to underestimate how this model of computation dominates our thinking. If one wanted to study language, for example, it would seem that measuring electrical and chemical signals inside the brain would be of as little use as measuring signals inside a digital computer if one wanted to study the programs that were running. This reasoning, however, may be profoundly misleading because digital computers were designed with very different constraints and for very different purposes.

Structural reasoning about biological systems can best be explained in a system that is already understood at a conceptual level. Take for example, the digestive system, which once was as obscure as the nervous system. How much can be learned about digestion if one examines only what goes in and what comes out? A great deal can be learned about the composition of food and the composition of waste; however, the transformation between them is highly under determined and will also depend on an unpredictable internal state. The problems of digestion were solved by looking into our bodies and discovering organs and functions for those organs. The concept of an enzyme, a key to the digestive process, followed from a detailed analysis of the internal fluids.

The architecture of the brain is much more closely tied to its function than is the architecture of general purpose digital computers. The anatomical connectivity, both within an area and between areas, is itself highly revealing about the type of computation that is performed. Many areas of the brain are dedicated to special tasks, such as photo-transduction in the retina and eye movements in the oculomotor nuclei, and the part of the brain that is most closely associated with cognition, cerebral cortex, is parceled into areas having different functions. From recent anatomical and physiological work in many cortical areas, a new view of cortical processing is emerging that is different from the von Neumann architecture. This view is based on parallel rather than serial processing, distributed rather than local representations and stochastic rather than deterministic algorithms, and may offer an alternative conceptual framework within which to think about our cognitive abilities.

Computational Constraints from Biology

Jerry Feldman
Department of Computer Science
University of Rochester

Even the crudest consideration of neural computation imposes severe constraints on the plausible organizations for cognitive processes. The most obvious constraint is the remarkably small number of sequential time steps involved in intelligent activity, but there are additional constraints imposed by the moderate number (~1011) of units, their limited (< sqrt N) connectivity, and the relative lack of plasticity in adulthood. The exploration of the computational consequences of these constraints has already been fruitful and could become a significant aspect of Cognitive Science.

One consequence of taking these computational constraints seriously is a profound reservation on the ultimate viability of many of the information processing models currently dominating the field. Any paradigm that depends on central control, data structures or symbol manipulation presents the problem of having no obvious reduction to the underlying computational system. Researchers motivated by biological constraints have tended to work on positive results rather than argue paradigms and have been exploiting insights gained through traditional approaches. But it does seem likely that many problems that appear intractable in conventional information processing paradigms will be accessible in a more natural formalism and that cognitive scientists from all domains should examine whether careful consideration of the biological constraint might be timely.

SUBMITTED PAPERS

A MODEL OF EXPERT DESIGN

Beth Adelson, David Littman, and Elliot Soloway

Cognition and Programming Project
Department of Computer Science
Yale University
New Haven, Connecticut 06520

1. Motivation and Goals[1]

In this paper we will present a model of expert software design which we have developed in the course of analyzing protocols of expert designers designing an electronic mail system. Two goals motivated this work: The first was to see experts solving problems which called upon their problem solving abilities, as well as their "routine cognitive skills". The second was to create a situation in which general problem solving operators could be seen to interact with a rich knowledge base. Towards these goals we presented expert software designers with a novel and complex problem from a domain with which they were familiar. The result was a model which unites several repeatedly found behaviors into an interpretable whole (Jeffries, Turner, Polson & Atwood, 1981; Kant and Newell, 1982; Atwood, Turner, Ramsey and Hooper, 1979).

2. Methodology

Subjects. Three expert designers. Each of our experts had worked for at least eight years in commercial settings designing a wide variety of software.

Procedure. We presented each of the designers with the following design task to work on.

> ***TASK*** -- Design an electronic mail system around the following primitives: READ, REPLY, SEND, DELETE, SAVE, EDIT, LIST-HEADERS. The goal is to get to the level of pseudocode that could be used by professional programmers to produce a running program. The mail system will run on a very large, fast machine so hardware considerations are not an issue.

The task we gave our subjects had several important properties: It was complex, requiring close to two hours of our expert's time. We hoped therefore to be able to see something of the skills that make up expertise. It was novel, none of our experts had designed a solution to the problem previously. This meant that we would not be seeing only "routine cognitive skill", but some problem solving as well. In addition, the problem we chose was similar to the type of problem with which our experts were familiar. Therefore, although none of our experts had designed a mail system before, they were used to designing other types of communications systems and so we would be able to observe them in a situation where they had rich knowledge bases to turn to.

Organization

In Section 3 we outline the main elements of the model. Next, in Section 4, we present larger portions of our experts' protocols in order to support our claims and to bring out some of the interesting interactions among the components of the model.

[1]This work was sponsored by a grant from ITT.

3. The Model

3.1. Components of the Model

Our model of the experts' design process contains four major elements:

- A Design Meta-Script. The function of the Meta-Script is to drive the design process by setting three general goals:

  1. To check the current state of the design for sufficiency. This means that all of the elements needed to specify the design at the current level are present.
  2. To check the current state of the design for consistency. This means that all elements of the design are compatible at the current level of specificity, and that no element causes an inconsistency with currently known constraints both at higher and at lower levels.
  3. To expand the design from its current level of specificity into the next level of specificity.

  The main operator used to achieve these goals is the running of mental simulations of the partially completed design. The function of these simulations is explained in the next section which describes how the elements of the model function.

- A Sketchy Model. The Sketchy Model resides in working memory and as the design process proceeds the model becomes increasingly less sketchy. The model is initially sketchy in that the designer does not yet understand its functionality down to a level of detail which would be sufficient to produce an implementable program. In addition, the constraints or assumptions of the design are not entirely understood. One way of picturing the model is as a tree that grows in both depth and breadth as the designer's understanding of the problem specification increases (Jeffries et al., 1981; Atwood, Turner, Ramsey, and Hooper, 1979.)

- The Current Long Term Memory Set. This is the set of long term memory elements that are currently under consideration. This set would consist of all of the known solutions appropriate to the aspect of the design that is currently being worked on. Choosing an element from the set currently under consideration allows the designer's model to become less sketchy because the element selected from long term memory is now added to the model in working memory. It also causes a different long term memory set to be considered on the next iteration of the design process.

- The Demons. These contain the designer's notes to himself. The notes are things to remember such as constraints, assumptions, or potential inconsistencies. A note will be placed in a demon if it is: a. too concrete to be resolved at the time that it is thought of and b. needs to be considered when the designer's model has reached a level of concreteness that matches the note. The demons are able to monitor the state of the Sketchy Model. When the level of detail of the Model is equal to the level of detail of the note the demon calls attention to itself. (The reader will notice that our demons are rather unusual in that they are active formation gatherers.)

Summary of the Model

At the most abstract level, the designers were performing a means-ends analysis driven by the Meta-Script. In the analysis the goal state was an implementable design specification and the current state was the designer's increasingly detailed Sketchy Model of the problem solution. The

designers moved towards the goal state by repeatedly simulating executions of their incomplete models. For all of the designers observed, this appears to be the most powerful and frequently used operator in the means-ends analysis. However these simulation runs of the partial models served to decrease the gap between the current and goal states in a number of ways. This will be expanded upon in the next section where we present portions of our protocols. The protocols will help to give the reader a clearer picture of how the model functions. It will also bring out some of the interesting interactions among the components of the model and generate some as yet unaddressed questions.

4. Recurrent Behavior Accounted for by the Model

Here we present behaviors which we found were repeatedly exhibited by our designers. The model unites these behaviors into an interpretable whole.

Observation I. How the Design Proceeded

There was a surprising degree of similarity in the time line of the subjects.

Figure 1: Subjects' Timeline

As illustrated in the figure above, first the designers described how a user would view the mail system, then they expanded upon the various assumptions and constraints of the problem (e.g, S1: "We will assume dumb terminals", S2: "the number of users will not be fixed"). Only then, approximately 20 minutes into the session, did the experts begin to construct a working model of the mail system. This model also changed over time in that it began as a very skeletal version of a mail system and then became increasingly concrete as the design progressed. The following quote from designer S2 illustrates this progression.

At 20 minutes into the task expert S2 said:

> "We can now start thinking about what type of processing structure is required for implementing (the mail system). In order to get the idea about the structure, we can see some kind of a state diagram which shows the dynamics of the system.... What we can see here is one state (accessing) several other states and after the operation is completed, control of the state transition going back to the initial state. This will help us structure our solution to the problem at a higher level. Then we will go into each one of the building blocks that help us write the processing step at each step in the state diagram."

Why did the experts spend the first twenty minutes of the session gathering information? The answer is not that they were hesitant, (e.g. S1's quote ten minutes into the session: "The program design gives me no cause for concern at all.") As discussed in the previous section, the expert

designers search long term memory for stored solution elements. An effective search results from first choosing the right set of memory elements and then choosing the right element from among the set. This type of choice would be aided by having a sufficiently rich set of constraints and assumptions. And it seems that experts do not begin their search until this information has been obtained.

Observation II. Maintaining Balanced Development

The designers always followed a course of *balanced development*. Our subjects attempted to develop each of the components of the design so that none of them acquired significantly more detail than any of the others. The following quote is representative of this behavior. Its significance becomes clear when we consider the next observation, *Simulation Runs of the Sketchy Model.*

> S3: ...you've never got so deep in that you can't improve it taking account of something you think of later... No, oh no I'm not going to take something down to its itty bitty conclusion because bet you I'm going to have to change it. When I take something else out and say (echh??) there's no logical consistency between that, remembering that mail is a two way thing. I'm going to have to reply to it and I may have to use this information to construct a message back ... I can see too many possible interactions between the pieces so it would be nicer if they all had some logical similarity.

Observation III. Simulation Runs of the Sketchy Model.

We observed all of our subjects repeatedly conducting mental simulation runs of their partially completed designs. The experts would consult the state of the sketchy model and then conduct a simulation of the model at its current level of abstraction. Thus we observed simulations which became increasingly concrete as the design progressed. For example, in S3's early simulation he saw the mailer as "information flowing through a system", whereas in a later simulation when considering his module for the READ function, S1 drew a state diagram for all of the states which could be reached from READ. What is the function of these simulations, which appear in our protocols and in those of Kant and Newell (1982)? Recall that the design process is driven by the Meta-Script. The goals of the designer's Meta-Script are to check the current Sketchy Model for consistency and completeness and if these criteria are met the next goal is then to try to expand the Model. Therefore, in order to meet the goals of the Meta-Script a simulation run of the Sketchy Model in its current state is conducted (e.g. S1 draws a state diagram for the current READ module to see how it behaves at this point in its development). We can now see why the designers maintain balanced development; *it would be difficult to run a simulation with elements at different levels of detail.*

Recall that the other goal of the Meta-Script was to expand the Sketchy Model. The simulation is used in this process as well. It points out an element of the Sketchy Model that needs expansion and the subject then accesses the appropriate long term memory set in order to choose how the expansion should proceed. This issue is further addressed below.

Observation IV. Notes and Interrupts

We found that the expert designers would frequently make "notes", to themselves about things to remember later in the design process. These notes had to do with constraints or partial solutions or potential inconsistencies which needed to be handled in order to produce a successful design. The reason that these notes were not handled immediately was that they were concerned with a level of detail which was greater than the level of detail of the current state of the Sketchy Model. This means that incorporating them into the design when they were thought of would

have violated the principle of balanced development. This in turn would have interfered with the process of running simulations, which, as mentioned above, was a process upon which the experts rely quite heavily. We also found that the expert designers would be reminded of previously made notes once the current state had reached a level of detail which would allow the note to be incorporated into the design without violating balanced development. Data of this sort is what led us to posit the existence of demons which were able to monitor the state of the Sketchy Model in order to interact with the design process without disrupting it.

5. Open Questions

Three issues of theoretical significance are raised but not settled by our observations.

- *How do expert designers decide which element in the Sketchy Model to expand upon?* We suggest that, the process of accessing long term memory serves both to guide the articulation of plans for expanding the current node in the design tree and for elaborating goals appropriate to the next level of detail.
- The second stems from the not unlikely possibility that the expert's long term memory will contain several potential solutions for some aspect of a problem. *This raises the question of how the expert chooses among them?* We believe that our expert designers could and did rely heavily on effective communication between the current state of their Sketchy Model and their stored knowledge about software design. That is, we believe that the expert designers' understanding of the current problem served to form an effective index into memory. However, we do not know what this index looks like. We also do not know whether the index handed to memory is the result of heavy inferencing. It could just as well be that memory does the inferencing about what is needed in the current context. Either way, the experts have developed a highly effective retrieval mechanism.
- *How is the Sketchy Model used to perform the simulation?* In our description, the Sketchy Model has the quality more of a structure than of a process, but somehow it is repeatedly used as a running system which the designer can "play with" in order to find emergent properties. The idea of a Sketchy Model needs to be conceived of in a way which takes this aspect of its functioning into account (Collins and Genter, 1982).

6. Concluding Remarks

We have chosen a complex and novel task for our subjects from a domain with which they were familiar. This has allowed us to develop a model which can account for several interesting behaviors such as constraint gathering, balanced development, and the building and running of simulations of partially completed designs. We have proposed a model in which there is a good deal of intelligence given to some of the components (e.g. demons which can understand the current state of the Sketchy Model), as well as frequent interaction between meta-knowledge about design and content specific knowledge about communication systems.

Acknowledgements

Thanks to Ruven Brooks, John Black, Larry Birnbaum, and Kate Ehrlich for their time and thought.

7. References

Atwood, M., Turner, A., Ramsey, R., and Hooper, J. An exploratory study of the cognitive structures underlying the comprehension of software design problems. ARI Technical Report 392. SAI-79-100-DEN. 1979.

Collins, A. and Gentner, D. Constructing Runnable Mental Models. Proceedings of the Fourth Annual Cognitive Science Society. 1982.

Jeffries, R., Turner, A., Polson, P. and Atwood, M. The processes involved in designing software. In Anderson (Ed.) *Cognitive Skills and Their Acquisition*: Hillsdale, NJ: Lawrence Erlbaum Associates, 1981.

Kant, E. and Newell, A. Problem Solving Techniques for the Design of Algorithms. Tech Report No. CMU-C S-82-145 1982.

COMMUNICATING ABOUT ROLES IN HUMAN INTERACTIONS

Airenti G.^, Bara B.G.^, Colombetti M.^^

^Unita` di ricerca di intelligenza artificiale
Universita` di Milano
^^Progetto di intelligenza artificiale
Politecnico di Milano

1. INTRODUCTION

In this paper we investigate the structure of communication in interpersonal planning.

Human action is mostly interpersonal and thus reveals the ability of actors to cooperate. Interpersonal plans contain both actions to be performed by the planner and actions to be performed by partners. In order to obtain cooperation, the planner has to induce the partner to perform the action assigned to him in the plan; this requires that the planner causes the partner's intention to perform such an action. The most usual way for the planner of achieving such a result is through communication.

To deal with communication within the context of action, we assume the standpoint of speech act theory (Searle, 1969). We consider inducing the partner to perform an action as a perlocutionary act performed by the planner. We assume that such a perlocutionary act is performed via an illocutionary act of the directive type (Searle, 1979), like an action of requesting.

In Airenti, Bara and Colombetti (1983a, b), we have argued that interpersonal action is regulated by script-like structures which we call games. Games define the roles of the actors within the interaction and in particular mediate between the request of the planner and the motivations of the partner.

In this paper we suggest that the communicative transaction aimed at gaining the cooperation of the partner is always paralleled by communication about the roles to be played by the actors in the game. The analysis of communication about roles is therefore necessary to account for the performance of the perlocutionary act of inducing, either successful or not.

2. INDUCING A PARTNER TO COOPERATE

In the following we shall analyze examples of communicative transactions of the type described in Fig. 1, where an illocutionary act of request is performed in order to induce somebody to execute an action.

(1) Anna needs a stamp, gets into the nearest shop and asks the shopkeeper for the local post office.

(1.a) the shopkeeper gives the information requested

Fig. 1.

(1.b) the shopkeeper answers he is not an information office

(2) Alexandra meets a well known psychoanalyst at a party and tells him her last dream

(2.a) the psychoanalyst stoically accepts to listen

(2.b) the psychoanalyst replies giving her the phone number of his office

(3) The boss and his secretary are sitting in their office with the window open.

(3.a) the boss says it is cold and his secretary stands up and closes the window

(3.b) the secretary says it is cold and the boss replies she can close the window

In example (1) Anna addresses a partner proposing an interaction different from his expectations. In fact Anna thinks she can get the information she needs by playing a courtesy game with the shopkeeper. The reply of the shopkeeper depends on his motivation to play the proposed game besides his usual role of dealer. The main point here is that Anna does not propose the interaction specific to the context of the shop, but a more general and broadly applicable one.

A different case is shown in example (2), where Alexandra proposes to the partner to play his usual professional role, but in an inadequate setting. Response (2.b) can be attributed to a refusal by the psychoanalyst to meet the request on the basis of a wrong context. Instead, response (2.a) can be viewed either as the playing of a politeness game, quite appropriate in the party context, or as the extension of the psychoanalysis game beyond the usual setting.

In example (3) the same statement assumes two different meanings depending on the respective roles of speaker and hearer. We can explain case (3.a) considering the statement of the boss as a request that the secretary close the window. In case (3.b) the statement of the secretary can be interpreted as a request of permission to close the window. Both cases admit alternative explanations according to different intentions of the actors. In (3.a) the boss could have no intention of indirectly requesting the cooperation of his secretary. In this case, her response could be interpreted either as a misunderstanding or as an intentional redefinition. In case (3.b) the statement of the secretary could be an indirect request that the boss close the window. Also the answer of the boss can be viewed either as a

actual misunderstanding or as an intentional redefinition.

The analysis of the examples shows that communicative exchanges are based on the roles played by the actors in the interaction. In fact, roles are determinant for playing a communicative act, for understanding it, and for planning the response. We must therefore postulate knowledge structures (games) which codify such roles in a specified context and are used for the functions just mentioned.Moreover, in order to explain why the partner accepts the role proposed by the planner or rather makes a new proposal, we must take into account the motivations of the partner.

3. COMMUNICATING ABOUT ROLES

We think of games as knowledge structures defining which actions should be performed by each player, at a given level of abstraction. The role of the player in the game corresponds to the actions assigned to him in that structure. The game provides for validity conditions which caracterize the context in which it is supposed to be played. In order to be played by two actors, a game must be shared, i.e. known to both of them, and it must include the two actors as possible players in the given validity conditions. Games may be shared by everybody (e.g. general laws of social behavior), by a group of people (e.g. in professional practice or in the underworld), or by a very restricted group (e.g. a family, or two old friends).

Different kinds of games are played on the basis of different motivations. For instance the motivations which underlie professional practice are different from those involved in friendship. For our purpose a motivation can be regarded as a mental structure which generates an intention under given conditions (compare with the concept of theme in Wilensky, 1983). For example, the motivation of preserving one's life generates the intention to run away from a dangerous situation.

In Airenti, Bara and Colombetti (1984), we provide for a formal treatment of the inference processes which underlie planning and understanding communicative transactions on the basis of games and motivations. The critical feature of motivations for playing a game is that they always contain, among their activation conditions, the fact that the planner is proposing himself as a player of the game. For instance, if one asks for a coffee in a coffee shop, he is proposing himself as a client and thus activates the waiter's motivation to do his job.

In Fig. 2 we give the complete sketch of a communicative transaction within an interpersonal plan, following the formal model presented in Airenti, Bara and Colombetti (1984). In order to induce the partner to perform an action, the planner has to induce the partner to play his role in the game which assignes to the partner the desired action. Such a result can be achieved if the partner has an adequate motivation which, as we have already seen, requires that he is convinced that the planner in turn intends to play his role in the game. Therefore, the planner has to convince the partner about his own intentions, and this amounts to performing a second perlocutionary act.

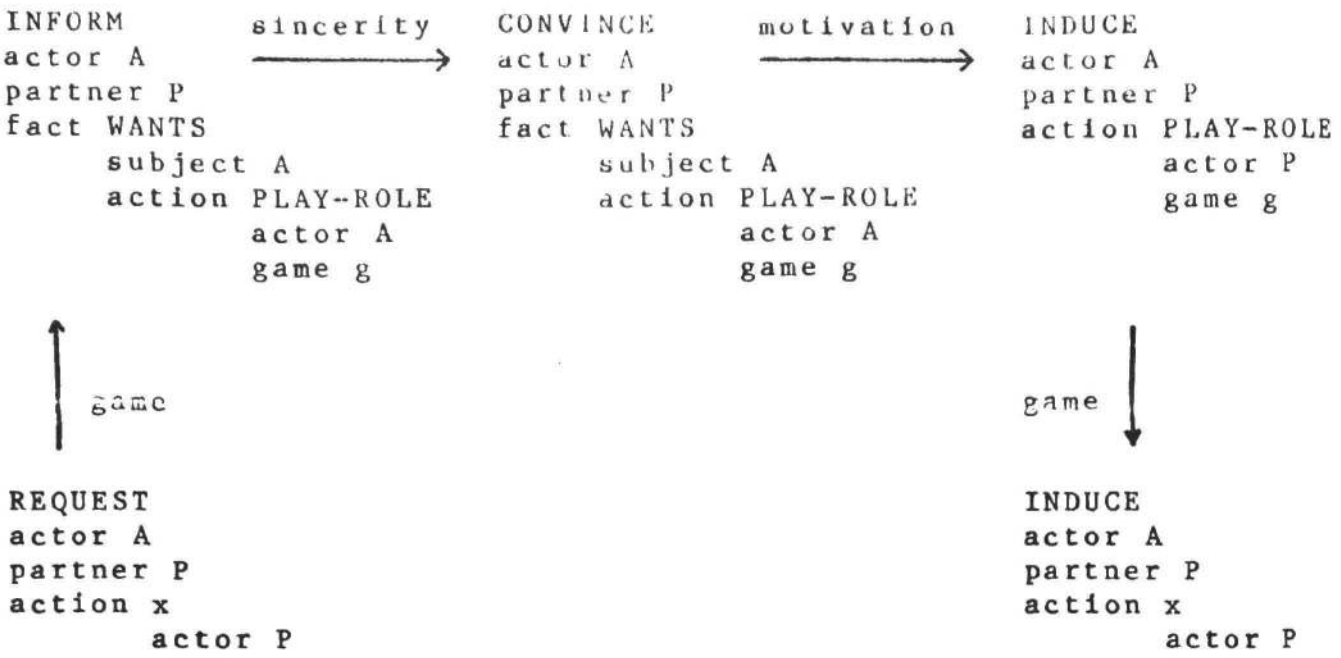


Fig. 2.

In order to convince his partner, the planner must perform the illocutionary act of informing him about his intention to play his role. The action of convincing is successful if the partner understands the illocutionary act and if he assumes that the planner is sincere. The main assumption here is that the illocutionary act of informing the partner about one's intention to play is realized through the illocutionary act of requesting the desired action within the validity context of the game.

Going back to example (1), we can now provide for a complete interpretation. The request of Anna is understood by the shopkeeper as the act of informing him that Anna wants to play the role of a person needing help in a courtesy game. The shopkeeper thinks that Anna is sincere about her intention and is therefore convinced she actually wants to play that game. If now the shopkeeper is motivated to play his role in the game we have case (1.a); otherwise we are left with a case similar to (1.b). Note that in this case it is reasonable to assume that the shopkeeper beliefs in Anna's sincerity; this can by no means be taken for granted in more complex interactions.

4. CONCLUSIONS

The main theoretical implications of the model we have presented are:

- the request of an actor, intended to induce a partner to perform an action, is always paralleled by communication about the respective roles, in order to meet the partner's motivation to cooperate

- communicating about the respective roles requires only that the request is performed within the validity context of the game. Therefore the action that the planner has to perform is the same described in Fig. 1
- an action of inducing may be performed through the same request, via different game-motivation pairs. For instance in (2) the psychoanalyst could have interpreted Alexandra's request as a seductive approach and either accept or refuse the interaction on that basis.

REFERENCES

Airenti G., Bara B.G., Colombetti M., 1983 a. Planning perlocutionary acts, Proceedings 8th IJCAI, Karlsruhe

Airenti G., Bara B.G., Colombetti M., 1983 b. The role of interpersonal games in perlocutionary acts, Proceedings 5th CogSci, Rochester,N.Y.

Airenti G., Bara B.G., Colombetti M., 1984. Planning and understanding speech acts by interpersonal games. In: Bara B.G., Guida G., eds., Natural language processing: computational models for production and comprehension, North-Holland, Amsterdam, in press

Searle J.R., 1969. Speech acts, Cambridge University Press, Cambridge

Searle J.R., 1979. A taxonomy of illocutionary acts. In: Expression and meaning, Cambridge University Press, Cambridge

Wilensky R., 1983. Planning and understanding, Addison-Wesley, Reading, Mass.

Acknowledgments

This research has been supported by a grant of the Consiglio Nazionale delle Ricerche (C.N.R.), under contract CT83.00355.04

Authors' address

Istituto di Psicologia della Facolta' di Medicina, via F. Sforza 23, 20122 Milano, Italy

Parallel Logical Inference

Dana H. Ballard and Patrick J. Hayes
Computer Science Department and Cognitive Science Program
The University of Rochester
Rochester, NY 14627

February 1984

Abstract

The inference capabilities of humans suggest that they might be using algorithms with high degrees of parallelism. This paper develops a completely parallel connectionist inference mechanism. The mechanism handles obvious inferences, where each clause is only used once, but may be extendable to harder cases.

1. Motivation

The prospect of automating inferences has long been the goal of researchers in artificial intelligence. The most obvious advantage is a more compact representation of knowledge bases (KBs). Without inference ability all relevant facts must be explicitly represented in the KB. Using inference, only a subset of the facts need be explicitly represented, since the rest can be derived when required. However, despite the huge payoff, this goal has so far proved elusive. One reason for pessimism is that the known algorithms for reasoning fall into the class termed NP-complete. In a nutshell, this classification means that no better algorithms are known than ones that try out all the possibilities. For theorem proving, the number of possibilities can be open ended. In contrast to this pessimistic result stands human performance data. Psychologists have shown the following *performance result*: *a huge variety of forced-choice decisions can be made by human subjects in under a few hundred milliseconds.*

This is a huge discrepancy in results. The theoretical result implies that problems of even a modest size can overwhelm today's computers, whereas the practical tests show complex decision making in 100 - 400 ms. Furthermore, we know that humans bring huge numbers of facts to bear to solve a specific problem. Thus we are led to conclude that either: (1) humans do not make complex inferences; or (2) humans use a better algorithm and/or data structure. In this paper we explore the second possibility. Our aim is to show that theorem proving can be done using a parallel probabilistic relaxation algorithm. The algorithm requires that problems be formulated as the intersection of (possibly huge) numbers of local constraints represented in networks. The intersection process takes a worst-case time proportional to the diameter of the network but in practice often runs in constant time. Of course any machine of constant size will not be able to handle hard

theorems in constant time. However, our conjecture is that: *theorems that humans can solve in a few hundred millisconds have a constant time solution on a parallel machine.*

For many scientific applications, an inference mechanism that handles only the simpler cases, and fails in many cases, might not be useful. However, for human inference mechanisms, this may not be the case. The reason is that the human inference mechanism can be viewed as one component of several in a perception-action process. For example, in our model, if the inference mechanism fails to identify a visual object, one of the options available is to move closer and gather more data. Thus our goal is to develop an inference mechanism that allows many inferences to be made in parallel but may also fail in many cases.

A general formulation of theorem proving is that of Robinson [1965]: to prove S $=>$ W where S and W are sets of clauses, we attempt to show that S $\cup$ ~W is unsatisfiable. One classical way of doing this is to use *resolution.* Two clauses, P(x)Q(x) and ~P(a), can be resolved to produce Q(a). The process of constraining the bindings of variables in the clauses is known as *unification.* The resolution theorem proving technique resolves pairs of clauses with the objective of producing the null clause. If this is done, the unsatisfiability of the set of clauses S $\cup$ ~W has been demonstrated, and consequently the theorem S $=>$ W is true.

The approach has several important assumptions: (1) *clauses may be used only once*; (2) *the knowledge base must be logically consistent*; and (3) *the method uses a large network that must be preconnected.*

Our approach uses observations by Kowalski [1975] and Sickel [1976]. First we try and filter the clauses using various kinds of constraints. This filtering process is parallel and removes options that are not compatible with the constraints. Once this is done we resolve clauses in parallel. During the development, the reader must constantly keep in mind the nature of the result: it is not guaranteed to work, but the hope is that it will work in most cases.

The overall organization of our parallel inference is shown in Figure 1. The machine has three basic parts:

1) *Consistency Constraints.* The first part has the goal of activating a logically consistent set of constraints. This is the focus of other research, and we assume that the enterprise is successful.

2) *Inference Constraints.* Filtering constraints [Sickel, 1976; Kowalski, 1975] deactivate parts of the network that do not apply to the problem.

3) *Resolution.* The last part of the algorithm uses a second filtering technique based on resolution. In this phase, parts of the network are deactivated if they correspond to pairs of clauses that would resolve where one of the pair contains a single predicate. If the entire network can be deactivated in this way, a proof has been found; otherwise, the result is inconclusive.

The formulation of the algorithm is in terms of a connectionist network [Feldman and Ballard, 1982] using a recently-developed probabilistic relaxation algorithm [Hopfield, 1982]. One of the key contributions of this paper is to show that theorem proving can be described in terms of this formalism. The formalism has several advantages, but the main one is elegance: the problem can be described in terms of nodes which have binary states. During the course of the computation, constraints cause nodes to be turned off or on.

2. The Filtering Process

The objective of the filtering process is to define a set of local constraints that reflect the rules of predicate logic. Starting with a predicate logic formulation, we can examine the set of clauses and derive constraints that must hold between them, the predicate symbols, and the terms. These constraints are expressed in a common network formalism. The network consists of nodes which have binary states as described in the previous section. At each step in the filtering process the constraints for a particular node can be evaluated by evaluating that node's local input. If it cannot be part of the solution based on this local evaluation, it is turned off. The turning off of a node may cause other nodes to be turned off. This process converges when no more node state changes can be made.

The filter network has five sets of nodes: (1) C, the set of clause nodes; (2) P, the set of predicate letters and their complements; (3) F, the set of clause fragments; (4) B, the set of bindings between fragments; and (5) S, the set of substitutions. In any set of clauses there will be one cluase node, $c \in C$ for each clause in the set. There will be one clause fragment node $f \in F$ for each predicate letter mentioned in the clause. There will be a separate binder node $b \in B$ for each possible resolution between complementary predicates. Finally there will be a substitution node $s \in S$ for each possible substitution involving a binder. For example, in the following set S $= \{c_1$: P(x,a), c_2: $\neg$P(b,y)$\}$,

$$
\begin{aligned}
C &= \{c_1, c_2\} \\
P &= \{P, \neg P\} \\
F &= \{(c_1, P), (c_2, \neg P)\} \\
B &= \{((c_1, P)\ c_2, \neg P))\} \\
S &= \{xb, ya\}
\end{aligned}
$$

There are five different kinds of constraints: (1) a predicate letter constraint; (2) a clause-predicate substitution constraint; (3) a clause constraint; (4) unification constraints; and (5) a substitution constraint.

The Predicate Letter Constraint. The predicate letter constraint is derived from propositional logic. If in the set of clauses a predicate letter appears without its complement or vice versa, then that symbol can be pruned from the solution. In terms of the filter network, this constraint is easily expressed as an excitatory constraint between different nodes representing predicate letters, as shown in Figure 2. The weights and thresholds are arranged so that both the node and its complement must be on to keep each other turned on.

The Clause-Predicate-Substitution Constraint. This constraint is derived from the clauses in a straightforward way. Each clause may be decomposed into triples consisting of: (clause symbol, predicate letter, term). For example, C_1: P(x)Q(a) may be decomposed into (C_1, P, s_1) and (C_1, Q, s_2) where s_1 and s_2 are appropriate substitutions (these will be discussed further as part of the substitution constraints). In the filter network, there are a set of clause fragment nodes *F*, one for each triple. A clause fragment node *f* is connected to each node in the triple by mutually excitatory connections as shown in Figure 3. Its threshold is such that it will turn off if any of its constituents turns off.

The Clause Constraint. The clause constraint captures the notion that a clause can only be part of the solution if all of its fragments have viable bindings. Thus the fragments are connected to the node with a conjunctive connection. Figure 4 shows an example of a clause with three fragments. The conjunctive connection means that if any of the fragments are turned off, the clause will be turned off.

The Unification Constraints. The unification constraints capture possible bindings between terms. The clauses that can potentially resolve constrain possible bindings, and these possible bindings are realized by a set of binding nodes *B*. Bindings that are incompatible are connected by mutually inhibitory connections. Compatible bindings are connected by mutually excitatory connections. For example, in the set of clauses -P(a,b), P(x,y)Q(y,z), -Q(c,d), -P(a,c), the possible bindings are xa, yb, yc, and zd. Of these, compatible pairs are: (xa, yb), (xa, yc) and (yc, zd), and there is one incompatible pair: (yc, yd). This example is simple and does not capture all the constraints possible in unification. At least two others are necessary. These relate bindings between constants and variables. One is that if a variable is bound to a constant and another variable is bound to a different constant, then the two variables cannot be bound to each other. The other constraint is that if a variable is bound to a constant and the same variable is bound to a second variable, then the second variable can be bound to the constant. These constraints are summarized below:

$$x, y : \text{var} ; c, d \text{ const}$$

$$xc \,\&\, yd \Rightarrow \neg\, xy$$

$$xc \,\&\, xy \Rightarrow yc$$

$$xc \Rightarrow \neg\, xd$$

In the network there are potentially $|T|^2$ nodes where T is the set of literals used in the formulae to denote all possible variable-constant pairings. Thus the constraints in above are connected between all relevant groupings. A representative network fragment is shown in Figure 5.

Substitution Constraints. The possible substitutions constrain the network in two important ways. One additional constraint is necessary to link the different bindings together. In the logical formalism this constraint can be derived by observing the potential resolutions between clauses. (In Sickel's notation, these are arcs.) Substitution nodes *S* relate substitutions to bindings. The second constraint relates the substitution nodes to the clause fragment nodes. Each clause fragment node

mentions the terms in its predicate. If these terms are also mentioned in the substitution then there is a two-way positive link between the two nodes. Formally, a node s $\in S$ is positively linked to a node f $\in F$ if f is positively linked to a term $t_1 \in T$ and s is positively linked to a binding t_2x where t_2, x $\in$ T and $t_1 = t_2$. Figure 6a shows the assignment node to relate three bindings between two clause fragments. Since all the assignments must be satisfied, the connections *into* the assignment node are conjunctive.

3. The Resolution Process

The filtering constraints combine to reduce the network to a state where none of the bindings are inconsistent. If there are choices, they are decided arbitrarily. For example, the set $\{c_1$: P(x), c_2: ¬P(a), c_3: ¬P(b)$\}$ results in a network with two inconsistent substitutions: xa and xb. The probabilistic relaxation algorithm will make an arbitrary choice between these two possibilities. Thus the objective of the filtering process is to reduce the network to an *essential state*, wherein only one clause fragment can resolve with its complement. This might seem very restrictive, but the filter network can make many examples into this form. For example, the proof used by Henschen [1976] to introduce resolution can be reduced to this form without using resolution. However, the usefulness of this strategy will have to be tested with many different examples.

Once the network has been put into an essential state, the way constraints are handled can be changed slightly and the network will perform resolution. To do this, three changes are made:

1) the thresholds on clauses are changed so that singleton clause nodes are turned off;
2) the thresholds on clauses are changed so that dropping one fragment input does not turn off the node unless it is the last one; and
3) the binder network is fixed, so that no further changes take place.

The effect of turning off singleton clauses is to remove the fragment associated with their complements. Turning off all the nodes in this fashion is equivalent to finding a proof by resolution.

4. Examples

To describe the process, consider the example where S $\cup$ ¬W is given by: {P(x)Q(y)W(y), ¬W(z), ¬P(a), ¬Q(b)}. The network is shown in Figure 7. Note that all the clauses are essential. Removing any clause will cause all the nodes in the network to be turned off. For example, without clause C_4, f_6 is turned off. This could propagate as follows: ¬Q, Q, a_3, f_5, c_2, f_1, f_2, P, ¬P, s_1, s_2, c_1, f_4, W, ¬W. Of course, many other sequences are possible; the exact sequence in any given case depends on the probabilistic relaxation process.

Let us change the substitutions slightly and see what happens. Suppose c_2 is changed to P(z)Q(y)W(y). This modification is shown as a dotted line in Figure 7. Now the substitution constraint network comes into play. The network is prewired so that if za and yb are turned on, then they will turn off yz. Once this is done its effect will propagate through s_2 to turn off the entire network.

To continue the example, we now describe the resolution phase. Note that in this case turning of all the singleton clauses, c_2, c_3, and c_4, is sufficient to turn off the remaining clause c_1. Note that the clauses in {P(x)Q(y), W(y)¬P(a), ¬Q(b), ¬W(z)} can also be turned off in the same manner, but not those in {P(x)Q(y), W(y)¬P(a), ¬Q(b)¬W(z)}, which has no singletons, or those in {P(x)Q(y)W(y), ¬W(z)¬P(a), ¬Q(b)}.

5. Summary and Conclusions

The implementation of the first order logic constraints results in two coupled networks: (1) a clause network that represents the clause syntax; and (2) a binding network that represents the relationships between terms in different clauses. The method for resolving bindings, unification, can be as complex as the entire inference mechanism. Thus for our purposes we depend on the actual bindings in the KB to have a simple structure.

At the outset, the possibility of reusing clauses was ruled out, but there are some limited cases that can be handled. To see the necessity of reusing clauses, consider {SU ¬ W} = {C_1:P(a), C_2:P(b), C_3:¬P(x)Q(x), C_4:¬Q(a)¬Q(b)}. This can be handled by resolution in a straightforward way. The resolution tree is: ((C_1, C_3), ((C_2, C_3), C_4)). However, note that C_3 appears twice. The consequence of this is that since the unification constraints do not allow xa and xb simultaneously, the network will not pass the filter test. To handle this case we note that both possibilities for C_3 involve constant bindings. Thus we can resolve this by making two copies of C_3: ¬P(a)Q(a) and ¬P(b)Q(b). Once this is done, the inference mechanism will find the proof.

The main intent of this paper has been to force a new look at formal inference mechanisms from the standpoint of performance. Our contention is that models that do not have a parallel implementation are unlikely candidates for models of human inference. This realization may prove catalytic for approaches that try to unify the complementary goals of competence and performance.

The technical contribution of this paper is in the detailed specification of a network and inference mechanism. The network runs in parallel and can handle obvious inferences in first order logic. The running time is bounded from below by O(1) which occurs when all the constraints are local and O(diameter of network) which occurs when the constraints have to propagate the full extent of the network.

6. References

Feldman, J.A. and D.H. Ballard, "Connectionist models and their properties," *Cognitive Science* 6, 205-254, 1982.

Henschen, L.J., "A tutorial on resolution," *IEEE Trans. Computers C-25*, 8, 770-772, August 1976.

Hopfield, J.J., "Neural networks and physical systems with emergent collective computational abilities," *Proc., National Academy of Sciences USA 79*, 2554-2558, 1982.

Kowalski, R., "A proof procedure using connection graphs," *JACM 22*, 4, 572-595, 1975.

Robinson, J.A., "A machine-oriented logic based on the resolution principle," *JACM 12*, 1, 23-41, January 1965.

Sickel, S., "A search technique for clause interconectivity graphs," *IEEE Trans. Computers C-25*, 8, 823-835, August 1976.

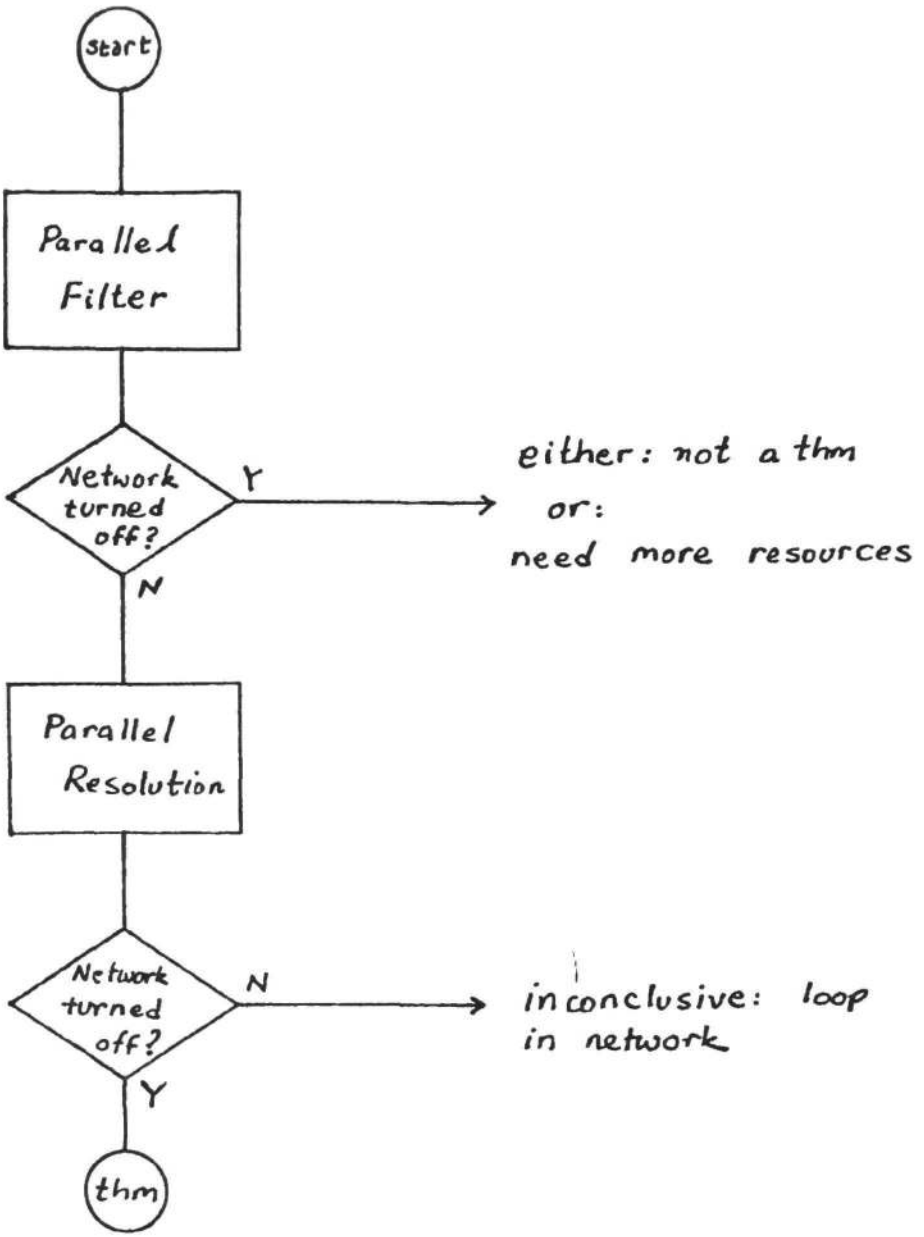

start
Parallel Filter
Network turned off?
Y
either: not a thm
or:
need more resources
N
Parallel Resolution
Network turned off?
N
inconclusive: loop in network
Y
thm

Fig. 1.

Fig. 2

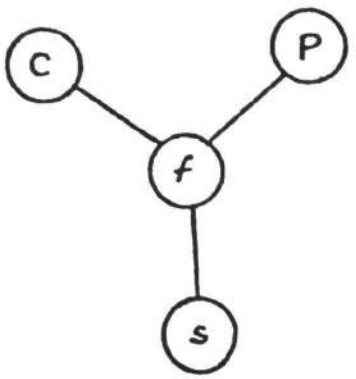


Fig. 3

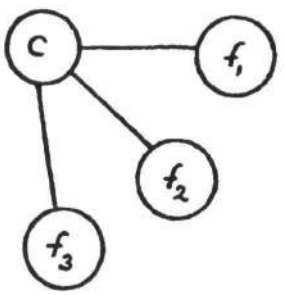
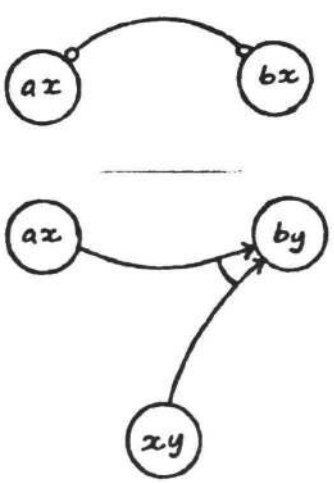


Fig. 4



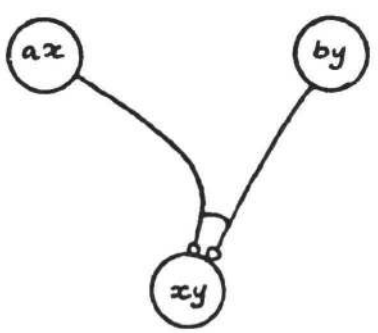
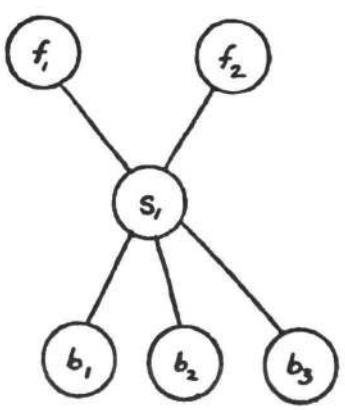


Fig. 5



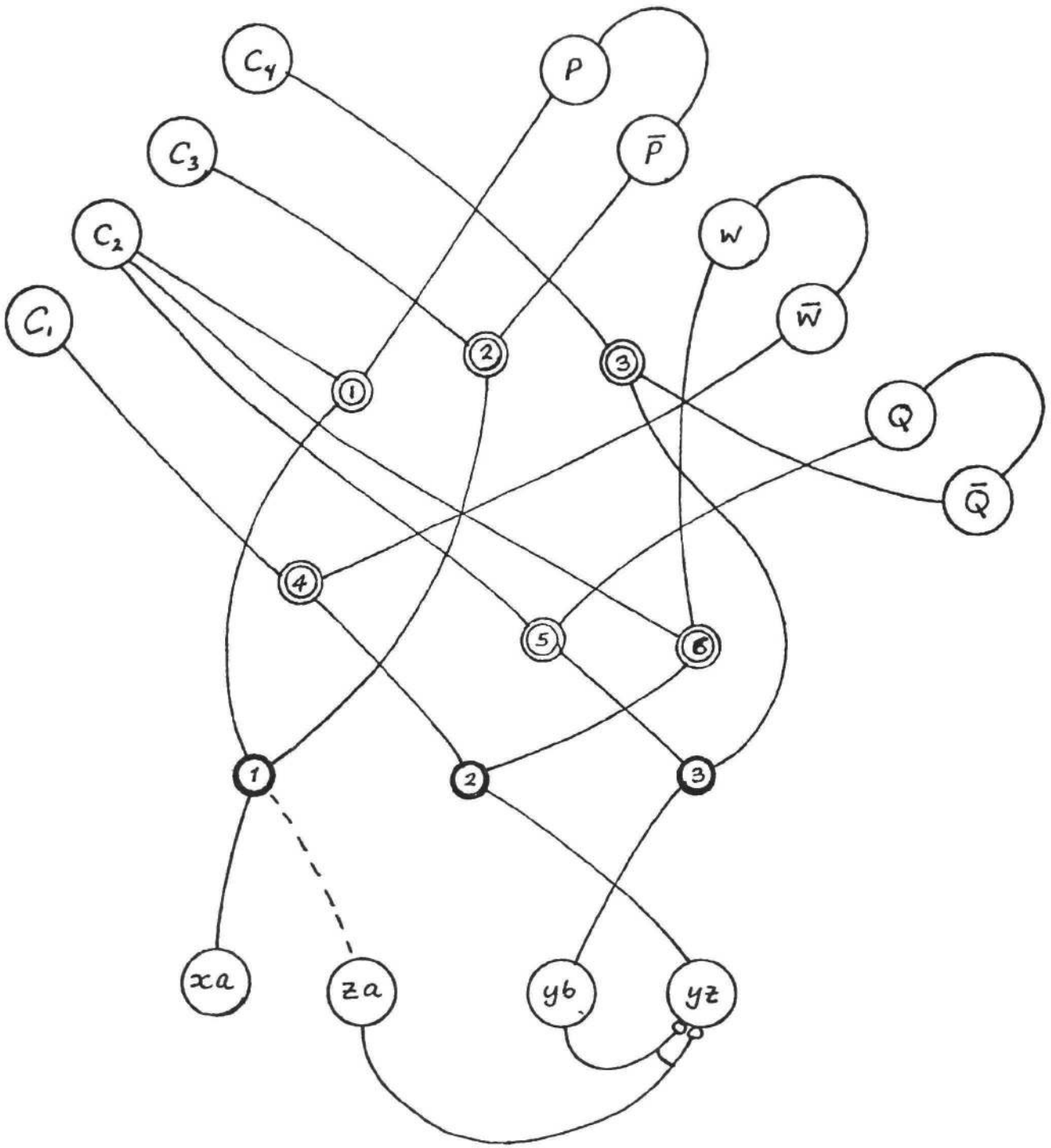


Fig. 7

OPPORTUNISTIC PLANNING AND FREUDIAN SLIPS

Lawrence Birnbaum and Gregg Collins

Yale University
Department of Computer Science
New Haven, Connecticut

Freud's study of the psychology of errors (see, e.g., Freud, 1935), including notably "slips of the tongue," led him to the conclusion that many such errors are not merely the result of random malfunctions in mental processing, but rather are meaningful psychological acts. That is, they are *intentional* actions in every sense of the word, reflecting and indeed *carrying out* the goals, whether conscious or not, of the person who commits them. In particular, Freud argues, such errors stem from attempts to carry out *suppressed* intentions, intentions which have been formed but then in some sense withdrawn because they conflict with other, more powerful intentions.

For example, in the simplest case a person may decide to say something, but then change his mind and decide to say something else instead. Nevertheless, the original intention somehow intrudes itself into his utterance. Freud (1935) discusses the example "Dann aber sind Tatsachen zum *Vorschwein* gekommen," ("and then certain facts were *revealed/disgusting*"), in which *Vorschwein* is a conflation of *Vorschein* (revealed) and *Schweinereien* (disgusting). The speaker relates that he had originally intended to say that the facts were disgusting, but controlled himself and decided to say something milder instead. In spite of this decision, however, the suppressed intention apparently exerted an influence on his speech.

Examples of this sort show that goals, once formed, can influence subsequent behavior despite intervening decisions to suppress them. Viewed from an information processing perspective, however, there are two radically different interpretations of this fact, corresponding to two distinct models of how the influence might be exerted. On one account, no further processing of the goal is undertaken after its suppression, and the influence is simply a residue of the processing that took place prior to that suppression. In the above example, for instance, it may simply be that the prior contemplation of the goal to say the precise word, "Schweinereien," activated that word in memory, and that this residual activation had an effect on the process of choosing what words to say, thus causing the slip. On this account, although the slip does in some sense *reflect* the suppressed goal, it is not really an attempt to carry out the goal.

However, more complex examples show that this sort of residue explanation is not, in general, adequate. Consider Freud's example of the toast "Gentlemen, I call upon you to *hiccough* to the health of our Chief," in which the word *aufzustossen* (hiccough) has been substituted for the word *anzustossen* (drink). In his explanation, Freud argues that this slip is a manifestation of an unconscious goal on the part of the speaker to ridicule or insult his superior, suppressed by the social and political duty to do him honor. However, notice that in this case, in contrast with the simpler example above, one cannot reasonably expect that the speaker's intention to ridicule his superior gave rise originally to a plan involving the use of the word "hiccough." That word can only have been chosen in the course of attempting to retrieve the consciously intended word "drink," to which it bears a close similarity in German. Yet, if we accept Freud's analysis of the example, the word "hiccough" was selected because it achieves the speaker's goal to ridicule his superior. Thus we are forced to conclude that this goal was still

active during the attempt to retrieve the word "drink," *despite the fact that it was suppressed prior to that attempt.*

The mere fact that suppressed goals are able to affect the overt behavior of planners is enough to justify the assertion that they are active. However, the sense of *activity* implied by examples like the above transcends this ability alone. There is no way that a planner could have reasonably anticipated that the goal of ridiculing or insulting its superior would be satisfied by uttering the word "hiccough." If for no other reason, this is because there are hundreds of *a priori* more plausible words and phrases that can be used to insult or ridicule someone. However, if the planner was not looking for *this* opportunity *in particular*, then it must have been looking for *any* opportunity *in general*. In this case, recognizing the opportunity involved realizing that the substitution of the word "aufzustossen" (hiccough) for the word "anzustossen" (drink) would, *within the context of the toast*, result in a ridiculous and insulting utterance. Because the effect of the substitution depends on the context, considerable inference is needed to determine whether it would, indeed, serve to carry out the goal of insulting the superior. Thus, the planner must have expended considerable cognitive resources in checking potential opportunities from the time of the goal's formation to the time that this particular opportunity in fact arose.

But why would a planner expend such resources on a goal *which it had already determined not to pursue*? In fact there is no coherent way to view the planner *as a whole* as the agent behind the expenditure of cognitive resources in the pursuit of suppressed goals. What examples like the above seem to indicate, therefore, is that the goals *themselves* are active cognitive agents, capable of commanding the cognitive resources needed to recognize opportunities to satisfy themselves, and the behavioral resources needed to take advantage of those opportunities. In a very real sense, such goals must be actively *observing* the mental processing being carried out for other goals, not only inspecting features of that processing, but also drawing inferences about how those features might be useful in their own satisfaction. They are not merely, for example, data structures in some monolithic planning system, which could be trivially suppressed simply by being erased or marked as inactive. They must be actively suppressed, and such suppression may in fact fail.

We now come to the central question of this paper: Is the conception of goals and goal processing needed to explain Freudian slips functionally justifiable, or does it merely reflect an accidental attribute of human psychology?

Fundamental to the above explanation of Freudian slips is the ability to recognize and seize opportunities. In Birnbaum (in press), it is argued that this ability is a fundamental element of intelligent planning in general. To take a simple example, suppose you go to the store to buy something. If, while you are at the store, you notice an item that you want on sale, you may then decide to purchase the item, even though you did not originally go to the store in order to satisfy that intention. The point here is that it is not, in general, possible to foresee all the situations in which an unsatisfied goal may be satisfiable. Intelligent behavior requires the ability to recognize and seize such unforeseen opportunities to satisfy goals.

As we saw in the case of Freudian slips, recognizing opportunities may entail significant inference. This is particularly true if we consider people's ability to seize *novel* opportunities. It is easy enough to suppose that some features of situations would point directly to goals that they satisfy. For example, it is arguable that, indexed under the feature "money," we have the goal of possessing money. Thus, it isn't hard to see how the opportunity implicit in seeing some money on the street would be recognized.

On the other hand, suppose a person goes to a hardware store and sees a gadget he did not previously know existed, e.g., a router. People seem perfectly capable, at least sometimes, of

constructing the inferential chain necessary to recognize how such a novel opportunity might facilitate the achievement of a goal that they could not, ahead of time, have known that it would facilitate. For example, someone who had the goal of possessing bookshelves would seem perfectly capable of realizing that a router would be useful in building them. This seems plausible even if he had not intended to build the bookcases, but rather had intended to buy them. In that case, he probably wouldn't have given much thought to how they might be built. But, once he understands what a router does, he may realize that it can be used to cut channels in the side boards of the bookcase, into which horizontal boards can be fitted as shelves.

While the need for this kind of opportunistic processing provides us with a functional justification for the ability of a goal to recognize the means for its own accomplishment when they unexpectedly present themselves, it remains to be explained why goals which have for good reason been suppressed should be able to overcome their suppression when opportunities for their achievement arise. That is, why should an intentional system lack the means to deny such a suppressed goal access to mechanisms for producing real behavior?

Surprisingly, it turns out that opportunistic processing even offers a functional justification for this seemingly unproductive characteristic of an intentional system. Consider first what it *means* for a goal to be "suppressed." A goal would need to be suppressed if it were found to be in *conflict* with another goal in the system. There are two ways that a goal conflict could arise: either because the goals themselves are inherently mutually exclusive, or because some rather more contingent problem arises in attempting to plan for both of them. That is, it might be that two goals are found to be in conflict based on the planner's judgment of the resources and options available under the circumstances in which the goals are being weighed. (See Wilensky, 1983, for an analysis of the considerations involved in making such judgments.) For example, the goal of insulting one's boss is presumably suppressed because it conflicts with more important social and political goals. However, the conflict between these goals is situation-dependent. It is perfectly possible, albeit unlikely, that there may be some future situation in which insulting the boss and achieving one's political ends would be compatible.

Once a goal conflict is recognized, a planner must decide to suppress one goal and pursue the other based on an assessment of which course of action is most reasonable in light of current or expected future circumstances. However, it is quite possible that in fact future circumstances will be different than originally foreseen. Thus an opportunistic planner must be able to override previous decisions about which of its goals to pursue. Decisions made when formulating the plans currently being pursued should not be immutable.

Consider the following example. Suppose a person is out in the forest and is both hungry and thirsty. Given his knowledge about food sources and water sources, and whatever other pragmatic considerations pertain in the circumstances, he decides that these two goals conflict, and that he will suppress the thirst goal while he pursues the aim of satisfying his hunger. While pursuing his plan to obtain food, however, he comes upon a stream which he hadn't previously known about. This is precisely the kind of situation in which we would expect -- or, indeed, demand -- an opportunistic response, regardless of any previous decision to suppress the thirst goal.

The implication here is that the decision to suppress a goal is really just a decision to forego planning for that goal, and that *in an opportunistic processor, no goal is ever really "suppressed."* Viewed in this light, the fact pointed to by Freudian slips, that goals which have putatively been suppressed can still take advantage of opportunities for their own achievement, can not only be understood, but can be seen to be a desirable and possibly necessary aspect of a planner.

What yet remains unexplained, however, is why opportunities would be acted upon even when further reflection by the planner would presumably reaffirm the decision to suppress them, as is undoubtedly the case with Freudian slips. It would seem somewhat counter-productive not to demand that the planner be allowed to reconsider the reasons why a goal was suppressed, in light of the sudden appearance of an opportunity to achieve that goal. We might expect, for example, that despite the opportunity to insult or ridicule one's boss, this opportunity would not be taken, since it would still be impolitic to do so. We might, in fact, assume that this is often what happens. In the case of the hungry and thirsty person, for example, it would make sense for that person to reconsider why he thought there was a conflict between those goals upon finding the stream.

There will not always be time to do this, however. The fortuitous presence of a rock or stick, for example, noticed in the course of a struggle with an animal, is an opportunity which would have to be seized virtually without thought to be helpful. Thus, we might expect that when there is severe time pressure in deciding whether to pursue an opportunity or not, action can be taken without due consideration by the planning mechanism as a whole. Lexical selection, while lacking the life-or-death implications of struggles with predators, is nevertheless a process which must occur in split-seconds to produce smooth vocalizations. We might, therefore, view Freudian slips as an unfortunate but unpreventable side-effect of the need for this kind of opportunistic short-cut to behavior.

In conclusion, we have argued that in order to accept Freud's intentional explanations for slips of the tongue, we must postulate that goals are active mental agents, commanding the cognitive resources needed to recognize opportunities to satisfy themselves, and capable of acting on such opportunities even when suppressed or unconscious. We have further shown that such a conception of goals can be functionally justified on the grounds that it fulfills the requirements of opportunistic processing. In particular, we have seen that the ability of such goals to manifest themselves even after their "suppression" is not merely a flaw in human beings, but is a necessary attribute of an adequate opportunistic processor. Thus, it seems that the kind of intentional machinery needed to support opportunistic planning would quite naturally exhibit Freudian slips.

References

Birnbaum, L. (in press). The role of opportunistic planning and memory in arguments. Submitted to *Cognitive Science.*

Freud, S. 1935. *A General Introduction to Psychoanalysis.* J. Riviere, translator. Liveright, New York.

Wilensky, R. 1983. *Planning and Understanding.* Addison-Wesley, Reading, MA.

Knowledge Structures Involved in Comprehending Computer Documentation

Darlene Clement

Institute of Human Learning
University of California, Berkeley
Berkeley, CA 94720

ABSTRACT

A model of computer-manual comprehension is proposed in which four processes operate simultaneously: task-mapping of the structure of regular procedures onto the structure of computer commands, constructing a mental model of the computer system, inducing the command language grammar, and learning the structure of computer procedures. Findings from a study of five novices' comprehension problems with UNIX documentation are analyzed in terms of these four processes. Two of the four processes—task-mapping and procedure learning—are described in this paper. The analysis focuses on the knowledge structures involved in comprehending a technical text.

The solution to the problem of computer manual comprehension has been narrowly viewed as simply a matter of eliminating jargon, using "good" sentence structure, and so on. That is, the emphasis has been on low-level linguistic aspects instead of the fundamental cognitive ones involving the knowledge structures tapped by a technical text. The comprehension problem may be considered from the perspective of each of the three factors that give rise to it, viz., the system, the reader, and the writer.

First, a computer manual gives directions for operating a device that is unlike any other machine. Computer systems are difficult to learn both because they are operated symbolically, and because they "operate on invisible objects with consequences that are not readily apparent" (Nakatani and Rohrlich, 1983). In particular, the reader must understand both the system's conceptual model and interface.

Second, the reader's knowledge base must be taken into account. If the reader is a non-programmer there are only two things that he or she can bring to the text: a model of how conventional editing is done, and some expectations about text structure. Beyond this, non-programmers are at a loss. They have no understanding of computer programming which some have argued is the basis for a generative model of the system (e.g., Sheil, 1981).

Third, the technical writer's ability to convey the new information must be analyzed. The writer's role is one in which he or she must compensate for both the novice's naiveté, and any unnaturalness in the system's design. To date, writers have only been given very general suggestions for accomplishing this feat (e.g., "write clearly"). The gap between the suggestions and their implementation must be filled entirely by intuition. Cognitive psychology, and in particular text comprehension research, can provide a means of bridging this gap by elucidating the specific schemata tapped by a technical text.

In order to investigate users' problems learning a system with a manual, an in-depth qualitative study was carried out in which 5 novices attempted to learn UNIX[1] with only a manual to guide them. The subjects were asked to read a section of two locally produced tutorials in advance of meeting with the researcher. The tutorials covered file manipulation and text editing with a line-oriented editor called "Edit." During the meetings subjects used the computer to follow the instructions in the tutorial. The 5

[1] UNIX is a trademark of Bell Laboratories.

sessions lasted two hours on average and were tape recorded, yielding approximately 10 hours of tape from each subject.

The model derived from the analysis of the data partitions the information contained in computer manuals into four classes, each with a corresponding comprehension task. **Functional** information describes the purpose of each command and triggers a **task-mapping** comprehension process. In this process, users map the new functional information given in the manual onto their tacit models of regular text-editing and general office procedures. Examples of such preexisting models are the familiar procedures of cutting and pasting text in documents, creating new files, and typewriter editing. **Structural** information describes the underlying structures and processes of the computer system itself. A description of a device triggers a **model-building** process in which the reader attempts to construct a mental model of how the device functions, for example, how the editor buffer and disk (important entities which the user never sees) are related. **Command** information describes the way in which commands are issued. It triggers the **command learning** process in which users attempt to learn the syntax and semantics of the command language. **Procedural** information provides directions for navigating through the system, i.e., knowing which command to issue in which context. The corresponding **procedure learning** process entails recognizing these different program contexts, and learning the order in which commands must be issued.

Though each of these processes taps radically different knowledge structures, they are fundamentally the same: in each case it is necessary for the new information to connect with the reader's knowledge base. It is apparent from the problems novices had that the documentation they used had serious shortcomings in each of these areas. Because of space limitations, only two of the four comprehension processes will be described here—the task-mapping process and the procedure-learning process.

TASK-MAPPING

The task-mapping process was initially described in Clement (1983) as a global process of mapping the structure of the regular editing task onto the corresponding computer version of the task. So, for example, the regular editing procedure of changing a word by crossing it out and writing another word above it, gets mapped to the text editor's substitute command. Recently, this same process has been described in more detail by Moran (1983). Moran states that the user's knowledge of editing procedures consists of at least eight editing functions (add, remove, change, transpose, move, copy, split, join) which operate on five text entities (character, word, sentence, line, paragraph). The 37 tasks that result from the combination of editing functions and text entities constitute the core knowledge the user possesses. This knowledge comprises the "external task space." The computer system also has entities and operations defined within it, but these may be very different from the ones the user knows. The entities and operations internal to the computer constitute the "internal task space." Moran gives the example of a system that defines only one entity (a character string), and only three editing operations. With this system users must learn to conflate the five separate text entities they are familiar with onto this one system entity, and the eight editing functions they are used to must be collapsed onto three: cut, paste, and insert. In other words, the task-mapping process requires that the user learn to carry out familiar tasks by means of unfamiliar functions which operate on unfamiliar entities.

The operation of this process was especially evident when subjects attempted to learn the UNIX **read** command. This command allows a file to be inserted into the file currently being revised, that is, it allows the user to cut and paste. How is it similar to conventional cutting and pasting? In both the computer version and the regular version of the cutting and pasting procedure the point at which the new information is to be inserted must be located. Then the pasting action can be carried out. In the computer procedure the "read" command is issued; in the regular procedure the material is actually pasted in. There are two ways in which the procedures differ. First, in regular cutting and pasting the material pasted in typically no longer exists in its original location. In contrast, in the computer version of the task, the file pasted in still exists as a separate file. Second, in regular cutting and pasting usually only *part* of a remote document is spliced into the document under revision. In contrast, in computer cutting and pasting the *entire* remote file is pasted in, not just a section of it.

The manual described the command as follows.

Reading additional files (r)

The **read (r)** command allows you to add the contents of a file to the buffer at a specified location, essentially copying new lines between two existing lines. To use it, specify the line after which the new text will be placed, the **read (r)** command, and then the name of the file. If you have a file named "example", the command

:$r example
"example" 18 lines, 473 characters

reads the file "example" and adds it to the buffer after the last line. The current filename is not changed by the read command. (*Edit: A Tutorial* p. 22)

In general, the subjects had difficulty understanding this paragraph. After they were told that it referred to cutting and pasting further discussion revealed their attempts to map the structure of the regular procedure onto the computer procedure. Two subjects thought that the file pasted in disappears from its original location. Notice that this is what would be predicted from a model of regular editing. Another subject wondered if only part of the remote file is pasted in, or the whole file.

From a text comprehension standpoint this paragraph from the manual is reminiscent of passages used in text comprehension studies in the early 70's (e.g., Dooling and Lachman, 1972) where subjects were presented with texts that were incomprehensible without a title. Once a title was provided the texts were easily comprehended because *the title triggered the schema that the text was about.* Similarly, this text would have been easier for the subjects to assimilate had the cutting and pasting schema been activated at the outset, say, in the heading. This is a point that can be of use to document developers. Once the appropriate regular-editing schema is activated then the task-mapping process can be carried out more easily. The document developer can further facilitate the task-mapping process by explicitly comparing the similarities and differences between the regular editing procedure and the computer procedure. This would reduce the amount of inferencing the reader would have to engage in, and would simultaneously answer the reader's questions.

PROCEDURE LEARNING

According to the Card, Moran, and Newell (1980) model of the manuscript editing task, an expert's knowledge structure consists of goals, operators, methods, and selection rules. That is, experts have pre-stored information about the sequence of operations and alternative methods available for performing an editing task. It is this knowledge that the novice must acquire from the manual and from interactions with the system.

It is clear from the data that novices come to the procedure learning task with a rudimentary procedure schema containing slots for goals, steps, and methods. However, the process of filling these slots is not easy to do if the role of each piece of information is not clearly marked in the text. For example, one section of the tutorial described how to correct typographical errors with a line-oriented editor. To carry out this task the user must understand three things: 1) how the editor functions (the need to position it on the relevant line); 2) the sequence of steps necessary for carrying out the task; and 3) the various methods that can be used to carry out the task. The structure of the task is as follows:

Goal: Correct typographical error in text.

Step 1: Position editor on relevant line.

Method 1: Search for pattern on relevant line.

Method 2: Type number of relevant line.

Step 2: Issue substitute command.

After performing only the first step in the procedure, two subjects assumed that the task had been completed, i.e., that the correction had been made. This indicates that the manual did not make clear the two-step nature of the task. One subject read about the two methods for carrying out a step and assumed that each method was a necessary part of the sequence. This indicates that the various methods were not clearly marked in the manual as alternatives. After reading the two pages describing the procedure, one subject, after much thought, managed to induce the two-step structure of the task. Together these examples show how much inferencing the subjects were forced to do, and

how difficult it was for them.

This analysis describes novices' attempts to induce the structure of a particular editing task. Yet we know from the research previously described that the expert's knowledge is a finely articulated goal structure in which the steps and methods are clearly differentiated. The writer could facilitate the construction of this structure by simply making it explicit. If the goals, steps, and methods of the procedure were explicitly marked in the text, then the novice would be able to assimilate each piece of information, as it is read, to the appropriate slot in the schema. Like the suggestion put forth in the task-mapping section, this suggestion also reduces the amount of inferencing the reader would have to do.

Research on the schemata novices bring to the text, as well as the schemata ultimately constructed by the expert can lend more precision to the task of *packaging* information in a technical text. In particular, this kind of analysis gives rise to psychologically-based heuristics for document development which address the important conceptual aspects of comprehending computer documentation.

REFERENCES

Card, S. K., Moran, T. P., & Newell, A. Computer text-editing: An information-processing analysis of a routine cognitive skill. *Cognitive Psychology,* 1980 *12,* 32-74.

Clement, D. Comprehending Computer Documentation, unpublished manuscript, (March, 1983).

Dooling, D. J., & Lachman, R. Effects of comprehension on retention of prose. *Journal of Experimental Psychology,* 1971, *88,* 216-222.

Edit: A Tutorial. Documentation produced by the U.C. Berkeley Computer Center.

Moran, T. P. Getting into a System: External-Internal Task Mapping Analysis. *Proceedings of the Conference on Human-Computer Interaction.* Boston, MA., 1983.

Nakatani L. H., & Rohrlich, J. A. Soft Machines: A Philosophy of User-Computer Interface Design. *Proceedings of the Conference on Human-Computer Interaction.* Boston, MA., 1983.

Sheil, B. A. Coping with complexity, (Tech. Rep. CIS-15). Xerox Palo Alto Research Center, 1981.

Ethnic Attitude in Discourse: a Competition-frame Analysis

Marten J. den Uyl Teun A. van Dijk

University of Amsterdam

1.INTRODUCTION

The problems of ethnic prejudice and intergroup conflict have traditionally been a focus of attention in social psychology. In this long research history little effort has been invested in a systematic analysis of the forms of informal conversation in which attitudes towards ethnic groups may be expressed. Yet, everyday talk is one of the most important media for the difusion of ethnic prejudice (e.g. Van Dijk 1984). One reason for the neglect of this topic is that it requires an interdisciplinary approach. Only when insights from discourse analysis, social psychology and cognitive modelling, particularly work on belief systems, are combined, one can hope to be successful in answering the following closely interrelated questions:
a) How are knowledge, beliefs and feelings towards ethnic groups organized? A representation model for etnic belief systems is needed.
b) What processes operate in discourse involving etnic attitudes? This entails in fact two questions: 1) how can discourse production generally be analyzed as a strategic process (Van Dijk & Kintsch 1983); and 2) what strategies in conversation are specific to discourse relating to ethnic minorities?
c) On a social psychological level, the main question is what role informal discourse plays in intergroup relations.

Our analysis of ethnic attitude in discourse is based on an extensive data base drawn from informal interviews conducted in a number of field studies. We cannot deal with the discourse properties of these interviews here (cf. Van Dijk 1984). The (open and unstructured) interviews were held with autochtonous inhabitants of a neighbourhood in Amsterdam with a high percentage of ethnic minorities. The largest minority groups in the Netherlands are immigrant workers (Turks, Maroccans) and people from the former dutch colony of Suriname.

The research described in this paper is sponsored in part by the Netherlands Organization for the Advancement of Pure Research (ZWO). We thank Adri van der Wurff for comments and suggestions.

2.THE STRUCTURE OF ETHNIC ATTITUDE

Ambivalence of Ethnic Attitude. We assume that positive and negative ethnic attitudes could be represented, as a first approximation, by postulating different **goaltrees** (Carbonell 1979) for the "pro" and "con" orientations. However, it has been noted before (e.g Allport 1954) that ethnic attitudes often give an impression of ambivalence and inconsistency. For instance, it is not uncommon to hear someone argue for equal rights at one time and for protection of majority interests a little later. This would seem to imply that the goals that determine positive and negative attitudes can be present simultaneously within an ethnic attitude. This becomes even more likely, once it is realized that these goals are of a different nature; the positive side appears to be based on general values and norms in society. The negative side, we propose, is based on a general, schematic, representation format for intergroup conflict.

Competition-frames. Competition-frames are based on the **Triangle** representation for social conflicts introduced by Schank & Carbonell (1979). A typical fragment from an interview may serve to introduce this notion.

> *"Now these are big houses, you can see that. But they are all foreigners that come to live here. Don't we have any Dutch anymore who need a house? (...) Why should the foreigners have all those nice big houses? They all go down the drain, those houses."* (approximate translation)

In a competition two parties, labelled WE (e.g. "the Dutch") and THEY (e.g. "the foreigners"), are in conflict over some ISSUE (e.g. "the distribution of houses"). A third party, the DISTRIBUTOR determines the outcome of the conflict.
Some important features distinguish perceived competition between groups from most other social conflicts.
- In a competition it is not a single object that is at stake. Rather, the ISSUE is an **ongoing conflict of interests.** This implies that competition might continue as long as the needs of the groups involved remain unchanged and can be settled in a definite way by very drastic means only.
- The groups in a competition need not be proper social actors. Fuzzy categories such as "foreigners" or "autochtonous Dutch" cannot perform social acts. This is an important difference with social conflict Triangles. The latter derive their usefulness as representational devices mainly from the possibility of analyzing social conflicts in terms of a very limited number of basic social acts. Such acts may be nonexistent in the context of competition-frames.
- The DISTRIBUTOR slot in a competition-frame is not always filled by an authority. At the extremes we distinguish "closed competition" where the outcomes are completely determined by some authority that is believed to be both impartial and effective, and "open competition" where no authority is believed to have any influence on the outcomes.
- At any moment the state of a competition can be evaluated. We assume such evaluation always to take place from the WE perspective. The state of a competition is a function of three elements: the CLAIM of each party to a share of what is at stake in the competition; the GAIN, that what each party has received sofar; and the RULES, the distribution rules that the DISTRIBUTOR is perceived to use. An evaluation usually takes the form of comparing the present state of the competition to other (past or future) states of the same competition or to other competitions.
- The essential step in maintaining a competition-frame representation for some

social situation is identification with the WE-group. This identification entails more than the categorization of self as a member of this group, it implies internalizing the group interests at stake as personal concerns. The fulfillment of group interests then becomes equivalent to the fulfillment of personal goals.

Functions of Competition-frames. The most evident function of a competition-frame is that it enables one to explain negatively valued social situations. For most social problems it can be argued that WE are in a bad position because THEY harm our interests. The wide-spread phenemenon of **scape-goating** illustrates the point.

Competition-frames further play a role in the interpretation of particular events called **incidents.** An incident is an event involving members of the competing groups that touches upon the interests of these groups in the competition. Everyday events are understood by creating an episodic structure, a "situation model" (Van Dijk & Kintsch 1983) in which specific information about the event is integrated with general knowledge about the context and background of this event. Part of a situation model are the beliefs about the needs and motives of actors used to explain their actions. A common mechanism for inferring explanations of human action is identification; i.e. we understand someone's actions if we feel we would have done the same in his/her place. Competition-frames enhance identification with WE-group members, but **suppress identification** with members of a THEY-group. When interpreting actions of the latter, the mechanism of identification is replaced by an interpretation process that makes use of group interests and characteristics represented in the competition-frame. Thus, when an event such as a Turkish family moving into a new house is interpreted as an incident in the competition for houses, the individual concerns of these actors are not represented in the situation model. Instead, the perceived threat to in-group interests is represented. The result of such an interpretation process may be illustrated in another citation from the same respondent we quoted before:

> *"Look, and when a Dutchman gets such a big house, then I say yes, that's.. that's great. But why should those foreigners all sneak into those houses?"*

Competition-frames have a function in the organization of knowledge in memory as well. Competitive relations can be perceived to exist between two groups on many different ISSUES, both material (e.g. housing) and immaterial (e.g. religion, power). We assume hierarchical relations can obtain between competition-frames with the same opposing groups. Specifically, we assume that all such frames are dominated by a high level competition-frame where the ISSUE is left unspecified. Such a hierarchical organization of competition-frames is equivalent to a goaltree.

3.COMPETITION-FRAMES IN DISCOURSE

Polarization Strategies. Competition-frames have yet another function: they control a speakers contributions to a conversation.

A mayor aim of informal conversation is *self-expression.* People attempt to express their affective evaluation of the topic under discussion as convincingly as possible, so as to make the hearer share their views. The urge to do so is especially strong, we suggest, when the topic has **personal relevance** for the speaker, and when the speaker considers him/herself to have sufficient **expertise** on the topic.

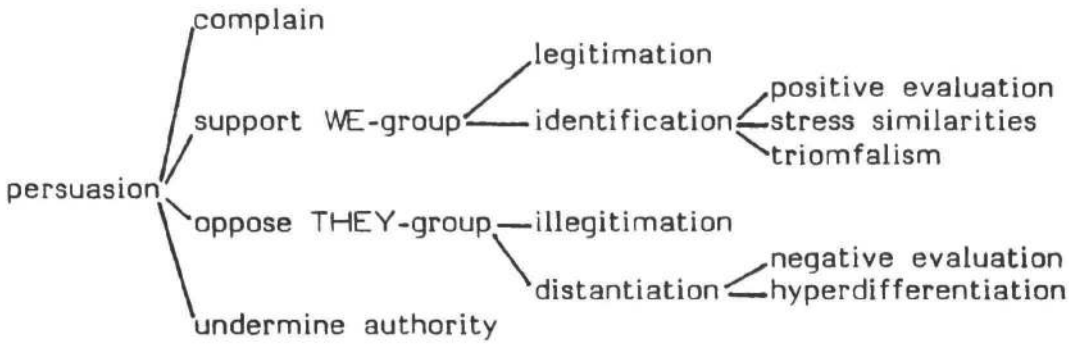
The evaluation of a competition leads to a set of characteristic opinions

regarding the four core elements of the frame:

- The ISSUE is a serious problem.
- WE are in a bad position.
- THEY form a threat to our interests.
- The DISTRIBUTOR is partial and/or ineffective.

The expression of these opinions in conversation is a form of persuasive communication that can be described by a set of *polarization strategies* (see figure 1).

figure 1
polarization strategies



- By **complaining** the seriousness of the ISSUE is stressed. Incidents are frequently used to illustrate how badly group interests have been hurt in the past, but also extrapolations into the future are effective persuasive moves: *"It's getting worse everyday"*.
- **Support WE-group;** in this strategy it is argued that WE are in a bad position and hence are in need of support. One form this takes is **legitimation,** the legitimacy of the CLAIMS of the WE-group is elaborated upon. Another form is the enhancement of positive **identification** with the WE-group, which branches into a number of substrategies.
- **Oppose THEY-group;** this is the most central polarization strategy. The basic argument is that since THEY form a threat to our interests THEY should be opposed more firmly. Two substrategies here are the inverse of support strategies: in **illegitimate** the CLAIMS of the opponents are played down; in **distantiation** the evaluative distance to the THEY-group is increased. Two major aspects of distantiation are **negative evaluation** of the THEY-group and **hyperdifferention,** the underscoring of presumed deviant characteristics.
- **Undermine authority,** refers to attacks against authorities that function as DISTRIBUTOR in the competition. It should be noted that a competition-frame interpretation of social situations almost unavoidably leads to an evaluation of authorities as partial and ineffective.

a Dilemma. People in informal conversation have more concerns than self-expression. Another important concern in most social situations is *self-presentation.* People prefer to present favorable images of themselves. In discussing ethnic relations these two concerns can easily come into conflict. Negative evaluations may backfire if insufficiently substantiated. Explicit discrimination and racism are not only generally frowned upon, but actually punishable by law. The strategic problem our respondents face, then, is how to express their negative evaluations of minorities, without appearing prejudiced. This dilemma is apparent in many ways in conversation. Negative remarks are frequently introduced with "positive" phrases such as *"I don't have anything against foreigners but..."*. Negative generalizations are often presented jokingly or in a highly exaggerated form. However, the most popular solution

to the strategic problem is telling stories.

Strategic Storytelling. A considerable part of informal discourse on ethnic minorities consists of storytelling. The topic of most of these narratives is a minor everyday incident. On closer analysis some consistent patterns become evident in these seemingly innocent stories.
First, almost all stories concerning minorities have negative complications. Second, almost always members of minority groups are held responsible for the unpleasant events. Third, actors from these groups hardly ever get any further introduction; they are only presented as representative members of their groups. Explanations for their behavior are left implicit, or refer to stereotypical group characteristics. Fourth, the solution category is often missing. The negative impact of the story is thereby enhanced. Sixth, in the coda a generalizing conclusion is drawn from the narrative: *"That happens to us all the time."* All these features can be explained on the assumption that the main strategical aim of such stories is distantiation.

4. THE ROLE OF DISCOURSE IN INTERGROUP RELATIONS

It may have been noted that a vicious circle is inherently present in what we have outlined sofar: competitive interpretations of societal problems trigger the use of polarization strategies, which in turn enhance the belief in competitive analyses, and so on. There are counterforces, such as the system of equalitarian values that, in combination with the concern for self-presentation, also has its influence on discourse.
In ethnic attitudes pro and con orientations are in dynamic balance. Intra-group conversation can be conceived of as the *social switch mechanism* that regulates the degree of antagonism between groups. Detailed analysis of discourse can unravel the many factors that at any moment may tip the balance. These factors include the presence of ISSUES, candidate problems for a competitive interpretation; the -lack of- confidence in authorities that function as DISTRIBUTOR; the saliency of potential THEY-groups.
Most important, perhaps, is identification with the WE-group. Recent social psychological theories assume that social identification is the result of social comparison processes motivated by the need for positive social identity (e.g. Tajfel 1982). This may be correct for the minimal group paradigm commonly employed in experimental research. An analysis of natural discourse, however, favors the hypothesis that people identify with groups in order to acquire a sense of control over everyday concerns.

REFERENCES

Allport, G. W. (1954) *The Nature of Prejudice.* New York: Addison-Wesley.

Carbonell, J. G. (1979) *Subjective Understanding.* New Haven: Yale University Ph. D. Diss.

van Dijk, T. A. (1984) *Prejudice in Discourse.* Amsterdam: Benjamins (in press).

van Dijk, T. A. & Kintsch, W. (1983) *Strategies of Discourse Comprehension.* New York: Ac. Press.

Schank, R. C. & Carbonell, J. G. (1979) The Gettysburg Address, representing Social and Political Acts. In: Findler (ed.) *Associative Networks* New York: Ac. Press.

Tajfel, H. (ed.) (1982) *Social Identity and Intergroup relations.* Cambridge: Cambridge U.P.

Mood, Emotion and Action: a concern-realization model.

Marten J. den Uyl Nico H. Frijda

University of Amsterdam

1.INTRODUCTION

A promising starting point for attempts at a further understanding of the relations between affect and cognition are the effects of *moods* or *feeling states* on cognitive processes. From recent reviews of these effects (e.g Bower 1981; Clark & Isen 1982) it can be concluded that at least the following cognitive functions are influenced by feeling states:
memory; "Mood State Dependent Retention" (MSDR) refers to the observation that recall of information is facilitated if the mood states at the time of learning and of recall correspond, but is inhibited if these moods are different.
selection; the "Mood Congruity Effect" (MCE) refers to the increased saliency of mood-congruent materials; processing of information is facilitated, when the affective valence of this information is congruent with the mood of the subject.
production; it has been found that subjects in production tasks tend to generate responses that are congruent to their mood. E.g. in free association and in interpretive tasks angry subjects give more angry responses.

2.A NETWORK THEORY

Bower (1981) proposes an extension of general semantic network theories to explain these results. In his theory each distinct emotion is represented by a specific node in the memory network. All information specific to a particular emotional state is assumed to be linked to the node for this emotion and thus can generate activation towards this node. Moreover, episodic representations of events in which this emotion was experienced are also connected to the emotion node.
The explanation in this model for the MSDR phenomenon is that mood functions as a retrieval cue; the mood during learning will become linked to the episodic representation then formed. If the same mood is experienced at the time of recall, the emotion node will generate activation towards this episodic representation.
The selection and production effects are also explained by the principle that mood spreads activation to related information in memory. This principle could explain selective learning of mood-congruent materials in a number of ways. For instance, mood-congruent materials might become connected to more information in memory, or might be elaborated upon more.
The mood effects on production are implied by the assumption that increased activation of mood-related materials increases their availability.

A question unanswered sofar is how emotion nodes become activated. Bower

& Cohen (1982) present a model of emotional appraisal of events that addresses this question. They extend the network model by adding a *blackboard control structure*. This control structure facilitates interaction and integration of information from different knowledge sources by allowing all sources to put hypotheses on the blackboard. The contents of the blackboard are changed by a set of production rules. Each production has a left-hand side (LHS) that can be matched against the contents of the blackboard, and a right-hand side (RHS) that changes the plausibility values of the hypotheses on the board.
Emotional appraisal of events is the result of the application of *emotional-interpretation* (E-I) rules. In the LHS of E-I rules a cognitive interpretation of events is specified, the RHS specifies some adjustment of the activation level of emotion nodes. As an example of the use of E-I rules consider the event of "a big boy Sam hitting a smaller boy Johnny". This event could match an E-I rule with LHS "a bully hurts a weakling". This rule then assigns emotional interpretations to the components of the event, e.g. it associates anger with Sam and sympathy with John. As a result both the anger and the sympathy emotion nodes become more highly aroused.
Bower & Cohen assume the existence of large numbers of E-I rules that vary in generality and are hierarchically organized. Thus, many E-I rules may be applicable to any given situation, but the most specific E-I rules are favored. However, the selection of E-I rules is also dependent on their momentary priority ordering.
Interactions between emotional states and emotional interpretations of events -one reacts differently to a slight annoyance in a good or a bad mood- are accounted for by postulating *interaction rules*.

2.1. problems for network theories

The main problem with the network theory is that, even with the blackboard as extension, it is at best only half a theory of emotion. The question remains unanswered what the implications are of the emotional interpretation of an event for the observer. There is little explanatory power in postulating emotion nodes for each emotion. The model may predict that emotional states influence the processing of information that has a corresponding emotional quality; the model does not predict what information will have that quality.
The problem of pinpointing what information one can expect to be affected by a given mood manipulation is highlighted by some recent experimental results. Johnson & Tversky (1983) had subjects estimate the frequency of fatalities due to various causes. Before making estimates, experimental subjects read a story describing one fatal event in some detail. The striking results of these experiments were, that experimental subjects rated all risks considerably higher than control subjects, but did not overestimate risks related to the mood inducing story more than unrelated risks. As Johnson & Tversky (1983) conclude:

> The pervasive global effect of mood and the absence of a local effect pose a serious problem to memory-based models of this effect, such as spreading activation within a semantic network. In such models ... Risks that are closely linked to the story should be influenced more than unrelated risks, contrary to the present findings.

We claim that two assumptions are needed to explain these results. A first assumption is that mood selects information not on the basis of some affective quality, but on the relevance of information to planning. Specifically, we propose:
in positive moods the processing of knowledge concerning the success of goal-based action is facilitated, while in negative moods information relating

to failure is more readily processed.
Thus, in a depressed mood all we know about unattainable goals, unsuitable instruments, incompetent actors and bad outcomes comes easily to mind. In an elated mood, on the other hand, we are experts on success.
However, there is a problem associated to this proposal: it seems very hard to envisage an associative memory model wherein information relating to success of certain plans can be activated independently of information relating to failure of those same plans. A solution to this problem is to assume that it is not the *data-base* but the *logic-engine* that is the source of mood effects.
We propose that:
the main locus of mood effects in cognitive processing lays not in the activation patterns in the general semantic network, but in the 'tuning' of certain inferential procedures that operate on the network.
We will briefly outline a general theory of emotion, to show how these two assumptions could be incorporated in a model of mood effects.

3.A CONCERN-REALIZATION MODEL OF EMOTION

Emotions can be regarded as changes in action readiness elicited by significant events (Frijda 1984; i.p.). Action readiness can change in three ways: change in *activation*, the general readiness for action; change in *inhibition*, the blocking of action readiness; and changes in the readiness for specific kinds of actions. Such a specific readiness or "impulse" is called *action tendency*.
Action tendencies can be conceived of as the highest level of planning (i.e. *meta-plans*, Wilensky 1981). They differ from plans in that they are steered by the goal of changing the current situation, rather than by that of achieving an anticipated state. For example, the action tendency *attack* is controlled by the goal of changing the mode-of-existence of the object of the attack in the present situation, not by the goal of decreasing the well-being of that object to some specifiable degree.
The significance of events is determined by *concerns*. A concern can be defined generally as a disposition to prefer certain states of the world over others. In the computational context of a blackboard control structure, a concern is represented by a *Concern-Realization Rule* (C-R rule). A C-R rule consists of a LHS and a RHS part:
the LHS contains a description of a preferred state, the concern proper. The preferred state is matched against the interpretations of situations posted on the blackboard. This matching process can signal three types of event relevance: the event can be *congruent* to the concern, resulting in positive emotions, or *discordant* with the concern, for negative emotions, or *relevant*, without being evidently congruent or discordant; the latter is the condition for attention and interest.
The RHS of a C-R rule increments or decrements specific action tendencies, general activation, or inhibition.

The concern-realization model further assumes the existence of a general planning mechanism that transforms action tendencies into plans and puts these to action. Following Wilensky (1981) we distinguish three classes of rules in the planning mechanism:
Proposer Rules take in action tendencies and propose plans to effectuate them;
Projector Rules take in plans and test the feasibility of these plans. Projector rules may produce scenarios of potential failures of the plan.
Revisor Rules attempt to revise plans so as to overcome difficulties projector rules have come up with.

The crucial difference between the present proposals and the Bower & Cohen (1982) model is that the concern-realization model is action oriented. C-R rules map events into specific, object-related, action tendencies. We suggest that this feature enables one to account for the emotional appraisal of events with a far more coherent and comprehensive set of rules. Some appreciation of this point may come from a further reflection on the earlier example of "bully Sam hitting little Johnny".

Imagine that the mothers of both Sam and Johnny happen to be watching this event. In the network model Johnny's mothers appraisal of this event will, for any likely set of E-I rules, lead to a strong activation of both the "anger" and the "sympathy" nodes, originating from anger with Sam and sympathy with Johnny. It is possible, if perhaps unlikely, that some set of E-I rules would lead to the same degree of activation of these nodes for Sam's mother, but with Sam as the single source of this activation. (E.g. Sam's mother is angry for Sam staining his clothes, she is sympathetic to his show of strength and she doesn't care about Johnny.)

If *"A person's current emotional state may be described as the activation level of a set of N emotions like fear, anger ..."* (Bower & Cohen 1982) then this would seem to imply that both mothers are in the same emotional state. Intuitively, this does not seem right. It should make some difference whether the objects of anger and sympathy are different or the same. In the concern-realization model the reactions of both mothers can be described in terms of two action tendencies, *attack* and *support*, roughly corresponding to anger and sympathy respectively. For Johnny's mother, these two action tendencies are in this situation virtually identical (i.e. they share the same goal of changing Sam's presence in the situation). For Sam's mother however the action tendencies "attack Sam" and "support Sam" clearly conflict.

The general point should be clear. It is one of the essential features of nodes in an associative semantic network that they can accumulate activation, but do not represent where that activation came from. Thus, any model that represents emotion in terms of the activation of emotion nodes will have a problem in representing the distinction between true emotional ambivalence (which is always based on the presence of conflicting action tendencies) and *emotional complementarity*, that results from an action tendency implying different orientations towards different objects. Solution to this problem will be entirely ad hoc unless an action component is added to the model.

4.MOOD RECONSIDERED

There exists considerable confusion over the meaning of the concepts "mood" and "emotion". Sometimes no distinction appears to be made at all (e.g. Bower 1981), sometimes the distinction is only a matter of degree (e.g. Clark & Isen 1982). We risk to add to this confusion in arguing that mood in fact is not a feeling state at all. Feeling states refer to the experience of concern relevance and of action readiness, including interoception of activation and inhibition. Mood, we suggest, refers not to action readiness and concern relevance, but to the *tuning of the concern-realization system*. We have mentioned in passing that the application of E-I rules is dependent on a momentary priority ordening. The same holds for planner- and C-R rules. The tuning of the system then, is the momentary priority ordering of the rules that make up the system.

Our earlier hypothesis on the relation between mood and knowledge about success or failure can thus be further understood. We have assumed that the planning mechanism entails Proposer and Projector rules. A Proposer is by nature an optimist, it generates plans that might succeed. A Projector,

however, is a pessimist: its task is to demonstrate how plans may fail. We propose then, that elated and depressed moods correspond to high priority assignments to Proposer and Projector rules, respectively. This proposal finds some support in a study by Isen & Nowicki (1981). These authors found that subjects in a creative problem solving task generated more ideas and more often reached a solution after seeing a funny movie.

REFERENCES

Bower, G. H. Mood and Memory. *American Psychologist*, 1981, 36, 129-148.

Bower, G. H. & Cohen, P. R. Emotional Influences in Memory and Thinking: Data and Theory. in: M. S. Clark & S. T. Fiske (eds.): *Affect and Cognition*, New Jersey: LEA 1982

Clark, M. S. & Isen, A. M. Toward understanding the relationship between feeling states and social behavior. In: A. Hastorf & A.M. Isen (eds.): *Cognitive Social Psychology.* Amsterdam: Elsevier 1982.

Frijda, N. H. Towards a Model of Emotion. In: C.E.Spielberger,J.Sarason & P.E.Detares (eds.): *Stress & Anxiety* Vol 1, N.Y. Hemisphere 1984 (in press)

Frijda, N. H. *The Emotions* New York: Cambridge U.P. (in preparation).

Isen, A. M. & Nowicki, G. P. Positive affect and creative problem solving. Paper presented at the annual meeting of the Cognitive Science Society, Berkeley, 1981.

Johnson, E. J. & Tversky, A. Affect, Generalization, and the Perception of Risk. *Journal of Personality and Social Psychology* 1983, Vol.45, 20-31.

Wilensky, R. Meta-Planning. *Cognitive Science* 1981, 5, 197-233.

Toward a Reader-Based Model of Thematic Comprehension

Marcy Dorfman
Coordinated Science Laboratory
University of Illinois

1. INTRODUCTION

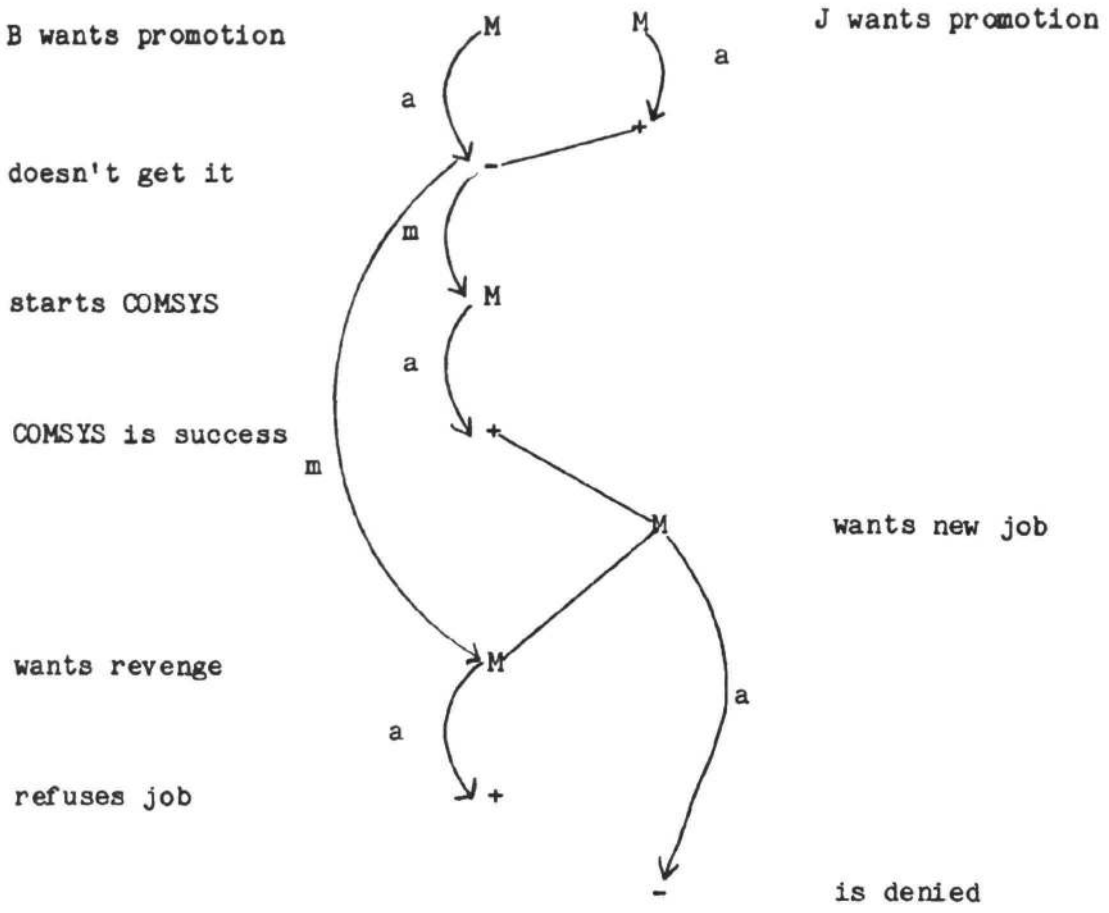
During the past decade, researchers in psychology and artificial intelligence have become increasingly interested in the plans, goals, motivations, and emotions of story characters [Schank and Abelson, 1977; Wilensky, 1978; Dyer, 1982; Lehnert, 1982; Wilensky, 1983]. An inevitable outcome of this process has been an increasing concern with what makes stories interesting [Wilensky 1983], enjoyable [Brewer, 1982; Brewer and Lichtenstein, 1982], and thematically significant [Dyer, 1982; Lehnert *et al.*, 1982]. However, despite recent progress in analyzing stories, current story processing models will have little success in recognizing high-level constructs such as themes unless they include, in addition to a model of the characters' plans, goals, motivations and emotions, a reasonably complete model of the reader's belief system. Ideally, the reader's belief system should enable the reader to understand the author's point in writing the story. For example, if a story is ironic, the reader must understand the author's implicit beliefs well enough to know when the author is being sarcastic or satiric. Similarly, if a text is didactic, the reader must understand the author's attitude toward moral and immoral actions in order to grasp the point of the story.

In this paper we propose an alternative to current story processing models which includes, in addition to a model of the characters' motivations and actions, a model of the reader's belief system. Since our model is intended to account for the comprehension of didactic stories, such as fables, the reader's belief system is based on the concept of a "just world" [Lerner, 1980]. Lerner's "just world hypothesis" describes a system of moral justice in which it is intrinsically satisfying for good actions to triumph over evil actions. In the context of the story world, the just world hypothesis predicts that stories will be morally satisfying when "good" characters experience positive outcomes and "bad" characters experience negative outcomes. However, in order to assign characters positive or negative valence relative to the story's outcome, it is first necessary to compare each character's actions to a set of normative rules that describe appropriate conduct. To illustrate the importance of this process in determining the point of a story, we will compare the predictions made by our model to story understanding systems recently developed in artificial intelligence.

2. PREVIOUS AND RELATED WORK

Since the point of a story depends, in part, on the actions and emotions of story characters, an adequate model of story understanding will have to include a level of analysis that is able to represent what characters actually do and feel. Researchers in artificial intelligence have made several important contributions to this endeavor. For example, Dyer's [1982] system BORIS includes, among other knowledge sources, a component for representing the emotions and interpersonal interactions between story characters. Similarly, Lehnert's [1981; 1982] work on plot units provides an excellent notational device for capturing the emotional reactions of characters to narrative events. Lehnert suggests that narratives can be represented by three affect states generated by the dimensions Desirability and Attainment. These states include emotions associated with positive events (+), negative events (-), and mental states (M). In addition, Lehnert specifies four types of connections, or affect links, that describe an oriented arc between two states. MOTIVATION links (m), for example, connect positive or negative states to mental states, while ACTUALIZATION (a) links connect mental states to positive or negative outcomes. TERMINATION (t) and EQUIVALENCE (e) links connect mental states to other mental states, or events to other events. According to Lehnert, states and links can combine to produce affect units that are capable of representing different types of plot structures. An example of how Lehnert would analyze a simple story based on retaliation is presented below. The COMSYS Story

John and Bill were competing for the same job promotion at IBM. John got the promotion and Bill decided to leave IBM to start his own consulting firm, COMSYS. Within three years COMSYS was flourishing. By that time John had become dissatisfied with IBM so he asked Bill for a job. Bill spitefully turned him down.



Using Lehnert's summarization algorithm, we can generate the following summary based on the pivotal unit RETALIATION:

"Because John was promoted over Bill at IBM, Bill started his own company, and later refused to give John a job when he asked for one."

While the plot unit analysis provides an appropriate summary for the "COMSYS Story" (summarization was, in fact, Lehnert's goal), a summary is not as general as the story's point or theme. For example, if we were asked to generate a moral for COMSYS we might say that "one bad turn deserves another," replacing specific actors and actions in the story by an abstract general statement. In addition, a summary is not really evaluative, in that it does not tell us whether the actors or actions in the story were good or bad or right or wrong. Moral judgments of this kind, while triggered by the characters' motivations and actions, are typically embodied in the belief system of the reader. In the following section, we will describe an augmented model of story understanding that uses plot unit representations in conjunction with the reader's belief system in order to make predictions relevant to the point of the story.

3. A READER-BASED MODEL OF NARRATIVE UNDERSTANDING

The model we propose contains three levels of analysis (see Figure 1). The first level is based on Lehnert's notion of a plot unit, and represents the intentions, behaviors, and affective states of the story

characters. The second level of analysis represents the just world belief system of a hypothetical reader. During story processing, the reader (or story processor) compares the actions of story characters represented at the first level against just world rules describing normatively appropriate behavior. If a story character commits a moral transgression, that character's valence changes from 0 (neutral) to -. Conversely, if a story character upholds a normative rule, that character's valence changes from 0 (neutral) to +. Character valence is the final level of analysis that is necessary to generate predictions relevant to the point of the story.

Based on the preceding description, the analysis in Figure 1 can be summarized as follows. The first plot unit, COMPETITION, tells us that John has succeeded in defeating Bill. However, although John experiences positive affect as a result of success, the just world rules tell us that, if John causes negative affect for Bill, either Bill or some unknown agent will cause negative affect for John. Thus, although the character John experiences positive affect, the reader assigns negative valence to John and predicts a negative outcome for John in the future. The next plot unit, SUCCESS BORN OF ADVERSITY, does not satisfy the prediction spawned by COMPETITION, and spawns no further predictions of its own. However, the plot unit RETALIATION, in which Bill causes negative affect for John, satisfies the prediction spawned by COMPETITION. (A satisfied prediction is marked by an asterisk in the character valence column.) Once an active prediction has been satisfied, the point of the story is generated by forming a causal link between the antecedent of the just world rule that spawned the prediction (If X --> -Y) and the event satisfying the prediction (Y --> -X). This base rule corresponds to the moral of the story suggested in the previous section: "One bad turn deserves another."

4. CONCLUSION

The purpose of this paper has been to establish the importance of a non-character-based belief system in generating the moral or point of a story. The main difference between the model proposed here and other story understanding systems, such as Lehnert's, is that Lehnert uses plot units to generate the summaries of stories, while we view plot units as a means to generate predictions relevant to the point of the story. Although a representation of the characters' plans, goals, behaviors, and emotions is a necessary component of any story understanding model, the level of character analysis, while appropriate for the purpose of summarization, is not sufficient to generate the point of the story. Understanding story points requires both the ability to make moral judgments about the actions of story characters, and to use such judgments to make further predictions about character outcomes. A prescriptive belief system that describes normatively appropriate behavior is requisite in both of these cases.

An interesting side effect of a multi-leveled belief system is that readers and characters often experience different reactions to story events. For example, in the "COMSYS Story" we saw that, although John experienced positive affect in defeating Bill, the reader assigned negative valence to John. Similarly, although John experiences negative

affect at the conclusion of the story, the reader feels morally satisfied because the just world prediction B --> -J is fulfilled. As long as such differences in expectations exist, one needs to have a theory of story understanding that includes both reader-based character-based information.

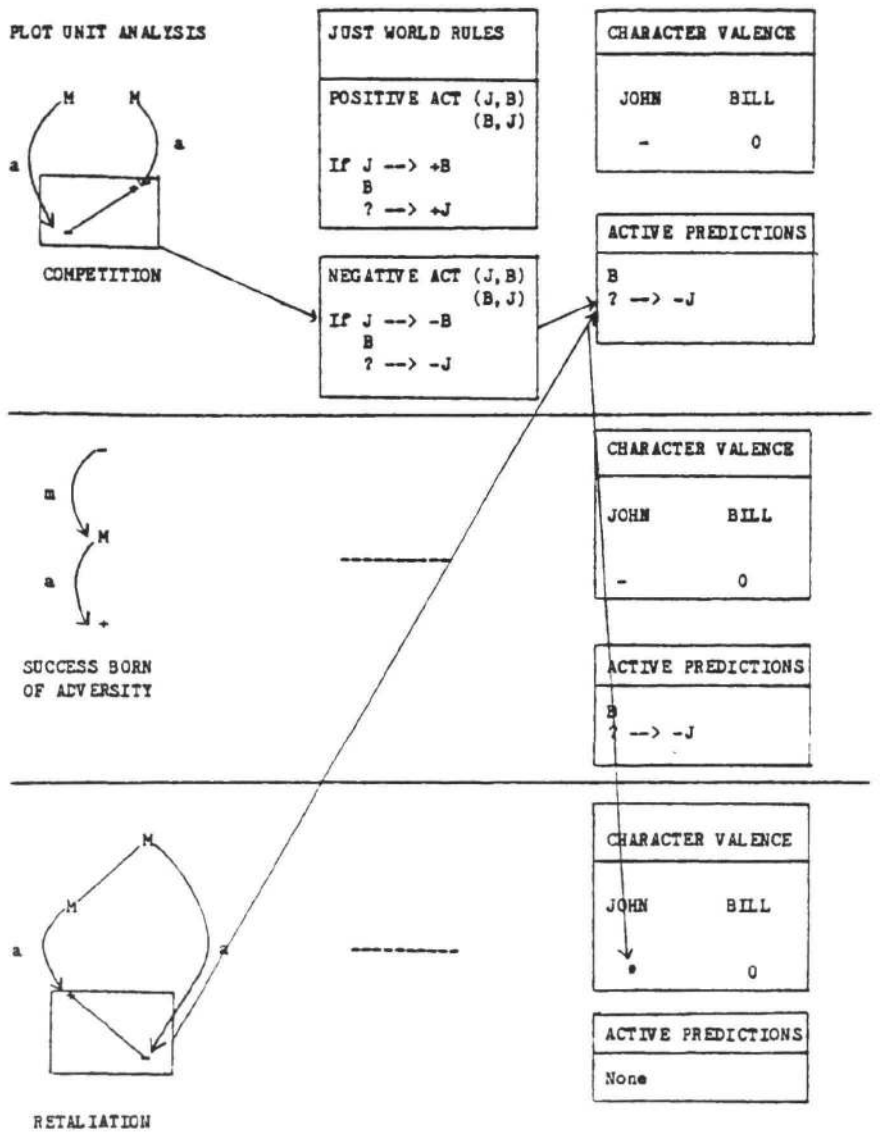


Fig. 1. Multi-level analysis of the "COMSYS Story"

5. REFERENCES

Brewer, W. F. (1982). Plan understanding, narrative comprehension, and story schemas. Proceedings of the National Conference on Artificial Intelligence, 262-264.

Brewer, W. F., & Lichtenstein, E. H. (1982). Stories are to entertain: A structural-affect theory of stories. Journal of Pragmatics, 6, 473-486.

Dyer, M. G. (1982). In-depth understanding: A computer model of integrated processing for narrative comprehension. New Haven, CT: Yale University Department of Computer Science, Research Report No. 219.

Lehnert, W. G. (1981). Plot units and narrative summarization. Cognitive Science, 4, 293-331.

Lehnert, W. G. (1982). Plot units: A narrative summarization strategy. In W. G. Lehnert & M. H. Ringle (Eds.), Strategies for natural language processing (pp. 375-412). Hillsdale, NJ: Erlbaum.

Lehnert, W. G., Dyer, M. G., Johnson, P. N., Yang. C. J., & Harley, S. (1983). BORIS - An experiment in in-depth understanding of narratives. Artificial Intelligence, 20, 15-62.

Lerner, M. J. (1980). The belief in a just world: A fundamental delusion. New York: Plenum Press.

Schank, R. C., & Abelson, R. P. (1977). Scripts, plans, goals, and understanding. Hillsdale, NJ: Erlbaum.

Wilensky, R. (1978). Understanding goal-based stories. New Haven, CT: Yale University Department of computer Science, Research Report No. 140.

Wilensky, R. (1973). Planning and understanding. Addison-Wesley.

Wilensky, R. (1983). Story grammars versus story points. The Behavioral and Brain Sciences, 6, 579-623.

WHY DOES A SAILBOAT GO WITH A POSTMAN? SCRIPT JUSTIFICATIONS OF 5-YR-OLDS & ADULTS

S. Farnham-Diggory
University of Delaware

This report addresses the general question of how we expand or contract schemata of everyday events to take account of new factors. Schemata for mail delivery in suburban settings seldom include sailboats, but if asked "How could a sailboat go with a postman?" you could easily think of several ways -- the postman could go sailing on his day off, mail could be delivered to islands on sailboats, walking around to deliver mail and sailing are both forms of movement, etc. Alternatively, you could if necessary exclude items that are normally part of everyday scripts. Asked "How could a letter not go with a postman?" you would probably cite computer mail -- but note that what you did was generate a whole alternative script around the letter. The questions before us: Can general strategies for script expansion or contraction be identified? And, if so, do they develop?

By way of background, in addition to the sparse current literature on script development (e.g., Nelson, 1978; Schank, 1982) there is an older literature on over- and under-inclusion stemming from attempts to diagnose defective thinking from categorization behavior (e.g., Goldstein & Scheerer, 1941). This was eventually absorbed into the more complex models of category decision making that we have today (Rosch & Lloyd, 1978; see especially the article by Miller, 1978).

Procedure

The studies reported here used the following procedures: subjects first saw a slide of an everyday scene familiar even to 5-yr-olds (e.g., a postman delivering mail, a girl combing her hair, a baby in a tub, a dentist in his office, etc.). Following the scene, subjects were shown a slide of an object which bore (in the opinion of judges) a varying degree of relationship to the scene. Type A objects (e.g., postman: letter; baby: towel) were clearly part of the scene though not visible in the preceding slide; Type B objects (postman: mailtruck; baby: sink) were less clearly related; Type C objects (postman: sailboat; baby: baby carriage) could not be part of the scene; and Type D objects (postman: tricycle; baby: stove) were even more remote. There were 10 scenes and 4 types of objects for each, randomly interspersed.

There were free choice and forced choice studies. In the free choice study, subjects decided whether or not an object went with a scene, and then explained the basis of their decisions. Decision times from onset of the object slide to the subject's

"yes" or "no" vocalization were obtained. Justification protocols were then requested.

In the forced choice study, subjects were required to give reasons why all objects either did (inclusion condition) or did not (exclusion condition) go with the preceding scene. Only justification protocols were collected, decision time variance being typically overwhelmed by response set factors.

Results

In the free choice study, the pattern of inclusion and exclusion decisions were predictable functions of the degree of object relatedness (Type A through D). Of main interest were the nature of the justifications, the decision times that preceded them, and developmental effects. Justifications fell reliably into 3 overall categories:

1. Given Script justifications -- the initial scene is the referent and the object is tested for inclusion suitability. E.g., given Girl Combing Hair followed by Pliers:

> Adult: No... She wouldn't be using pliers for any sort of grooming.
>
> Child: No... She doesn't use the pliers to brush her hair.

2. New Script justifications -- a new script is generated around the object; this new script becomes the referent, and the agent of the preceding scene is tested for inclusion suitability. E.g., given the above sequence:

> Adult: No... They're rather sophisticated cutting tools which I don't think a child would use.
>
> Child: No... Because she isn't cutting any flowers.

Note that the role of the object in both Given and New scripts can be classified as object of a direct action, instrumental, or part of the general scene.

3. Dual Script justifications -- both given and new scripts are constructed, and their compatibility is tested. If it is decided they are incompatible, exclusion results. E.g., Girl Combing Hair followed by Book:

> Adult: No... A little girl combing her hair probably wouldn't be very interested in reading a book, at the same time anyway.
>
> Child: No... Because she can't read a book if she's combing her hair.

However, subjects could also construct a higher-order script that relates the given and new scripts. E.g., Dentist at Work on a Patient followed by Kleenex box:

> Adult: Yes... Because if the poor fellow starts crying from the pain he'll have to blow his nose and wipe his eyes.
>
> Child: Yes... If he wants to blow his nose while the dentist cleans his teeth he could.

As the examples suggest, children produced about the same percentage of Given Script, New Script and Dual Script justifications as adults did. However, children were more likely to base Given Script justifications on the direct object role of the item being tested.

Decision times for both groups reflected the mental sequence of first representing the given script, then generating the new one, and then comparing the two. In the adult group but less so in the children's group there was also an effect of negation.

The forced choice study corrected for scene-specific response patterns. If either inclusion or exclusion were required for all scene-object pairs, would the same frequency of justifications appear?

In general, across the object types (A to D), children were like adults in both inclusive and exclusive Given Script justifications. However, they were far more likely than adults to use New Script justifications, and less likely to use Dual Script justifications.

The general implication is that 5-yr-olds are well-equipped with the logical reasoning mechanisms needed for expanding and tuning schemata for everyday events. Their conceptual development would thus be primarily a function of the experiential opportunities available to them.

References

Goldstein, K. & Sheerer, M. (1941) Abstract and concrete behavior: an experimental study with special tests. Psychological Monographs, 53, No. 2.

Miller, G. A. (1978) In E. Rosch & B. Lloyd (Eds.) Cognition & categorization. Hillsdale: Erlbaum.

Nelson, K. (1978) How children represent knowledge of their world in and out of language: a preliminary report. In R. S. Siegler (Ed.) Children's thinking: What develops? Hillsdale: Erlbaum.

Rosch, E. & Lloyd, B. (1978) Cognition & categorization. Hillsdale: Erlbaum.

Schank, R. C. (1982) Reading and understanding. Hillsdale: Erlbaum.

Interactive Student Modelling in a Computer-based Lisp Tutor

Robert G. Farrell
John R. Anderson
Brian J. Reiser

Advanced Computer Tutoring Project
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Students have extreme difficulty learning their first programming language. This difficulty is magnified by the learning environment - a cold terminal, an unforgiving textbook, and an inaccessible teacher. The student may be entirely lost until a teaching assistant or more experienced student volunteers his or her expertise. We estimate that private instruction is somewhere between two and four times as effective as classroom instruction. Our research program is aimed at finding those aspects of private tutoring that can be implemented as a computer program, so that we can provide automated private tutoring to a large number of students.

In this paper we describe an initial version of a computer-based tutor for LISP that incorporates some of the ingredients of good human tutoring. We will first describe how we have applied our previous cognitive modelling work to the domain of intelligent tutoring. We then describe the structure of our tutoring system for LISP. We show how this tutor makes learning easier by conveying the problem-space structure, reducing working-memory demands, and directing problem-solving. Finally, we report the results of some initial evaluation studies with our tutor and give some future directions for our research.

Interactive Student Modelling

To interactively model a student, a tutoring system must continually recognize students' goals and their procedures to achieve those goals. We developed a Goal-Restricted Production System (GRAPES) to model how novices write LISP functions (Sauers and Farrell, 1982) to achieve programming goals. We have used GRAPES to construct an ideal model that incorporates all of the correct procedures that students might use in a particular tutoring session. During students' problem-solving, the tutoring system must discover which procedures or deviations from procedures a student is actually using. We also developed a **knowledge compilation** mechanism for GRAPES which accounts for the difference in performance between novices and advanced students (Anderson, 1983). Using knowledge compilation, we can adjust instruction according to the student's

learning rate.

Tutorial Interaction

Our LISP tutor consists of a domain-independent interpreter that incorporates tutoring strategies, a set of LISP programming rules for modelling the student, a set of tutorial rules that analyze student code and provide feedback, and various problems characterized by an initial goal and a problem statement.

Our tutoring system interacts with the student by first explaining a LISP problem-solving goal; the student reads the goal description and enters an answer that should achieve that goal. If the student's choice is acceptable, the tutor pursues the chosen path and generates more problem-solving goals. If the student's choice is unacceptable, the tutor explains why the choice was incorrect and permits the student to try again. If the student cannot generate a good answer, the system will explain the best possible response.

We plan to use our tutor to teach a short course in LISP, including basic structures and functions, function definition, conditionals and predicates, helping functions, recursion, and iteration.

Conveying Problem Space Structure

Producing a program in any language consists of a medley of algorithm design, coding, and debugging (Brooks). A good human tutor can converse with the student in a variety of problem spaces. In this section we describe how our tutor communicates in the problem-spaces involved in algorithm design and coding. We are not concerned with debugging since we never allow the student to produce an final solution that is incorrect.

Our tutor currently utilizes four problem spaces for coding and algorithm design:

* The LISP coding problem space is used in normal problem-solving. The student enters LISP code in a syntax-based editor. The hierarchical structure of the problem is represented by symbols to be expanded. For instance, (cons <1> <2>) tells the student that he or she can choose to produce code for either <1> or <2>.

* The means-ends analysis space is used when the student is having trouble producing code for a problem that can be characterized by a set of successive operations on an example. The student produces a solution by supplying LISP operators that reduce differences between the current state and the goal state in the example.

* The problem decomposition space is used when the student is having trouble producing code for a problem that can be easily decomposed into pieces. The system displays a menu of possible decompositions of the problem and

the student must pick a correct decomposition.

* The case analysis problem space is used when the student is having trouble producing code for a problem that has a decomposable input-output behavior. The student specifies an action for each input-output case and then produces code that achieves all of the actions.

Reducing Working Memory Demands

In previous work (Anderson, Farrell, and Sauers, 1983) we estimated that half of students' time spent solving programming problems is spent recovering from working memory failures. A good human tutor constantly reminds the student of the information necessary to solve the problem that the student is attempting. Our tutor reduces working memory demands in the following ways:

* The tutor always displays the problem statement in a separate window.

* The tutor displays the entire student answer and that portion of the answer that is correct so far.

* The tutor provides a tree-structured help facility that describes all of the LISP operators that the student has learned so far.

Directing Problem Solving

Novices spend a large amount of time exploring incorrect solutions that result in little learning. A good human tutor directs the student toward correct answers, while still letting the student learn from mistakes. Lewis and Anderson (1984) have shown that students learn more slowly when they are given feedback about their erroneous applications of operators only after a delay.

Our tutor directs problem-solving by first focusing the student on a single problem-solving goal. The tutor formulates a query that directs the student to supply a particular piece of LISP code to do a specific task. The tutor can supply examples to illustrate a sample input or output to the code it is requesting.

The tutor keeps the student from generating incorrect solutions by providing immediate feedback on errors. If the student cannot solve a problem subtask after a small number of tries, the tutor explains the best answer. Both explanations and queries are generated by instantiating natural language patterns associated with each rule or goal. The resulting english is modified by a set of transformational rewrite rules to enhance readibility.

Conclusion

We have constructed a computer-based tutor for LISP based upon some abilities of good human tutors. The tutor can interact with the student in a number of different problem spaces, corresponding to different student solution strategies. Our tutor reduces working memory demands by use of pop-up windows and directed dialogue. It also directs problem-solving by immediately intervening when a student generates an unacceptable answer. Our system interactively models the student by updating a set of production rules. We have performed an evaluation study on our tutor which confirms our belief that our tutor is about twice as effective as classroom instruction, but is only half as effective as a good private tutor. We plan to further test the pedagogical effectiveness of our tutor by automating a short LISP course taught in the fall of 1984.

Evidential Inference in Activation Networks

Jerome A. Feldman and Lokendra Shastri
Computer Science Department
University of Rochester

Introduction

Psychological and biological results suggest that many cognitive tasks like visual recognition, categorization and associative retrieval do not take more than 100 computational steps. This follows because typical neuronal firing rates are a few milliseconds and the response time of cognitive agents during numerous experimental tasks is a few hundred milliseconds. Given that most cognitive tasks require access to a large body of information, the above observation imposes a major constraint on the manner in which conceptual information may be organized and accessed by cognitive processes. In particular it seems to preclude an interpreter that examines the knowledge base. This paper briefly outlines a framework for organizing and accessing conceptual information that appears to offer several advantages over previous work [Fahlman 79]. The proposed framework suggests an evidential semantics for knowledge and describes how the above may be encoded as an active and massively parallel (connectionist) network [Feldman & Ballard 82]. The resulting system has been run on simple examples and is capable of supporting existing semantic network applications dealing with problems of recognition and recall in an uniform manner. The framework also provides a natural way of representing "inconsistent" or conflicting information and using it in making inferences. It embodies an important class of inference that may be characterized as working with a set of competing hypothesis, gathering evidence for each hypothesis and selecting the best among these. A detailed treatment of this framework appears in [Shastri & Feldman 84].

Overview

In the proposed framework, conceptual knowledge is organized as a network of active elements which interact with one another via controlled spreading of activation. The information encoded in the "memory" network is accessed via other network fragments, each of which is a connectionist encoding of a *routine*. We present a simple example to introduce the notation and the overall framework. Figure 1 depicts the interaction between a fragment of an agent's restaurant routine and a part of his memory network. The routine fragment decides whether some food goes well with red wine on the basis of the food's taste. A routine is represented as a sequence of nodes (units) connected so that activation can serve to sequence through the routine. In the course of their execution, routines pose queries to the memory network by activating relevant nodes of the memory network. The memory network returns the answer by activating appropriate units in the routine. We depict action steps as oval-shaped nodes, queries as hexagonal nodes and answer nodes as circular nodes. In this routine fragment, the task of deciding on a wine results in a query to the memory network about the taste of food and the decision is made on the basis of

the answer returned by the memory network. Answer nodes in a routine mutually inhibit each other and the answer node receiving the maximum activation from the memory network triggers the appropriate action. The memory network in the example encodes the following information:

HAM and YAM are two concepts in the domain.
Concepts in the example domain are characterized by two properties, *HAS-TASTE* and *HAS-FOOD-KIND.*
HAM is SALTY in taste and is a kind of MEAT, YAM is SWEET in taste and is a kind of VEGETABLE.

Each arc in the network represents a pair of links, one in either direction. The triangular nodes associate objects, properties and property values. Each node is an active element and when in an "active" state, sends out activation to all the nodes connected to it. A node may become active on receiving activation form another node in the memory network or a routine node. Triangular nodes behave slightly differently in that they become active only on receiving simultaneous activation from a pair of nodes.

The crude deroutineion given above is sufficient to demonstrate how simple recognition and retrieval tasks may be handled by such networks. To find the taste of HAM a routine would activate the nodes *HAS-TASTE* and HAM. The triangular node b1 linking *HAS-TASTE* and HAM to SALTY will receive coincident activation along two of its links and become active. As a result, it will transmit activation to SALTY which will ultimately become active. Figure 2 shows the activation levels of various nodes during the processing of the above query. If a routine needs to find an object that has a salty taste it would activate the nodes *HAS-TASTE* and SALTY. This will cause the same triangular node to become active and transmit activation to HAM. Eventually, HAM will become active completing the retrieval. The two examples roughly correspond to how retrieval and recognition may be processed by the network. In the rest of the paper we will focus on representational issues and hope that the example discussed above will give the reader an idea of the dynamics of network operation.

Representational framework

The semantic information forms a *conceptual structure* defined over a space spanned by *conceptual attributes.* All domain knowledge is defined in terms of these attributes and their values. Examples of attributes are: has-shape (with values such as round, triangular), has-color, is-an-instance-of and is-a-part-of.

The primary level of organization in the conceptual structure is in terms of *Concepts.* These are *labelled* clusters of "coherent" <attribute, value> pairs. The value of an attribute is also a Concept and hence Concepts may be arbitrarily complex. Concepts may refer to different sorts of things in the domain such as individuals, categories, events, properties, locations and relations. Attributes are classified into two broad categories: *PROPERTIES* and *structural links.* Properties correspond to the intrinsic features of Concepts and may vary from domain to domain. Thus, physical objects may have properties like *HAS-SHAPE* and *HAS-COLOR,* while events may have properties like *HAS-LOCATION* and *HAS-DURATION.* Structural links are fairly

domain independent and define "inheritance-like" inference paths. The most representative of these is the *is-an-instance-of* link that is used to organize information in hierarchical structures in semantic networks. Our formulation employs an extended notion of property inheritance and includes other structural links such as the *is-a-part-of* and the *occurs-during* links [Allen 83] besides the *is-an-instance-of* link.

Concepts are classified into Types and Tokens. Tokens refer to instances and Types refer to abstractions defined over Tokens. Abstractions may in turn be defined over Types to yield more abstract Types, or a Type may be differentiated to result in more refined Types. In this framework, a Type is not viewed as a set and its structure is similar to that of a Token viz. a labelled collection of <attribute, value> pairs. The *is-an-instance-of* structural links encode the relation between a Token and a Type while *is-instantiated-by* links encode the inverse relationship.

We use a graphical notation for the representational framework. Figure 3 displays a sample network encoding the following information:

"Birds are a kind of Things, Swan is a kind of Bird, Hansa is a Swan, Things have the property color, Swans are generally White and White is a Color."

The representation uses three kinds of nodes: the Type node, the Token node and the Binder node. Arcs in the network represent bidirectional links. Type and Token nodes label clusters of <attribute, value> pairs, each of which is represented by a Binder node. For instance, b1 represents the fact: "Things have the property color", while b2 represents the fact: "Swans are generally colored white" i.e. "the value of the property color for Swans is generally White". The framework permits associating properties as well as property values with concepts. For example, we may represent that fruits have color without specifying any particular color values.

A weight is associated with each link and these provide the basis for the evidential semantics of knowledge. A link from node A to node B may be interpreted to mean "A provides evidence for B". Consider the links from Type nodes to their Binder nodes. The weights on these links provide a way of encoding the strength of generalizations represented by a Type. Thus, the link from SWAN to b1 in figure 3 is a quantitative measure of the evidence provided by the assertion "x is a Swan" to the assertion "the color of x is White". Cases with more than one typical value are easily represented as shown in figure 4. If red is a more typical color of Apple than green, the weight w1 will be greater than w2. The use of weights has other interesting consequences. For instance, if the node Apple is activated (the network is "imagining an Apple") activation from the node Apple will drive the associated Binder nodes. The Binder nodes corresponding to the most typical property values will receive the highest activation resulting in the activation of what would amount to a virtual Token corresponding to the most typical instance of the Type. Thus, the color of the imagined Apple will more likely be red than green. In this framework, the representation of a Type does double duty and acts as if it were a prototypical representation [Rosch 75], besides being an abstract representation of a class of Tokens.

The use of weighted links from a Type to its Binders provides a more natural

interpretation of "exceptions" and "cancellations" and gives a clean semantics of the *is-an-instance-of* link. In this framework, one cannot both say: "All Swans are White" and "Giselle is a Swan whose color is black". However, one may say: "Most Swans are White" and "Giselle is a Swan whose color is black". This is illustrated in Figure 5. The crucial point is that Giselle may not be attached as an instance of Swan unless the weight of the link from Swan to b2 is reduced to a value less than 1.0. In Figure 5 the link from Swan to b2 is a statement of typicality and hence has a weight less than 1.0, whereas the link from Giselle to b3 encodes a definite statement and hence has a weight of 1.0.

Just as weights on links from concepts to Binders were significant, the weights on links from Binders to Concepts also serve an important function in categorizing an instance (assigning a Type to a collection of <attribute, value> pairs). The weights on links from Binders to Concepts can be used to assign a metric to the significance of a match between the <attribute, value> of a Type and that of an instance. The process of categorization easily translates into a "best fit" situation. Each Type receives evidence from Binders that match the input data. Type nodes accumulate this evidence and their level of activation provides a quantitative measure of the goodness of match. The Type with the highest activation wins [Feldman 82]. Furthermore, this also provides an interpretation of the notion of a prototypical instance of a category. If the property values of an instance match the typical values of the Type then the occurrence of this instance results in the higher activation of the Type node. Consequently, such an instance appears to be more prototypical. Thus, a Robin matches the properties in the representation of the Type Bird more strongly than a Penguin.

Conclusion

The representation and use of conceptual knowledge remains a core issue in cognitive science. This paper presents an approach to these problems that appears to offer several advantages over previous work. The basic ideas of evidential reasoning, multiple hierarchies and connectionist implementation fit together remarkably well and could form the basis for a detailed modeling of how knowledge is handled in natural systems.

References

[Allen 83] Allen James F. Maintaining knowledge about Temporal Intervals. *Commun.* ACM 26, 1983, 832-843.

[Fahlman 79] Fahlman Scott E. *NETL: A System for Representing and Using Real-World Knowledge.* The MIT Press, 1979.

[Feldman & Ballard 82] Feldman Jerome A., Ballard Dana H. Connectionist Models and their Properties. *Cognitive Science*, 6, pp 205-254, 1982.

[Feldman 82] Feldman Jerome A. Four Frames Suffice: A provisionary model of vision and space. TR99. Computer Science Department, University of Rochester, September 1982.

[Rosch 75] Rosch E. Cognitive Representations of Semantic Categories. In *Journal of Experimental Psychology*: General 104, 192-233, 1975.

[Shastri & Feldman 84] Lokendra Shastri, Feldman Jerome A. Semantic Networks and Neural Nets. TR 131. Computer Science Department, University of Rochester, January 1984.

Learning and memory in machines and animals: An AI model that accounts for some neurobiological data[1]

Richard H. Granger
and
Dale M. McNulty

Artificial Intelligence Project
Computer Science Dept.
Cognitive Sciences Program
and
Center for the Neurobiology of Learning and Memory
University of California
Irvine, California 92717

ABSTRACT

The CEL model of learning and memory (Components of Episodic Learning) *[Granger 1982, 1983a, 1983b]* **provides a process model of certain aspects of learning and memory in animals and humans. The model consists of a set of asynchronous and semi-independent functional operators that collectively create and modify memory traces as a result of experience. The model conforms to relevant results in the learning literature of psychology and neurobiology. There are two goals to this work: one is to create a set of working learning systems that will improve their performance on the basis of experience, and the other is to compare these systems' performance with that of living systems, as a step towards the eventual comparative characterization of different learning systems.**

Parts of the model have been implemented in the CEL-0 program, which operates in a 'Maze-World' simulated maze environment. The program exhibits simple exploratory behavior that leads to the acquisition of predictive and discriminatory schemata. A number of interesting theoretical predictions have arisen in part from observation of the operation of the program, some of which are currently being tested in neurobiological experiments. In particular, some neurobiological evidence for the existence of *multiple, seperable memory systems* in humans and animals is interpreted in terms of the model, and some new experiments are suggested arising from the model's predictions.

1. Introduction to the problem

1.1 Characterization of learning processes

The amnesic patient identified by his initials 'H.M.' is apparently incapable of learning any new information; since the operation that removed a part of the limbic system of his brain, he has been unable to learn to recognize new people or situations. For instance, he re-introduces himself to his doctor Brenda Milner every time she visits him, even though she has visited him many times a week for many years! In contrast, his pre-operation memories appear not to be impaired, nor is his ability to carry on a relatively normal conversation or other everyday functions.

However, H.M. *can* acquire certain categories of new abilities. For instance, he has been tested on the 'mirror-writing' task of writing while seeing only a mirror image of what he writes. Every time the experimenter came in the room, once a day for several weeks, H.M. had to be re-introduced to both the experimenter and the experiment, and insisted that he had of course never seen either before, and that he didn't know how to do this (mirror-writing) task. *Yet his performance on the task improved steadily over the several-week period; in fact, he learned the task at about the same rate that control subjects did.* When confronted with examples of his poor early trials compared with his much-improved recent trials, he is unable to explain how the differences arose, and doesn't remember ever performing those experiments.

These and other results in humans and animals have led inescapably to the hypothesis that there are *multiple memory systems*, i.e., separable biological systems that semi-independently establish

[1] This research was supported in part by the National Science Foundation under grant IST-81-20685, and by the Naval Ocean Systems Center under contract N66001-83-C-0255.

long-term memories from experience. As suggested by H.M.'s behavior, these two systems have distinct characteristics; i.e., each is only capable of learning certain types of information.

However, an accurate *characterization* of precisely *which* tasks are learnable by which mechanism has proven elusive; there currently exist a number of competing neuropsychological hypotheses characterizing the different memory systems (see e.g., *[Squire 1980; Squire, Cohen and Nadel 1982; Mishkin 1982]*).

A long-term goal of the research described here is to attempt to characterize these different learning systems in terms of the types of learning behavior they produce. Our current subgoal is to create a system in which the behavior of learning and memory systems can be characterized. We hope to be able to build two different systems out of similar functional components, each of which has a particular set of learning abilities. We would then be able to show what differences in the models gave rise to the differences in learning abilities.

Recent research on learning and memory in AI has focused primarily on advanced human abilities (see e.g., *[Schank 1983; Schank and Burstein 1981; Lebowitz 1982; Kolodner 1983; Carbonell 1982; Langley 1981, 1982; Mitchell 1981]*). We have adapted some ideas on MOPs *[Schank 1983]* and the indexing of E-MOPs *[Kolodner 1983a, 1983b]* to the tasks we are modeling. Our focus is on much lower-level domains of learning and memory, especially 'subcognitive' tasks that lower mammals (e.g., rats) can learn. This has enabled us to concentrate our models on the *processes* underlying learning and memory, rather than on complex memory *structures*; our approach has been to attempt to identify a candidate set of mechanisms sufficient to allow the acquisition, storage and retrieval of simple episodic information, and to compare our results against experimental data on learning and memory processes in animals.

A key point here is that these 'subcognitive' learning and memory tasks, as far below 'higher' human abilities as they are, are nonetheless still difficult and elusive, and therefore eminently worthy of being the focus of an AI learning mechanism. These and related tasks have been extensively studied by cognitive psychologists and neurobiologists in their experimental approaches to learning and memory; yet their theories of human and animal learning and memory have been insufficiently precise to allow for the construction of computer models for testing the theories. Still missing is a bridge between AI models of learning, and psychological and neurobiological experiments on learning.

1.2 Introduction to CEL

The **CEL** model of learning and memory *[Granger 1982, 1983a, 1983b]* provides a process model of the acquisition and operation of certain aspects of learning and memory in animals and humans. The model conforms to constraints provided by relevant results in psychology, neuropsychology and neurobiology; a number of behavioral data are explained in terms of the model, and certain specific theoretical lesions and modulations of the model predict behavioral effects that correspond to observed behaviors in similarly manipulated animals.

Parts of the model have been implemented in a computer program called **CEL-0**. **CEL-0** takes as input a sequence of experiential sensory events coded in terms of sensory modality and feature sets. The program operates on the inputs, building a memory database of information derived from the input streams. Sample domains that have been worked on include a simple 'feeding' microworld in which the model learns to predict (via classical conditioning) which events reliably and predictably lead up to its being fed; and a 'maze' microworld in which the model explores and learns (operantly) to identify where 'interesting' and rewarding areas of the maze are, and to create a simple 'cognitive map' *[Tolman 1932, O'Keefe and Nadel 1979]* of the maze environment. In the maze microworld, **CEL-0** interacts with MazeWorld, a simulated 'maze environment' program, that receives **CEL-0** input moves, and returns a value indicating **CEL-0**'s new location in the maze; hence, each move of **CEL-0** causes 'feedback' from the simulated maze, which in turn triggers **CEL-0**'s next move.

Some of **CEL-0**'s unexpected behavior in the MazeWorld has triggered some new theoretical ideas which are presented here. For instance, we have identified seven different categories of learning, i.e., seven different ways that new memory traces can be created in **CEL-0**, each corresponding to a different 'calling sequence' of operators, each of which in turn seems to correspond to a logical class of training situations that might arise in the real world. These seven classes of learning will be briefly discussed in a later section of this paper.

Other examples of theoretical ideas that have arisen from working with the program in the MazeWorld include: a mechanism for active 'exploratory behavior' during learning, a mechanism for creating subgoals from goals during learning, acquisition of 'landmarks' during learning that serve as useful index points, and a comparison of 'efficient' learned behavior vs. 'superstitious' learned behavior; some of these are described in some detail in *[Granger and McNulty 1984]*.

Attempts to find detailed correspondences between the model and experimental data in neurobiology have so far been fruitful. A number of specific predictions arising from work with the model are in the disparate areas of selective attention, modulation of memory, and rapid forgetting and learning deficits associated with certain limbic lesions (*[Granger 1983b]* presents detailed analyses of these three substantive areas of CEL's modeling efforts).

This paper will present first a description of CEL-0's behavior in the MazeWorld simulation, and then a specific neurobiological prediction dealing with multiple memory systems; the prediction is currently being tested in a neurobiological lab at the Center for the Neurobiology of Learning and Memory at Irvine.

2. Introduction to the model: The twelve CEL operators and their functions

The **CEL** model proposes a characterization of the constituent functional operators that comprise learning processes, in the hope that these primitive operators may each have specific instantiations that can be identified in the neural substrate. The model identifies a set of twelve 'primitive' memory operators which operate in parallel to collectively perform five classes of memory manipulation: reception, recording, retrieval, reconstruction and refinement. The model consists of the operation of these twelve operators on memory representations we term episodic schemata. Detailed descriptions of the functions of these operators and their (often nonintuitive) interactions are provided in *[Granger 1982, 1983a, 1983b]*.

In brief, the twelve operators have the following functions:

- Reception operators:

  DETECT - *set of sensory input channels and any associated hard-wired preprocessing performed by those input channels, such as visual and auditory processing;*
  SELECT - *'tunable' input filter to selectively attend to some inputs over others on the basis of prior experience;*

- Recording operators:

  NOTICE - *matches inputs against known desirable and undesirable states; triggers COLLECT when a match occurs;*
  COLLECT - *packages recent stream of inputs into a kernel episodic schema;*
  INDEX - *creates new indices, and hooks into existing indices, for each new episodic schema;*

- Retrieval operators:

  REMIND - *matches inputs against indices for existing schemas; triggers ACTIVATE when match occurs;*
  ACTIVATE - *incorporates REMINDed schemas into current predicive schema; triggers the Reconstruction operators;*

- Reconstruction operators:

  ENACT - *performs any efferent actions in current predictive schema; 'tunes' SELECT's filter to attend to predicted afferent events;*
  SYNTHESIZE - *matches inputs against predicted events in current predictive schema; triggers Refinement operators to modify schema in response to matches and mismatches;*

- Refinement operators:

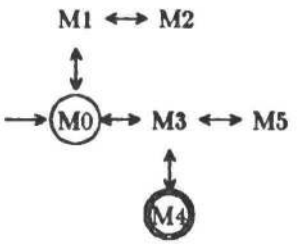
  REINFORCE - *incrementally strengthens current schema(ta) according to SYNTHESIZE's judgment of its successful predictiveness (i.e., matches);*
  BRANCH - *creates a branch in current schema(ta) according to SYNTHESIZE's judgment of unsuccessful predictiveness (i.e., mismatches);*
  DETOUR - *creates a non-pursuable branch in current schema according to NOTICEings of undesirable events, predicted or not.*

These operators act in parallel, asynchronously and semi-independently in the **CEL** model, and complex interactions among them at run time enable these relatively straightforward operators to give rise to a rich set of learning and memory behaviors.

3. A brief example of the operation of the CEL-0 program

3.1 Introduction to the CEL-0 environment

The setting described here for CEL-0's operation is a relatively simple maze, that CEL-0 moves through by interacting with MazeWorld, a simulated 'maze environment' program. MazeWorld receives CEL-0 input moves, and returns a value indicating CEL-0's new location in the maze; hence, each move of CEL-0 causes 'feedback' from the simulated maze environment, which in turn triggers CEL-0's next move. Following is a schematic view of the relatively simple MazeWorld maze that we will use for the examples in this section; 'M0' is the entry point into the maze, and 'M4' contains water, which will be used for a 'reward' under circumstances to be described later.



The following sections describe a connected set of examples of CEL-0's operation in this maze. The description will be in three phases:

Phase 1 ('Exploration' phase): CEL-0 uses 'innate' (built-in) episodic schemas to move through the maze, establishing episodic traces corresponding to its 'routes' through the maze.

Phase 2 ('Effectiveness' phase): CEL-0 has an added desired state (satisfy-thirst) that drives its behavior; it searches for and finds a (not necessarily most efficient) route through the maze to any location of water.

Phase 3 ('Efficiency' phase): CEL-0 refines its already-effective routes through the maze to reward locations. (This 'phase' is actually going on in parallel with the other two).

In each phase, CEL-0's behavior can be described in terms of three lists: a sequence of CEL operators, the corresponding sequence of overt moves in the simulated environment (if any), and the corresponding additions or changes to long-term memory (if any).

3.2 CEL-0's exploratory behavior in MazeWorld

For the purposes of this example, CEL-0 will start at location M0 in the maze, facing towards M3. The internal representation is described in *[Granger and McNulty 1984]*; it simply consists of information about what views are in front of, to the right of, behind and to the left of the current position of CEL-0 in the maze. Hence, the starting position has a view of M3 in front, walls to the right and behind, and M1 to the left.

(This is an admittedly huge oversimplification of a 'realistic' maze situation, but it seems justified for two important reasons: (1) selective attention to relevant features is the key thing that gets slighted by this oversimplification, and we have already done some analyses of selective attention in complex environments (see *[Granger 1988b]*); and (2) there are a number of interesting and complex processing problems that arise even *with* this simplification, and these problems would be difficult to present without first simplifying away the selective attention problems for pedagogical reasons.)

Because of the extremely simplified inputs for this example, DETECT and SELECT essentially just attend to everything here; see *[Granger 1988b]* and the selective attention section in this paper for an explanation of how these operators become much more complex in the face of more complex inputs.

Once SELECT has entered a representation into temporary memory, NOTICE attempts to match it against desirable and undesirable states, and REMIND attempts to match it against any existing schemas that might be relevant to the situation. There are three built-in 'exploratory schemas' in CEL-0, two of which get REMINDed by this input. Each of the three schemas (ES1, ES2 and ES3) is simply two events long, each corresponding to the 'impetus' to move in a particular type of situation, essentially corresponding to the following sequences:

ES1: see front opening ⇒ go straight

ES2: see obstruction ⇒ look around (360°)

ES3: see side opening ⇒ turn towards opening

So at location M0, REMIND will find both ES1 and ES3a, then ACTIVATE will have to choose at most one of them to pursue; for this example let it choose ES1. (ACTIVATE in fact contains a set of (currently six) 'preference metrics' that it uses to decide among proposed (REMINDed) alternative schemas – *[Granger and McNulty 1984]* describe these in detail). ENACT and SYNTHESIZE then begin to reconstruct ES1. ENACT does so by performing any events in the schema, and SYNTHESIZE by comparing new inputs that result from successive ENACTed events against the 'predicted' inputs in the schema itself. SYNTHESIZE notes that the match between the event and the more generalized representation in ES1 is only a partial match, and because it's not an exact match, calls BRANCH to create a new branch of the schema, and begins recording this new branch.

CEL-0 continues in this fashion, making the following moves through the maze: M0 - M3 - M5 - M3 - M0 - M1 - M2 - M1 - M0 - M3 - M5 - M3 - M4 - M3 - M0 - M1 - M2 - etc. An extensive description and explanation of the operator sequences driving these moves can be found in *[Granger and McNulty 1984]*.

Note that when M4 is arrived at, the fact that there is water there will cause a REMIND of another innate (built-in) schema that essentially says when water is seen, drink it. However, this schema may not be reconstructively ENACTed unless ACTIVATE lets it be (or unless there are no alternative schemas that get REMINDed); one of ACTIVATE's preference metrics says not to prefer schemas that do not match any currently desirable state, as specified on the 'Desirable State List' (DSL).

3.3 Effective goal pursuit in CEL-0

The result of the 'exploration' phase is the creation of a number of schemas describing various 'routes' through the maze, indexed by their starting and ending positions (more detail on INDEX is provided in *[Granger 1988b]* and *[Granger and McNulty 1984]*).

In phase 2, we simply add a desired event to **CEL-0**'s Desirable State List (DSL) – this is the list that NOTICE matches incoming events against, and that ACTIVATE checks to see whether or not to bother to ENACT a REMINDed schema. Hence, if we add 'drink water' to the DSL, **CEL-0** will now 'act thirsty', in the following three senses: (1) it will drink water if it sees any (via REMIND, ACTIVATE and ENACT of the built-in schema that says when water is seen, drink it); (2) it will tend to prefer sequences that lead to seeing water (via ACTIVATEs preference metric for currently-desirable states); and (3) it will store any sequences of events that lead to water (via NOTICE and COLLECT). Hence, via all three of these mechanisms, **CEL-0**'s memory will now contain schemas that are 'effective' with respect to the achievement of its goal of finding and drinking water.

3.4 Active exploration by CEL-0: Sensitivity analysis

A schema that leads to M4 at the end of the exploration phase is: M0 - M3 - M5 - M3 - M4. Note that while *effective*, this schema is not maximally *efficient* – it could simply go M0 - M3 - M4. The fact that it doesn't is simply an accident of the exploration phase (see Section 4.2 below). **CEL-0** has a process that causes schemas to be tested for their sensitivity to changes in the sequence; the process makes multiple variations of schemas by deleting various features or events from the event sequence, and then tests the resulting variations for their effectiveness.

The process effectively establishes a set of multiple internal 'hypotheses' as to which of the features of the episode are the most critical and predictive. Hence, this process amounts to a test of the sensitivity of the new episode to changes in those features. This process of testing episodes for their sensitivity to changes is termed 'sensitivity analysis'. The following subsections briefly outline the process.

3.4.1 Introduction to sensitivity analysis

When the model senses an instance of an episode, say, a pursuit-type episode such as M0 - M3 - M5 - M3 - M4, which results in some NOTICEd desirable state, that episode is COLLECTed into a long-term memory trace. The INDEX operator then begins to choose features of the events in the episode to use as indices, which will be used as recognition cues at retrieval time, i.e., whenever similar events happen subsequently. Depending on which features are chosen as indices, of course, subsequent retrieval either will or will not take place based on the presence or absence of any particular feature in the new input trace. Hence, the effective recognition of any new instance of a learned episode is sensitive to the feature-indices that are created at INDEX time during recording of the episode.

During the establishment of these feature indices on a new trace, the INDEX operator performs a multi-step process which has the effect of creating multiple traces of the episode, each with a different feature or set of features deleted from the trace. The multiple versions of the episode that result from this process serve the purpose of enabling **CEL-0** to test instances of the episode for their sensitivity to changes in the constituents of the episode: the long-term trace undergoes ongoing modification and refinement depending on which versions of the episode turn out to accurately match subsequent instances of the input stimuli.

Intuitively, what is happening is that the INDEX operator is hypothesizing a series of variations of the instance of the episode, implicitly predicting that these versions might serve as useful predictors of subsequent instances of the episode. Those predictive hypotheses are tested each time the set of variations of episodes are retrieved and reconstructively compared to a new input instance (via ENACT and SYNTHESIZE). In this way, the sensitivity analysis process allows the model to learn more than was contained in the single instance of the episode: it learns the ways in which that episodic instance might be sensitive to changes and variations. Furthermore, the process has the effect of robustly reducing any dependance on the order of presentation of events, making the model eventually learn the same things about the maze regardless of what order it happens to acquire them in.

3.4.2 The five steps of sensitivity analysis

1. When a new long-term trace is written, INDEX's first step is to search for any existing feature indices that match any features in the new trace. If so, then those indices are 'attached' to the trace, i.e., each index now points to the new trace in addition to any other traces it may already be pointing to.
2. For each sensory feature in the input, create a new index for that feature, that points to the episode.
3. For each feature-index pointing to the episode, either found by step 1 or created by step 2, begin creating variations of the episode by leaving out one or more of the features contained in the initial copy. Each variation is written into memory as a 'near-miss' copy of the episode.
4. For each of the new episode-variations, search for an existing index that has the new subset-features of the variation; if found, attach it to the episode.
5. For each feature-set index created, attempt to find others with subsets of the same features. For each such index set found, create a new higher-level index (see *[Granger 1988b]*), corresponding to the shared features that points to each of the members of the index set.

The combined effect of these steps will be to create a growing set of indices pointing to the episode, each of which will be triggered by a different set of feature cues at retrieval time. At the same time, multiple copies of the episode itself are being created, each a slight variation of the others; i.e., no two are exactly alike. The indices will slowly become a hierarchical set, because step 5 creates higher-level or 'second-order' indices, each of which points only to other indices (see *[Granger 1988b]*). For instance, 'template indices' are examples of higher indexes that contain only event-sequence information, with specific sensory information deleted.

4. Some insights resulting from experience with **CEL-0**

There are a number of difficulties that have arisen during the programming of **CEL-0** that have the form of interesting theoretical problems that were not obvious until the implementation difficulties arose. Some of these are discussed here, with the focus on the emergence of seven categories of learning, based on seven different 'calling sequences' of **CEL** operators all of which are capable of establishing or modifying a memory trace, i.e., learning.

4.1 Seven ways to establish a memory trace in CEL

The twelve **CEL** operators do not call each other serially; hence, although COLLECT is the primary way for episodic traces to be established in permanent memory, there are four distinct calling sequences that may result in the creation of a new trace, each of which constitutes a category of learning in **CEL**; in turn, these four categories have between them a number of different subcategories, for a total of seven. These are listed here, followed by a set of brief descriptions and examples of each subcategory.

Goal-based establishment:

1. *pursuit* of desirable result *(Pursuit-based learning)*
2. *avoidance* of undesirable result *(Avoidance-based learning)*

Expectation-based establishment:

1. match between expectation and environment *(Success-driven learning)*
2. mismatch between expectation and environment *(Failure-driven learning)*

Exploration-based establishment:

1. analysis of relevance of schema features *(Sensitivity analysis)*

Coincidence-based establishment:

1. schema activated simultaneously with newly-created schema *Append-driven learning)*
2. two schemas concurrently activated *Splice-driven learning)*

4.1.1 Goal-based trace establishment

When the NOTICE operator finds that an incoming event matches something on either the Desirable or Undesirable state list (*DSL* or *USL*) (see *[Granger 1983b]*), NOTICE triggers the COLLECT and INDEX operator to make a record of the sequence of events that led up to the desirable or undesirable event.

Case one: Pursuit-based learning

In the *desirable* case, the INDEX operator simply indexes the sequence of events by SELECTed features (see *[Granger 1983b]*).

Case two: Avoidance-based learning

In the *undesirable* case, INDEX calls the DETOUR operator to attempt to create a link pointing to potential *alternatives* to the undesirable result, so that that path won't be pursued in the future.

4.1.2 Expectation-based trace establishment

While a schema is being reconstructively ENACTed after having been triggered (REMINDed and ACTIVATEd) by some cue, the SYNTHESIZE operator is constantly matching incoming real-world events against events in the schema (i.e., it is checking the schema's implicit expectations). Both matches and mismatches can cause new things to be written into memory.

Case three: success-driven learning

If SYNTHESIZE finds a match, then it calls REINFORCE to add 'strength' to the links pointing to the successfully predictive schema.

Case four: failure-driven learning

If a *mis*match is found, BRANCH is called to create a new link between the index and the new sequence of events (whatever just actually happened), thereby effectively reducing the relative strength of the link from the index to the previously-expected result.

4.1.3 Exploration-based trace establishment

Apparent exploratory behavior by CEL arises from the operation of the 'sensitivity analysis' procedure described above (and described in more depth in *[Granger 1988b]*), combined with the existence of the set of simple 'exploratory schemata'. Recall that sensitivity analysis causes a number of variations of each schema to be created, each of which will be tested and either strengthened or weakened according to its success or failure. These will operate on the schemas collected during CEL-0's 'wandering' through the maze, to refine the model's representation of pathways through the maze, eliminate some redundancies, and identify some 'landmarks' that make useful indices to the set of pathways (see *[Granger and McNulty 1984]*).

As it collects sequences of paths through pieces of the maze, sensitivity analysis refines them by testing the relevance of their consituent events.

Case five: Sensitivity analysis

For instance, if an initial route through the maze is the sequence M0 - M3 - M5 - M3 - M4, a diminution of the route yields M3 - M5 - M3 - M4, which will work when the starting point is M3. Further diminution cuases the eventual creation of the route M3 - M4, which is actually an improvement over the original in terms of efficiency, since it can get to the presumably desirable state M4 without bothering to go through M5 and doubling back through M3. Note that in light of this new schema, the initial five-step route can be viewed as 'superstitious' behavior; i.e., the model is acting as though it 'thinks' that just because it went through M5 to get to M4 the first time, it must do so on subsequent trials. It is crucial to note that efficiency is not always best; in fact, mammals can be trained to repeat long sequences of otherwise 'superstitious' behavior, as long as that behavior is rewarded, while any variations go unrewarded (see e.g., *[Hilgard and Bower 1970]*).

4.1.4 Coincidence-based trace establishment

There are two cases of 'coincidence' that can arise in the model: either an existing schema gets REMINDed during the COLLECTion of a new schema, or a schema gets REMINDed during the ENACTing of another schema that has been previously REMINDed and ACTIVATEd.

Case six: Append-driven learning

If the model is COLLECTing a new schema that leads to, say, an undesirable result, such as an unpleasant taste, that NOTICEd taste may simultaneously cause a REMIND of, say an innate 'gag reflex' schema (i.e., it says to spit out after sensing a bad taste). In such a case, the INDEX (and DETOUR) operators create index links to both the sequence of events leading up to the bad taste, so that it might be avoided in the future, and to the sequence of events REMINDed by the event, so that this sequence might be substituted for the undesirable sequence the next time it happens; this is an instance of an 'active avoidance' situation.

Case seven: Splice-driven learning

If the model is currently ENACTing an active schema, e.g., running a maze toward a food reward, and during this, another schema gets REMINDed (e.g., a light flash that is known to lead to some different reward), then both schemas are indexed together by the same initiating feature, giving that feature added predictive power.

4.2 Note: Design decisions affecting CEL-0's performance

It should be noted here that a number of design decisions in CEL-0 (including the specifics of the ACTIVATE preference metrics, the details of the built-in exploratory schemas, and the details of the functions of the operators, notably SYNTHESIZE, REMIND and ACTIVATE) will affect the path it will take through the maze, and in many cases will affect whether or not the correct learning will take place at all. We have been experimenting with versions of CEL-0 to see which changes cause which behaviors, but we intend to continue to compare the resulting behaviors against the learning literature wherever possible (see esp *[Rescorla and Wagner 1972]*), and to suggest new experiments (and their predicted outcomes) when the literature doesn't provide the necessary data on some specific point about how a rat, for instance, should run the maze. Section 5 of this paper makes some brief remarks about our use of some results in animal learning as a 'requirements specification' for CEL-0's performance; *[Granger and McNulty 1984]* contains more discussion of this.

5. The neurobiology of multiple memory systems

5.1 The constellation of deficits in the amnesic syndrome

The patient H.M., like most other amnesics, exhibits a whole constellation of related deficits. The key deficit is the inability to consciously store new information, as described earlier in this paper. Two of the other major components of the overall amnesic syndrome are:

- *Retrograde amnesia*: H.M not only is incapable of consciously storing new information *since* his operation; he also has lost some of the memories that happened to him immediately *preceding* the operation, up to about two years before the operation, while memories older than that remain unimpaired. This striking finding *[Squire 1980]* is used as evidence that memory *consolidation* takes time (perhaps up to two years) before it becomes a permanent part of memory; hence, perhaps memories that were still being consolidated at the time of the operation were disrupted, and never got firmly established as permanent memories.
- *Rapid Forgetting*: H.M. is able to carry on conversations, and perform other tasks of long duration, *as long as the task isn't interrupted*; when interrupted for more than a few minutes, he completely forgets where he was, and starts over again 'from scratch', e.g., he might then have the exact same conversation all over again without realizing he's just done it.

5.2 Of rats and men

There are recently-discovered situations in which rats in a maze exhibit forms of learning previously only attributed to primates and humans. *'Learning-set learning'* refers to very rapid (usually just a single trial) learning of new situations that are similar to previously-learned ones, i.e., the animal seems to form a *'template'* that it can use to expidite the learning of subsequent situations. The rats' learning-set learning (*LSL*) system apparently is entirely separable from its more standard, slower 'associative learning' (*AL*) system – there are specific drugs and lesions that have been used to entirely eliminate abilities associated with the *LSL* system without affecting the performance of the *AL* system, and vice versa. This constitutes evidence that rats have multiple memory systems.

Furthermore, recent experimentation *[Staubli and Lynch 1989]* has shown that rats can be given amnesic symptoms strikingly similar to those in humans, by making corresponding lesions to the hippocampus and another limbic structure, the thalamus. In particular, rats are trained to select one of two odors for a water reward. This initially requires 50-100 trials before a minimal criterion of learning is met (i.e., associative learning (AL)). Over successive pairs of odors, the rats' behavior changes such that they come to learn the correct odor in subsequent odor-pairs in only 3-4 trials (learning-set learning (LSL)). Two forms of learning are thought to be involved: (1) abstract 'template-driven' (LSL-type) information about the task (e.g., the fact that it contains a 'correct' and an 'incorrect' olfactory cue), and (2) specific memory (AL-type) as to which particular odor was correct for a given pair.

One specific type of lesion (lesions of the connection between the dorsomedial nucleus (DMN) of the thalamus and the frontal cortical system) eliminates the animals' ability to go from the many-trial (AL) mode to the subsequent rapid-learning (LSL) mode over successive pairs of odors. This suggests that the rats are learning the *specific* memories for correct odors, but are failing to learn the *template* information about the existence of correct and incorrect odors in each pair.

Disconnection or lesions of the hippocampus, on the other hand, produces an apparent inverse of this result, with a time-dependency as well: the rats acquire the rapid learning mode (i.e., they appear to learn the abstract correct-incorrect information), but for any given pair of odors they cannot recall the right specific odor (i.e., cannot perform the task) if delays of more than about 5 minutes are interposed between trials (i.e., a deficit similar to rapid forgetting). Hence it seems that these rats are acquiring the abstract memory, but are failing to create a long-term trace of the specific memory.

5.3 Interpretation of the data

What ability, i.e., what specific knowledge or process, is available to the rat the LSL situation, but not in the AL situation, to enable template-driven learning? The problem for **CEL** (or for any other model of learning and memory) in attempting to provide a consistent account of these results, is that apparently the templates are learned but the specific memories leading to those templates are lost. We do not have a complete solution, but we have come up with a set of opposing hypotheses, either of which could potentially explain the data. These opposing hypotheses have been used to

design an experiment that is currently being run to help further clarify the the question, and to narrow down the set of possible consistent models of these two learning systems.

In the language of the CEL model, there are two classes of possible explanations: (1) the hippocampal (AL) losses are due to a 'storage-side' failure to either COLLECT or INDEX the specific information, or else (2) these losses are due to a 'retrieval-side' failure to correctly *use* the specific odor memory to find the water reward; i.e., perhaps REMIND finds both the template memory and the specific-odor memory, but ACTIVATE is not correctly using the specific-odor memory to instantiate the template memory in order to find the reward.

The articulation of these two opposing possibilities has suggested an experiment to try to test whether the specific odor was in fact present in memory at all. The memory seems not to show up in the odor-choice situation, but if explanation (2) above is correct, then the memory may be there but just not being used correctly in that situation. It turns out that there is a relatively simple experimental methodology for testing for 'raw memories' like this. Details are provided in *[Granger 1989b]*, but briefly, the experiment allows us to see whether a rat has any memory of a particular event (such as a specific odor) or no memory of that event. That is, the rat's behavior in the presence of some previously-seen event can be reliably distinguished from its behavior in the presence of an unrecognized event; hence, we should be able to tell whether or not the specific odor is in memory or not. This experiment, described in *[Granger 1989b]*, is currently being run at the Center for the Neurobiology of Learning and Memory at Irvine.

If it turns out that the memory shows up in this experiment, then we may hypothesize that the deficit is on the retrieval side, that is, the memory is present, but it cannot be correctly used to perform the choice behavior. In CEL terms, it is possible that ACTIVATE cannot instantiate the memory into the template that can use it to find the water reward. If, on the other hand, the rats exhibit no recognition of the specific odors, we will hypothesize that the deficit may indeed be a storage-side deficit, and we will have to attempt to alter the model to account for the loss of a spcific memory after the creation of a template from it.

Either way, the CEL model will have aided in suggesting a key experiment that can deicde the question of whether the rapid-forgetting phenomenon is a storage-side or a retrieval-side deficit. This brings us a step closer to an understanding of the nature of the multiple (LSL and AL) learning and memory systems.

6. Conclusions: Artificial and natural learning mechanisms

There exist many theoretical questions in learning and memory that rely on the consistent interpretation of an almost bewildering array of interrelated experimental results. The field of multiple memory systems is one particularly exciting current example of this; a battle over the characteristics of these systems is currently raging among memory researchers in the neurosciences *[Squire 1980; Squire, Cohen and Nadel 1982; Mishkin 1982; Tulving 1984]*.

The search for consistent interpretations of these data can be aided by artificial models of learning and memory, and, reciprocally, the development of consistent models can be furthered by the experimental testing of the models' predictions against natural learning systems. While it is not necessary for an artificial learning system to precisely account for all available psychological data on learning, it has happened time and again in AI that sincere attempts to provide consistent interpretations of problematic psychological results have resulted in both better psychological theory and richer and more productive computer systems.

There are certain specific processing problems that any learning system, natural or artificial, must have a way of solving. We are trying to characterize some of those processing problems in specific learning situations, in hopes of identifying the similarities among, and differences between, different instances of learning systems. The CEL model has so far been helpful in identifying and clarifying some of the possible theoretical interpretations of results in the area of multiple memory systems. We hope that by continuing to iterate the loop from theoretical suggestion to experimental result and back, we can further refine and narrow down the range of possible interpretations of multiple learning and memory systems, so that the study of artificial and natural learning mechanisms can productively use each others' results.

References

Carbonell, J., "Derivational Analogy and its Role in Problem Solving," in *Proceedings of the 1983 National Conference on Artificial Intelligence*, 64-69 , 1983.

Granger, R.H., "Identification of Mental Operators Underlying the Acquisition of Simple Predictive Behavior.," *Department of Computer Science Technical Report 191*, University of California, Irvine, November, 1982.

Granger, R.H., "Identification of Components of Episodic Learning: The CEL Process Model of Early Learning and Memory.," *Journal of Cognition and Brain Theory*, 6, 1 (February 1983a), 5-38.

Granger, R.H., "An Artificial-Intelligence Model of Learning and Memory that Provides a Theoretical Framework for the Interpretation of Experimental Data in Psychology and Neurobiology," *Department of Computer Science Technical Report 210*, University of California, Irvine, September, 1983b.

Granger, R.H., and McNulty, D.M., "The CEL-0 system: Experience with a computer model that learns to run a maze," *Department of Computer Science Technical Report 220*, University of California, Irvine, March, 1984.

Hilgard, E.R. and Bower, G.H., *Theories of Learning*, New York: Appleton, 1966.

Kolodner, J.K., "Maintaining Organization in a Dynamic Long-Term Memory," *Journal of* Cognitive Science, 7, 4 (1983a), 243-280.

Kolodner, J.K., "Reconstructive Memory: A Computer Model," *Journal of Cognitive Science*, 7, 4 (1983b), 281-328.

Langley, P., Bradshaw, G.,and Simon, H.A., "BACON.5: The discovery of conservation laws.," in Proceedings of the Third International Joint Conference on Artificial Intelligence, 121-126 , 1981.

Lebowitz, M., "Generalization and memory in an integrated understanding system," *Department of Computer Science Technical Report 186*, Yale University, New Haven, Conn., 1980.

Mishkin, M., "Memory in monkeys severely impaired by combined but not by separate removal of amygdala and hippocampus," *Nature*, 273, (1978), 297-298.

O'Keefe, J., and Nadel, L., *The Hippocampus as a Cognitive Map*, Oxford: The Clarendon Press, 1978.

Schank, R.C., "Failure-Driven Memory," *Journal of Cognition and Brain Theory*, 4, (February 1981), 41-60.

Simon, H.A., Langley, P. and Bradshaw, G., "Scientific discovery as problem solving," *Synthese*, 47, (1981), 1-27.

Slotnick, B., "Olfactory learning in rats," *Science*, 185, (1974), 796-798.

Squire, L., Cohen, N., and Nadel, L., "The medial temporal region and memory consolidation: A new hypothesis," in *Memory Consolidation*, Hillsdale, NJ: Erlbaum Assoc., 1982.

Squire, L.R., "Two forms of human amnesia: an analysis of forgetting," *Journal of Neuroscience*, 1, (1981), 635-640.

Tolman, E.C., *Purposive Behavior in Animals and Men*, New York: Century, 1932.

Tulving, E., "Episodic and semantic memory.," in *Organization of Memory*, E. Tulving & W. Donaldson (Eds.), New York: Academic Press, 1972.

Interaction Effects between Word-Level and Text-Level Inferences: On-line Processing of Ambiguous Words in Context

Richard H. Granger
Jennifer K. Holbrook
Kurt P. Eiselt

Artificial Intelligence Project, Computer Science Department
and
Cognitive Sciences Program, Social Sciences Department
University of California
Irvine, California 92717

1. Introduction

Ambiguous interpretations that arise during text understanding are triggered by meanings of words in context. Our recent research into on-line processes of text understanding has examined how readers choose between two equally plausible interpretations of a complete text [Granger & Holbrook, 1983; Granger, Eiselt, & Holbrook, 1983]. Other researchers have focused on how readers resolve ambiguity of individual words in context [e.g., Swinney, 1979; Tanenhaus, Leiman, & Seidenberg, 1979; Lucas, 1983]. The problem addressed in this paper focuses on the overlap between these two lines of research: in particular, how the on-line process of selecting from among ambiguous word-senses contributes to, and is itself affected by, the process of selecting from among alternative, equally plausible interpretations of the overall text.

Although initial context in a text may suggest a word-sense for an ambiguous word in context, the word's ambiguity often persists in reading. This can be illustrated by using a text in which the initial context is misleading. For example, compare the following three texts:

[1] The CIA called in an inspector to check for bugs. Some of the secretaries had reported seeing roaches.

[2] The management called in an inspector to check for bugs. Some of the secretaries had reported seeing roaches.

[3] The management called in an inspector to check for bugs. They knew their rivals would stop at nothing to get their trade secrets.

The word "bugs" is ambiguous in all three texts until the second sentence, yet the first sentence of each text suggests a particular reading. In the first text, the "microphone" meaning of bug initially appears to be more appropriate than the "insect" meaning. Yet the second sentence of text [1] makes it clear that the "insect" meaning is correct. In text [2], both sentences suggest the "insect" reading. In text [3], the first sentence suggests the "insect" reading, while the second sentence forces "insect" to be dropped in favor of "microphone".

This research was supported in part by the National Science Foundation under grant IST-81-20685 and by the Naval Ocean Systems Center under contracts N00123-81-C-1078 and N66001-83-C-0255.

Thus, even if a meaning of a word seems inappropriate initially, there must be some way to retrieve that meaning as more context becomes available. We call the process by which initially inappropriate meanings are reactivated, **cued reactivation**. There are two logically competing hypotheses as to what happens to the unselected word-senses once context has ruled them out. The first, proposed by Tanenhaus, et al. [1979], is known as **active suppression**. With active suppression, a meaning in accord with initial context is selected. The other meanings are no longer primed, that is, no longer available for quick recall. In fact, the priming effect for initially inappropriate meanings is lost much more quickly when the word is in a biasing context than when the word is not in a biasing context. Thus, a word and all its meanings would have to be re-examined if initial context was misleading. Consider again texts [1] and [2]:

[1] The CIA called in an inspector to check for bugs. Some of the secretaries had reported seeing roaches.

[2] The management called in an inspector to check for bugs. Some of the secretaries had reported seeing roaches.

If active suppression is correct, then with text [1], the context of "CIA" will choose "microphone" as the correct word-sense of "bugs". Once "microphone" is chosen, other meanings, including "insect", will be suppressed. After the word "roaches" is read, the reader will not be able to reconcile "microphone" with "roaches", so all meanings will have to be recalled, and a meaning chosen on the basis of the further context. If instead, the reader saw text [2], the "insect" meaning of "bugs" would be selected initially. Therefore, when the word "roaches" is read, there would be no conflict in meaning, and the several word-senses of bugs would not have to be re-activated. It should be possible to measure the difference in the processes needed to understand these two texts as a difference in reading times.

We propose a second theory, which we call **conditional retention**. With conditional retention, the initially inappropriate meanings of a word are not actively suppressed if further text is available. In other words, all meanings retain their priming when it is possible that further context could cause a re-interpretation of earlier context and therefore earlier word-sense choices. Thus, for example, when text [1] is read, the "microphone" meaning of "bugs" would be selected as appropriate to context. If the text ended after the first sentence, other meanings would be suppressed. However, if the text continued after the first sentence, the other meanings, including the "insect" meaning, would still be primed. Thus, when the word "roaches" is read, the connection between "roaches" and "insect" would be available, and there would be no need to re-process "bugs". Processing text [2] would be no different: the "insect" meaning would be initially selected, and the match between "insect" and "roaches" discovered in the same way. Therefore, there should be no significant difference in reading times if conditional retention is correct.

This paper describes two experiments we ran which were designed to decide between the active suppression and the conditional retention theories. The results from these experiments are presented as evidence for a new theory of on-line word-sense disambiguation during text processing. This new theory is one part of a new model of how all inference decisions are made during text comprehension. The theory we present here incorporates experimental results of other researchers [Swinney, 1979; Tanenhaus, et al., 1979; Lucas, 1983] with our own to create a more comprehensive theory of the interaction between word-level and text-level inference phenomena than has previously been possible.

2. Background

In building an earlier model of inference behavior [Granger, Eiselt, & Holbrook, 1983], we, as did other researchers, made implicit assumptions by pre-parsing the input to our model. We essentially started in the middle of the whole inference process. We have found that many of the inferences which could be made about a single statement were often triggered by a single word, and since most words are ambiguous to some degree, our model now had to include a method of resolving lexical ambiguity in a way consistent with our results from experiments on human subjects. As a result, we have incorporated some ideas about resolving lexical ambiguity which are suggested by several recent experiments in lexical access, the low-level processes that retrieve and use words and word-senses during reading [Swinney, 1979; Warren, 1977; Tanenhaus, et al., 1979; Lucas, 1983].

2.1. The Lexical Access Findings

Lexical access involves the translation of a word's phonological or orthographic code into its underlying meaning. Of course, many words are ambiguous to some degree, so some process is needed to disambiguate these words, selecting the most appropriate meaning for the context. Two such processes are possible: either context suggests the correct meaning before an ambiguous word is processed, so that the inappropriate meaning is never accessed, or all meanings of an ambiguous word are accessed initially, and context is subsequently consulted to determine the most appropriate meaning.

To decide between these two theories, experiments would utilize the effects of word-sense priming. Essentially, what has been found with priming is that when an ambigous word is presented to a subject, words which are related in meaning to any of the ambiguous word's senses are more quickly recognized than words which are unrelated to the ambiguous word's word-senses. However, when there is no context present, one meaning of the ambiguous word is chosen as a default, and the other meanings which are not chosen lose their priming; that is, they are no longer recognized more quickly than words which are not related to any of the meanings. If the first theory is correct, so that the context narrows the search down to only the meanings which are in accord with it, and the inappropriate meanings are thus never accessed, then the inappropriate meanings would not be primed when the word is read. However, if context is consulted only after all meanings have been accessed, then all meanings should be primed when the word is read. Once a meaning is chosen, the other meanings would presumably lose their priming in the same way that non-default meanings of words presented with no context lose their priming.

Swinney [1979] and Tanenhaus, et al. [1979] found that all meanings of a word are initially accessed, and context is then consulted to determine which word-sense is most appropriate. Lucas [1983] extended their results in two ways. First, she showed that before an ambiguous word is seen, context itself primes the appropriate word-sense. Second, after the word is seen, all word senses are accessed regardless of context. At 100 msec. after the ambiguous word is shown, the default meaning is "more" primed (slightly more quickly recognized) than non-default meanings, whether the default is appropriate to context or not. Within the next 100 msec., though, the context selects the most appropriate meaning; thus, context becomes fully active again as the disambiguation process continues. At 200 msec. after the word is seen, only the context-appropriate meaning of the word is still primed.

When an ambiguous word is presented with context, the inappropriate meanings remain active

for between 100 and 200 msec., as noted above. However, Warren [1977] indicates that when an ambiguous word is presented without context, all meanings are available for much longer than 200 msec. Therefore, Tanenhaus, et al. [1979] have suggested that when context is available for disambiguation, inappropriate meanings of a word are actively suppressed; in other words, disambiguation involves not only the identification of the correct meaning, but the immmediate erasure of primed but inappropriate meanings. Tanenhaus, et al. did not test this theory experimentally.

3. The Dilemma

As was shown with texts [1-3] above, initial context may suggest a misleading interpretation which is corrected by the later text. While active suppression explains the difference between the priming of meanings when a word is not in context versus when the word is in the context of a sentence, it may not explain what happens when there are two conflicting contexts, as in the texts above. It may be that when a word is presented with more context than simply the sentence in which it appears, active suppression does not occur. Readers may instead keep both meanings of the word primed, while waiting for confirming context. In contrast to the active suppression hypothesis, we will call this the conditional retention hypothesis; with this hypothesis, although initial context selects a meaning, the other meanings will be retained on the condition that there is further text which might suggest a second meaning, and suppressed on the condition that no more context is available.

Conditional retention is a possible solution to the questions raised when a theory based on active suppression is adopted: if the meaning of the word which is selected initially turns out to be incorrect, as in texts [1] and [3] above, how will the other meaning be recovered? Must both meanings be re-primed, and the second meaning be selected instead? Or will the second meaning not be recoverable, and the text be difficult to understand? Yet, if we discard active suppression altogether, how can we explain why inappropriate meanings are no longer primed when there is context, and are equally primed with no context? With the conditional retention hypothesis, these problems are solved.

4. The Experiments

Two experiments were conducted to test between the active suppression and the conditional retention hypotheses of cued reactivation. The first experiment had subjects read texts which consisted of sentence pairs. The first sentence had an ambiguous word in it, with the text preceding the word either biased toward one of the meanings of the word or biased toward neither meaning. The second sentence of the texts either did not bias toward either word-sense, continued biasing toward the same meaning as the first sentence, or biased toward the other meaning. Thus, the subjects saw four types of text with two sentences in each: (1) no biasing context, (2) bias follows the ambiguous word, (3) single bias surrounds the ambiguous word, and (4) two different biases on either side of the ambiguous word. Table 1 contains examples of each type of text.

After the subject read a text, he or she saw a string of letters and had to decide, as quickly as possible, if the string was a word. One of four strings of letters was presented after each text. One of the strings was a nonsense word. One of the strings was a word unrelated to either meaning of

No biasing context with ambiguous word (NONE):	They had someone check for bugs. It was a routine precaution.
Bias following ambiguous word (FOLLOWS):	They had someone check for bugs. The secretaries had reported seeing roaches.
Bias surrounding ambiguous word (SURROUNDS):	The CIA called in an inspector to check for bugs. The secretaries had reported seeing microphones.
Double bias around ambiguous word (DOUBLE):	The CIA called in an inspector to check for bugs. The secretaries had reported seeing roaches.

Table 1: Examples of the four types of text presented to subjects.

the word. Two other strings were words each related to one of the meanings of the ambiguous word. Table 2 contains examples of each type of letter string.

Nonsense string:	RUD
Unrelated word:	PEN
Related to meaning 1:	SPY
Related to meaning 2:	ANT

Table 2: Examples of the four string types presented after text containing the word "bugs".

If the initially inappropriate meaning is actively suppressed, then that meaning is no longer primed, and thus will have to be re-primed when further context reverses the appropriateness judgment made earlier in the text. If meanings are actively suppressed, ambiguity is essentially no longer present; therefore, the initially inappropriate meaning should need to be re-activated. Thus, a doubly-biased story should take more time to understand than an unbiased, a bias-following, or a bias-surrounding story because the latter three types do not initially bias inappropriately, and therefore do not force such a reinterpretation, whereas a doubly-biased story initially biases the understander toward one meaning which turns out to be inappropriate, forcing a reinterpretation of the ambiguous word. With active suppression, if both meanings are suggested in the text, we would expect a slower judgment of whether the string following the text was a real word than if only one meaning is suggested by the texts.

With the conditional retention hypothesis, because all meanings of a word are retained as long as further text is available, cued reactivation becomes a simple matter of selecting one meaning over the other using all available context, rather than initial context. Because there are still links left to the initially inappropriate meaning, there should be little difference in the time it takes to understand a story with both meanings of a word presented and a text with only one of the meanings presented.

The second experiment had subjects read the same texts as in the first experiment, but the subjects' task was slightly different: they had to decide as quickly as possible which of two words was most closely related to the meaning of the text. Each of the two words was related to one of the meanings of the ambiguous word in the text (see Table 2). If the active suppression theory is correct,

initially inappropriate meanings would have to be re-accessed, and initially appropriate meanings would then be suppressed. Thus, few mistakes would be made about which word was most closely related to the text. With the conditional retention hypothesis, both meanings would be retained, so that errors on the decision task should be fairly frequent.

5. Results and Conclusions

The data for the first experiment are not all collected. The results of the second experiment strongly suggest that a conditional retention hypothesis should be adopted.

sentence-pair types:	NONE	FOLLOWS	SURROUNDS	DOUBLE
percentage of errors:	0%	7%	7%	54%

Table 3: Percentage of errors made in word choice task.

When subjects were given unbiased texts which did not have an ambiguous word primed by text, they made no errors in judgment about which word was most associated with the text. When subjects were given bias-following texts with the first sentence unbiased and the second sentence biased toward one meaning of an ambiguous word, only 7% of the decisions were errors. When subjects were given bias-surrounding texts with both sentences biased toward the same meaning of the ambiguous word, only 7% of the decisions were errors. The difference between these conditions is not significant. However, when subjects were given doubly-biased texts with one sentence biased toward one meaning of an ambiguous word, and the second sentence biased toward a different meaning, 54% of the decisions were errors. The number of errors made in this condition is significant.

Although the results so far indicate that a conditional retention theory is correct, we are currently re-running our experiments. We are, among other minor changes, using a larger population to increase external validity.

These results have helped us to refine our model of inference decisions [Granger, Eiselt, & Holbrook, 1984]. We do not see these findings as applicable only to the lexical analysis level, because the ambiguous words were the basis for pragmatic inferences about the whole text. Therefore, we are incorporating our findings at both the lexical analysis level and the pragmatic inference level. We are using this model to predict the results of other experiments which we have designed.

6. Acknowledgements

The authors would like to thank Rogers Hall for his assistance. We also deeply appreciate the time and effort put into this project by Frank Kavanaugh.

7. References

Granger, R.H., & Holbrook, J.K. Perseverers, recencies, and deferrers: New experimental evidence for multiple inference strategies in understanding. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*, Rochester, New York, 1983.

Granger, R.H., Eiselt, K.P., & Holbrook, J.K. STRATEGIST: A program that models strategy-driven and content-driven inference behavior. *Proceedings of the National Conference on Artificial Intelligence*, Washington, D.C., 1983.

Granger, R.H., Eiselt, K.P., & Holbrook, J.K. The parallel organization of lexical, syntactic, and pragmatic inference processes. *Proceedings of the First Annual Workshop on Theoretical Issues in Conceptual Information Processing*, Atlanta, Georgia, 1984.

Lucas, M. Lexical access during sentence comprehension: Frequency and context effects. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*, Rochester, New York, 1983.

Swinney, D.A. Lexical access during sentence comprehension. (Re)-consideration of context effects. *Journal of Verbal Learning and Verbal Behavior*, *18*, 645-660, 1979.

Tanenhaus, M., Leiman, J., & Seidenberg, M. Evidence for multiple stages in processing of ambiguous words in syntactic contexts. *Journal of Verbal Learning and Verbal Behavior*, *18*, 427-440, 1979.

Warren, R.E. Time and the spread of activation in memory. *Journal of Experimental Psychology: Human Learning and Memory*, *3* (4), 458-466, 1977.

Jumping to Conclusions: Psychological Reality and Unreality in a Word Disambiguation Program

Graeme Hirst

Department of Computer Science
University of Toronto
Toronto, Canada M5S 1A4

1. Introduction

Human language understanding sometimes jumps to conclusions without having all the information it needs or even using all that it has. So, therefore, should any psychologically real language-understanding program. How this can be done in a discrete computational model is not obvious. In this paper, I look at three aspects of the problem:

- When is information ignored?
- When is a decision made out of impatience?
- When is no decision made at all?

I give illustrations of these problems in the domain of word disambiguation with the Polaroid Words system.

2. Polaroid Words

Polaroid[1] Words are a system for the disambiguation of words and case slots; they are a part of the Absity natural language understanding system (Hirst 1983a, 1983b). Their design is based in part on the results of recent psycholinguistic studies of word disambiguation that show that usually all meanings of an ambiguous word are activated and one is then chosen (Swinney 1979, Onifer and Swinney 1981, Seidenberg, Tanenhaus, Leiman and Bienkowski 1982). Thus in *The man walked on the deck*, both meanings of *deck*, 'pack of cards' and 'part of a boat', are activated below conscious awareness. This is in contrast to script-based models (Schank and Abelson 1977), in which the script acts as a context to predetermine a unique meaning for each ambiguous word, an approach clearly inadequate for polysemous words.

Each Polaroid Word (PW) is an independent procedure that is responsible for the disambiguation of one word or case slot in the input sentence. The PWs operate in parallel with one another and with other processes in the system.[2] There is one type of PW for nouns, another for prepositions, and so on. Each begins with a packet of knowledge that lists all possible meanings for its word or slot, and, as it obtains the knowledge to do so, eliminates all meanings that are inappropriate until just one is left. The possibilities in the PW's list are always well-formed semantic objects in the Frail frame system (Charniak, Gavin and Hendler 1983), and therefore may be used for retrieval and inference both by PWs and by other processes in the system, regardless of the extent to which disambiguation has or hasn't taken place. Polaroid Words and their many virtues are described more fully in Hirst (1983a) and Hirst and Charniak (1982); in the present paper we concentrate on their deficiencies.

3. Jumping to conclusions

3.1. Spreading activation and magic numbers

It has been shown that *semantic priming* by *spreading activation* (Collins and Loftus 1975) is important in human lexical disambiguation. Accessing a concept in memory temporarily

This work was carried out while I was at the Department of Computer Science, Brown University, Providence, Rhode Island. Financial support was provided in part by the U.S. Office of Naval Research under contract number N00014-79-C-0592. I am grateful to Eugene Charniak for discussions from which this work developed.

[1] *Polaroid* is a trademark of the Polaroid Corporation.

[2] In this respect, they bear a superficial similarity to Small's (1980) Word Experts.

activates both that concept and those closely connected to it, facilitating their subsequent retrieval. Seidenberg *et al* found that strong semantic priming of one sense of an ambiguous word was the only case where not all meanings were considered. For example, in *The bridge player trumped the spade*, the word *bridge* primes the 'playing card suit' meaning of *spade*, and 'digging instrument' is not considered.

To account for the effects of spreading activation in ambiguity resolution, Polaroid Words use *marker passing*, a discrete model of spreading activation in a network of frames and slots in the Frail representation. Marker passing can be thought of as spreading tokens along the arcs of the representation, marking each node reached, until all nodes within a certain distance of the origin have been marked. The trails of marks thus created are called *paths*. Markers may be passed along any connection in the network: from frame to slot, slot to filler, slot to constraint, class to sub-class, and so on. Markers are passed only to nodes within a few steps of the origin; otherwise, of course, the whole knowledge base would always get marked, a useless situation.

Before it does anything else, a PW checks whether one of its possibilities has, as result of previous activity, received a marker. If so, it decides immediately on this possibility without any further consideration. Otherwise, it asks Frail to start passing markers from each of its possibilities. If one of the paths so created intersects with a previously made path, this is taken as evidence that the origin is the appropriate sense of the ambiguous word. The closer the intersection is to the origin, the stronger the connection is considered to be; if the path is strong enough, the indicated sense is chosen. (If no such intersections are found, or only weak ones, the PW resorts to other methods, described in Hirst (1983a; Hirst and Charniak 1982).)[3] For example, in sentence (1):

(1) The plane taxied to the terminal.

the ambiguous words *plane* and *terminal* are resolved by finding the path between their aviation-related senses, but finding no path between any active concept and their other senses.

[3]Weak paths are not ignored entirely; rather, additional evidence is sought before a final decision is made.

The problem that immediately arises with this scheme is that of setting thresholds. How far from the origin should marker passing go? How strong does a path have to be before the PW can jump to a conclusion without considering other evidence? It is clear that there are psychologically real thresholds, for they sometimes result in people misinterpreting *negatively primed* ambiguities:[4]

(2) The astronomer married the star.

(3) The rabbi was hit on the temple.

(4) The sailor ate the submarine.

(5) The catcher filled the pitcher.

Although the selectional restrictions on *marry* in (2) are sufficient to uniquely determine the sense of *star* as 'celebrity', spreading activation from the meaning of *astronomer* causes most listeners to select the sense 'celestial object', despite the nonsensical result.[5] That is, the human disambiguation mechanism will sometimes wrongly jump to a conclusion — and PWs are likewise fooled by these sentences — even though information is present that would let it avoid the error. On the other hand, people generally have no trouble with the following sentences, which fall on the other side of the thresholds:[6]

(6) The lawyer bent the bar.
(*bar* ≠ 'legal profession')

(7) The dog chewed the bark.
(*bark* ≠ 'dog noise')

(8) The statistician sat on the table.
(*table* ≠ 'array of figures')

At present in Polaroid Words, markers are passed to nodes up to four steps away from the origin; but this threshold is just a magic number chosen arbitrarily, and is dependent upon the exact degree of coarseness of the Frail knowledge representation. What we need in order to determine a more realistic threshold is a large set of

[4]Sentences (3) and (5) are from Reder (1983).

[5]In the other sentences, *temple* = 'part of skull'; *submarine* = 'sandwich'; *pitcher* = 'jug'.

[6]While individuals vary as to exactly which sentences fall where, a disambiguation system with claims to psychological reality should be in accord with the general consensus.

data on the subjective semantic distance between many different concepts, and then see how this translates into "physical distance" in Frail (or other representation of choice). Word association norms (*e.g.*, Postman and Keppel 1970) may provide an initial base for such a set of data.

Getting the thresholds right in marker passing is important not only so that Polaroid Words can confidently use marker passing as a disambiguation cue, but also because properly deployed marker passing has many other uses in cognitive modeling; these include context determination and explanation finding (Charniak 1983).

3.2. Impatience

Recent psycholinguistic research (*e.g.*, Marslen-Wilson and Tyler 1980) has emphasized human language understanding's following the principle of "do it as early as possible" — that interpretation happens as soon as sufficient information is available, and the interpretation of earlier parts of a sentence is used to guide the interpretation of the later parts. This principle is followed by Absity (Hirst 1983a, 1983b), the system of which Polaroid Words form a part.

There are very few data, however, on how quickly lexical disambiguation takes place in humans. Almost all studies of disambiguation have only considered the special case in which sufficient disambiguation information is present when the ambiguous word occurs; often, the test word is the last of the sentence. Under these conditions, disambiguation is extremely rapid — between 100 and 200 msec (Lucas 1983).

But what of cases in which the necessary information is not initially present? How long will people wait for it before jumping to a conclusion with partial information? The following examples are both processed without error, although the final noun phrase has to be interpreted before *book* can be disambiguated as 'literary work' or 'printed volume':

(9) Nadia's favorite book is *The House at Pooh Corner*.

(10) Nadia's favorite book is her signed first edition of *The House at Pooh Corner*.

Thus, in at least some cases people will wait until the end of the clause. On the other hand, it is my intuition that *fans* is disambiguated as 'devotee' in (11) as soon as the verb *lined up* is processed:

(11) The fans were lined up for hours to buy the Stones tickets.

even though one can construct quite reasonable (albeit less probable) sentences that start the same way and in which *fan* means 'air-moving device':

(12) The fans were lined up awaiting their final factory inspection.

This suggests that PWs should use a *cumulating evidence* approach and jettison unlikely alternatives quickly if there is no positive evidence for them. That is, one does not make an immediate best guess, but one does make a reasonable guess as soon as there is enough information to do so, even if one cannot be definite. This has the advantage of helping to prevent combinatorial explosion.

However, I have been loath to consider using this approach in Polaroid Words, in view of the dearth of data on the corresponding human behavior and the fuzziness of the whole notion. Any interim solution would have to fall back on the magic numbers we have already bemoaned. Nevertheless, PWs do use the relative frequency of the various meanings of an ambiguous word in some of their decisions, but since we know little yet of how people use frequencies in disambiguation (see Hirst 1983a) we have limited their use in PWs to tidying up loose ends at the end of a sentence. Another possibility might be to add a mechanism that watches out for looming combinatorial explosion and forces PWs to make an early guess if it senses danger. (In Hirst 1983a, I discuss how the demands of structural disambiguation may force PWs to make an early decision, also in order to avoid combinatorial explosion.)

3.3. Cowardice

Despite everything we have said above, it is obvious that some sentences are genuinely ambiguous to people. It is therefore inappropriate for a disambiguation process to jump to a conclusion in such cases or to take extraordinary measures or go to heroic efforts to resolve residual problems. That is, PWs should be afraid to jump to a conclusion if the leap seems too great. If reasonable efforts fail, they can always ask the user what he or she really meant:

(13) USER: I need some information on getting rid of moles.

SYSTEM: Are you troubled by unsightly blemishes, by those lovable but destructive insectivorous garden pests, by uterine debris, or by enemy secret agents that have penetrated deep into your organization?

(PWs do not actually have such a natural language response component.)

4. Conclusion

The notion of jumping to a conclusion when there is "enough" evidence is an inherently fuzzy one, but one that is clearly involved in word disambiguation, as well as other cognitive processes. The "easy" solution, using magic numbers in a delicately balanced knowledge base, is obviously inadequate. A better understanding of the time course of human word disambiguation is needed before the psychological reality of Polaroid Words can be improved.

References

CHARNIAK, Eugene (1983). "Passing markers: A theory of contextual influence in language comprehension." *Cognitive science*, 7(3), July-September 1983, 171-190.

CHARNIAK, Eugene, GAVIN, Michael and HENDLER, James (1983). "The Frail/NASL reference manual." Technical report CS-83-06, Department of Computer Science, Brown University, February 1983.

COLLINS, Allan M and LOFTUS, Elizabeth F (1975). "A spreading-activation theory of semantic processing." *Psychological review*, 82(6), November 1975, 407-428.

HIRST, Graeme (1983a). *Semantic interpretation against ambiguity*. Doctoral dissertation (technical report CS-83-25), Department of Computer Science, Brown University, December 1983.

HIRST, Graeme (1983b). "A foundation for semantic interpretation." *Proceedings, 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, Massachusetts, June 1983. 64-73. (Also available as technical report CS-83-03, Department of Computer Science, Brown University, January 1983.)

HIRST, Graeme and CHARNIAK, Eugene (1982). "Word sense and case slot disambiguation." *Proceedings, National Conference on Artificial Intelligence (AAAI-82)*, Pittsburgh, August 1982. 95-98.

LUCAS, Margery (1983). "Lexical access during sentence comprehension: Frequency and context effects." *Proceedings, Fifth Annual Conference of the Cognitive Science Society*, Rochester, New York, May 1983.

MARSLEN-WILSON, William D and TYLER, Lorraine Komisarjevsky (1980). "The temporal structure of spoken language understanding." *Cognition*, 8(1), March 1980, 1-71.

ONIFER, William and SWINNEY, David (1981). "Accessing lexical ambiguities during sentence comprehension: Effects of frequency of meaning and contextual bias." *Memory and cognition*, 9(3), May 1981, 225-236.

POSTMAN, Leo and KEPPEL, Geoffrey (editors) (1970). *Norms of word association*. New York: Academic Press, 1970.

REDER, Lynne M (1983). "What kind of pitcher can a catcher fill? Effects of priming in sentence comprehension." *Journal of verbal learning and verbal behavior*, 22(3), April 1983, 189-202.

SCHANK, Roger and ABELSON, Robert (1977). *Scripts, plans, goals and understanding*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1977.

SEIDENBERG, Mark S, TANENHAUS, Michael K, LEIMAN James M and BIENKOWSKI, Marie A (1982). "Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing." *Cognitive psychology*, 14(4), October 1982, 489-537.

SMALL, Steven L (1980). *Word expert parsing: A theory of distributed word-based natural language understanding*. Doctoral dissertation (technical report 954), Department of Computer Science, University of Maryland, September 1980.

SWINNEY, David (1979). "Lexical access during sentence comprehension: (Re)Consideration of context effects." *Journal of verbal learning and verbal behavior*, 18(6), December 1979, 645-659.

Reservations about Qualitative Models

James D. Hollan

Edwin L. Hutchins

Navy Personnel Research and Development Center
San Diego, California 92152

Abstract

Very little of the knowledge that an operator of a complex physical system brings to the job is purely quantitative in form. Virtually all of an operator's knowledge can be represented as qualitative relations or quasi-quantitative relations such as rough proportionalities. The realization that computer-based instruction systems need to provide instructions and explanations in terms that students can use, that is, often in qualitative terms, has led to recent efforts in cognitive science and artificial intelligence to develop qualitative simulation models of complex dynamic systems. In this paper we discuss theoretical and pragmatic problems involved in using qualitative models to support automated explanation facilities.

Much recent work in cognitive science has addressed the nature of qualitative reasoning. A number of studies have provided detailed analyses of protocols of subjects reasoning about a variety of physical systems (Larkin, 1983; Williams, Hollan, and Stevens, 1983) and have documented the extensive use of qualitative forms of reasoning. In addition, there have been many recent attempts by AI researchers to develop qualitative calculi which might be used by a program to permit it to reason about various classes of physical devices or to provide qualitative explanations of the operation of such devices (de Kleer, 1975; de Kleer, 1979; Forbus, 1981; Forbus, 1982; de Kleer & Brown,1983). The fact that such qualitative forms of reasoning appear to be important in understanding the operation of physical systems might lead one to believe that qualitative simulations will be the most effective way of building automated qualitative explanation systems. Our experience in the development of Steamer (Hollan & Hutchins, 1984), however, leads us to a quite different view of human reasoning about physical systems and motivates us to discuss a number of limitations of qualitative models for supporting qualitative explanations.

We have two fundamental reservations about the use of qualitative models as the basis of qualitative explanation facilities. While there is strong evidence that mental arithmetic plays little role in everyday calculation tasks (Lave, Murtaugh, & de la Rocha, in press), we are struck by how much of human reasoning seems to rely on the use of judgements that are more precise than could be produced by purely qualitative calculation. Much reasoning that at first blush appears totally qualitative can, upon closer inspection, be seen to involve approximate magnitude estimates. Our first reservation is that since purely qualitative simulations will not support the processing of even approximate quantities, they cannot be the basis of explanations that contain such quasi-quantitative information.

Paper submitted to the *Conference of the Cognitive Science Society*, Boulder, Colorado, June 28-30, 1984. This research was supported by the Personnel and Training Research Program of the Office of Naval Research under Work Request N0001483WR30106/07, Work Unit NR 667-507. This paper has benefited from the critical comments of Johan de Kleer and Ken Forbus.

Second, purely qualitative models are underdetermined in the sense that they cannot account for many qualitative aspects of the behavior of some systems. For example, in negative feedback systems the qualitative character of the response of the system to perturbation (damping versus hunting) is not determined by the device topology alone but is dependent upon the actual *quantitative settings* of a number of component parts. Certain settings will result in a damping behavior and other settings will result in hunting.

Quasi-quantitative Reasoning

Close observation of people reasoning about physical systems shows that there is considerable reliance on the use of quasi-quantitative knowledge. Consider for example the knowledge and models that people use to give *ball-park* estimates of the rough magnitudes of various aspects of events. The knowledge that supports such predictions is not entirely qualitative, since quantity is roughly specified, and yet not entirely quantitative since the quantities are not computed exactly. Such models are essential for our predictions about many aspects of everyday as well as technical life. Examples abound: *How far can I step? How long will it take to bring this water to a boil? How much rudder is required to turn a ship around.* We are all quite capable of making these types of predictions. We do not make exact predictions but we have a very definite *feel* for the general magnitude of the quantities involved.

These types of *ballpark* estimates appear well designed to interact with feedback from the world. One produces an estimate that is in the neighborhood of the actual value. That estimate is applied and can then be tuned based upon the feedback received from having made an action based on the estimate. Much of what goes under the heading of *getting the feel of it* is probably the acquisition and tuning of this sort of quasi-quantitative knowledge. We believe that it underlies much of the reasoning in many domains. For example, the mental computations that Micronesian navigators make to determine the distance they have covered along their course seem to depend on the use of this type of knowledge (Hutchins, 1983). Likewise, the reasoning involved in understanding the operation of a heat exchanger (Williams, Hollan, & Stevens, 1983) or a steam reducing valve concerns predictions not only about the direction of changes but also the approximate magnitudes of the changes. The important connection between these domains is in the specification of the *units* in which the change is expressed. Just as the navigator expresses the changes he monitors in terms of units that suit the computations he needs to make (nautical miles or location under star bearings, depending on the computational apparatus he has at his disposal) so the user of a mental model of a steam plant or any other device is likely to assess the changes of variables about which he is concerned in terms of units tailored to support the necessary subsequent computations. Such units might express something like *sufficient to cause a state transition* or, in the more familiar domain of driving, a speed increase might be specified as *enough to get me past that car before the on-coming truck arrives.*

How are such units arrived at and how are measurements made using them? Imagine, for example, a golfer involved in trying to sink a put. He has never had just this lie but he has to calculate how hard to hit the ball and where to hit it to account for the speed and slope of the green. One might conjecture that he mentally imagines the trajectory of the ball and has a model of the effects along the way. A model of the initial velocity of the ball (not expressed in feet per sec, but in terms of visualizing it moving across the surface of the green: *analog* units) and a model of how the ball will lose speed (depending upon the hardness of the green, the length of the grass, etc) and a model (also in terms of the imagined effects on the trajectory of the ball) of the accelerative effects of the slope of the surface. One can construct the same kind of scenario to account for numerous other everyday and technical endeavors. The important point though is the omnipresence of such quasi-quantitative units in

reasoning about physical phenomena. Currently we know virtually nothing about how they are processed, how they are managed, or how they are connected to a performance.

Underdetermined Qualitative Models

Another precipitating motivation for this paper is a consideration for how difficult qualitative simulations are to construct. An appreciation for this difficulty arises from our experience with the Steamer system, from efforts we have made in building qualitative simulations (Williams, Hollan, & Stevens 1983), and from a lack in qualitative simulations of the types of quasi-quantitative reasoning devices that we have detailed above. The very interesting representational advances of de Kleer and Brown (1982, 1983) bear witness to the difficulty of constructing purely qualitative calculi that can support reasoning about physical devices. Part of this difficulty arises from the desire to make the qualitative calculi general and to avoid assumptions of system function in the specification of component device structure. This makes it possible for a single set of device models to qualitatively simulate the behavior of a large class of systems.The *no-function-in-structure* principle, while crucial to the construction of the *physics* that de Kleer and Brown (1982) are in search of, engenders a number of problems for supporting a qualitative explanation system and for capturing important qualitative and non-qualitative aspects of people's reasoning processes. For example, people normally violate this principle in reasoning about physical devices and systems. In fact, in dealing with a reducing valve, as they do in a recent paper (de Kleer & Brown, 1982), even they appear to need to violate their principle in order to explain the damping that occurs in this system. Their envisioning treatment of the reducing valve pulls a *damping rabbit* out of a hat. They provide an English language rendering of their program's explanation:

> An increase in source pressure increases the pressure drop across the valve. Since the flow through the valve is proportional to the pressure across it, the flow through the valve also increases. This increased flow will increase the pressure at the load. However, this increased pressure is sensed by [the sensing line] causing the diaphragm to move downward against the spring pressure. The diaphragm is mechanically connected to the valve , so the downward movement of the diaphragm will tend to close the valve, thereby pinching off the valve. *Because the flow is now restricted the output pressure will rise much less than it otherwise would have and thus remain approximately constant.* (de Kleer & Brown, 1982, p2; emphasis added)

There is, in fact, no way to predict from a purely qualitative analysis whether a negative feedback system is stable or unstable. This aspect of the behavior of the device depends upon the *quantitative* relationship of the controlled variable and the controlling action. Without knowing this relationship it is impossible to know if the value of the controlled variable will stabilize or continue to oscillate.

Others have also noted limitations with qualitative modeling. For example, Simmons (1983) has pointed out difficulties in expressing features, such as shape, in qualitative terms, that qualitative models are necessarily ambiguous "in that a single qualitative representation maps to many real-world situations", and that most users of qualitative representations have needed to make use of quantitative knowledge to deal with ambiguities (Simmons, 1983; Simmons & Davis, 1983).

February 28, 1984

Function in Structure

When an expert explains the behavior of such a device he assumes it is either working properly or somehow malfunctioning. Within the context provided by that assumed behavior a description of the propagation of effects in qualitative terms can be rendered. The determining variables are not described quantitatively, but are described relative to functionally embedded criteria. Consider the following typical statement about the behavior of a reducing valve: "If the gain is too high, then the system will hunt." The surface form of this statement is that of a prediction, but it is, in fact, not a prediction because the criteria for deciding the truth of the antecedent (whether or not the gain is too high) are based on the observation of the consequent. Seen in this light, the statement is a tautology, but a very useful one. It is based on the premise that there are correct settings for these parameters that will produce appropriate behavior. Certain abnormal behaviors are seen as diagnostic of deviations of parameters from their "correct" settings.

The utility of such functionally embedded specifications of parameter settings is that the device may be tuned without quantitative knowledge, as long as it was designed so that it will work with some settings. The tuning process need only be capable of interpreting the observed behavior as a symptom of a particular qualitative relation of a parameter to its "correct" setting. Based on this information, the operator can often hill climb to the correct setting without having any idea of its actual quantitative value. But notice that this strategy requires the assumption that the device can function properly. It also requires a description of proper functioning and a set of correspondences between device behavior patterns on the one hand and the relations of controlling parameters to their correct settings on the other. As we mentioned earlier, it is our contention that there is something very important about this form of interaction with the world. One begins with what is essentially open-loop ballistic behavior in the world, which requires quasi-quantitative representations and assumptions about function, and then one becomes part of a closed loop system, making use of qualitative evaluations to control the tuning process.

Conclusions

Qualitative physics is an important line of AI research, but models based on qualitative calculi may be inappropriate as a base for providing qualitative explanation in automated tutorial systems because 1. qualitatative calucli fail to represent important classes of features of events and objects, 2. they are fundamentally underdetermined with respect to some physical behaviors, and 3. the principles that guide the representation of events and devices in qualitative models (*no-function-in-structure* and the derivation of device behavior from component interactions) conflict with observed structures of human interpretation and explanation of device behavior.

February 28, 1984

References

de Kleer, J. Qualitative and quantitative knowledge in classical mechanics, MIT AI-TR-352, 1975.

de Kleer, J. Causal and teleological reasoning in circuit recognition, MIT AI-TR-529, 1979.

de Kleer, J., & Brown, J. S. Assumptions and ambiguities in mechanistic mental models, *Mental Models*, Gentner, D, & Stevens, A. (Eds.), Erlbaum, 1983.

de Kleer, J., & Brown, J. S. Foundations of envisioning, Proceedings of the National Conference on Artificial Intelligence, 434-437, 1982.

Forbus K. A study of qualitative and geometric knowledge in reasoning about motion. Master's Thesis, MIT, 1980; also MIT AI-TR-615, 1981.

Forbus, K., & Stevens, A. Project Steamer: III. Using qualitative simulation to generate explanations of how to operate complex physical devices, NPRDC TN 81-25, 1981.

Forbus, K. Qualitative process theory, MIT-AI-664, 1982.

Hollan, J. D., & Hutchins, E. L. STEAMER: An interactive inspectable simulation-based training system, manuscript submitted for publication.

Hutchins, E. L. Understanding micronesian navigation, *Mental Models*, Gentner, D, & Stevens, A. (Eds.), Erlbaum, 1983.

Larkin, J. The role of problem representation in physics. *Mental Models*, Gentner, D, & Stevens, A. (Eds.), Erlbaum, 1983.

Lave, J., Murtaugh, M., & de la Rocha, O. The dialectical Constitution of Arithmetic Practice. In *Everyday Cognition: Its Development and Social Context*, B. Rogoff and J. Lave (eds) Cambridge: Harvard University Press (in press).

Simmons, R. G. The use of qualitative and quantitative simulations. Proceedings of The National Conference on Artificial Intelligence, 364-368, 1983.

Simmons, R. G., & Davis, R. Representations for reasoning about change, MIT AIM-702, 1983.

Williams, M., Hollan, J., & Stevens, A. Human reasoning about a simple physical system, *Mental Models*, Gentner, D, & Stevens, A. (Eds.), Erlbaum, 1983.

Ambiguity Resolution in the Absence of Contextual Bias

Susan B. Hudson
Michael K. Tanenhaus
University of Rochester

Natural languages contain a high proportion of words that have multiple meanings. For example, organ can be a type of musical instrument or an internal body part and rose can be a flower or the past tense of the verb to rise. Understanding how the language processor deals with lexical ambiguity can provide important insights into the structure of the human natural language processing system. As a consequence, lexical ambiguity has been the focus of a great deal of psycholinguistic research. The fundamental issue has been whether one meaning or more than one meaning of an ambiguous word is typically accessed when the word is presented in a context that biases one of its possible meanings. A number of studies using variations of the so-called cross-modal lexical priming paradigm seem to have resolved this issue in favor of the multiple meanings alternative.

In the cross-modal priming paradigm, the subject listens to a sentence and responds to a visual target word presented at strategic points during the sentence. The availiability of component meanings of an ambiguous word is inferred from reaction times to either name or make a lexical decisiion to a target word related to either its contextually appropriate or inappropriate meaning. Lucas (1983), Onifer and Swinney (1981) Seidenberg, Tanenhaus, Leiman, and Bienkowski (1982), Swinney (1979) Tanenhaus and Donnenwerth-Nolan (in press) and Tanenhaus, Leiman, and Seidenberg (1979) have demonstrated that response times to targets presented immediately after an ambiguous word are equally facilitated when they are related to either the contextually biased or unbiased meaning. When targets are delayed until 200 msec after the ambiguous word, priming obtains only to targets related to the appropriate meaning. These results indicate that multiple meanings of ambiguous words are initially accessed regardless of contextual bias and context is then used to rapidly select the contextually appropriate meaning.

An important question unanswered by these studies is what happens when the context is not strongly biased towards one reading of the ambiguous word. Is selection of a single meaning rapidly made on the basis of lexical information alone such as the frequency of the alternative readings or are multiple readings held onto until biasing information becomes available or until some natural decision point such as a clause boundary?

Only one study using a cross-modal priming paradigm has addressed this issue. Seidenberg et al (1982) had subjects name a target word (e.g., SIP) that followed a sentence fragment which ended in either an ambiguous word with one meaning related to the target (STRAW), an unambiguous word related to the target (e.g. SODA) or a word unrelated to the target (e.g. WHEAT). If multiple meanings of ambiguous words are accessed, targets related to the ambiguous word should have been equally facilitated following the ambiguous word and the unambiguous word when compard to the unrelated control condition. If, however, only one meaning was accessed, then the ambiguous word and the target would be unrelated on some proportion of the trials and less facilitation would obtain to the target when it followed an ambiguous word than when it followed an unambiguous word. The results were clear. When the target followed immediately after the ambiguous word, equal facilitation

obtained in the ambiguous and unambiguous related conditions. When a 200 msec delay was introduced between the ambiguous word and the target word, targets in the ambiguous related condition showed approximately half as much facilitation as targets in the related unambiguous condition, indicating that listeners had already selected one meaning. Seidenberg et al argued that meaning selection takes place rapidly even in the absence of contextual bias. However, there is a serious problem with the Seidenberg et al study. The ambiguous word was always the final word of a sentence fragment. Thus in the 200 msec delay condition, the brief pause may have signalled the subjects that the sentence fragment had ended. This might have triggered meaning selection under circumstances where multiple meanings would normally have been maintained in continuous speech.

The present study explored the question of whether multiple readings of ambiguous words are maintained for longer that 200 msec in the absence of biasing context using a cross-modal lexical decision task and continuous speech. Subjects listened to sentences containing an ambiguous word or an unambiguous word related to one of its meanings. A target word that was either related to one of the meanings of the ambiguous word or unrelated was presented at one of three points: (1) immediately after the ambiguous word, (2) after 500 msec, and (3) at the clause boundary (several words downstream). When the target is presented immediately after the ambiguous word both meanings should be available resulting in equal facilitation for related targets when they follow ambiguous and unambiguous words. The most interesting question is what will happen at the 500 msec delay. If multiple meanings are retained in the absence of a biasing context, the same pattern should obtain as in the immediate condition. If, however, one meaning is selected, more facilitation should obtain to related tagets following unambiguous words than following ambiguous words. The results at the clause boundary condition should provide additional information about the time course of ambiguity resolution. Given that previous research suggests that the clause boundary is a major decision and integration point, we would expect to see ambiguity resolution by this point.

Method

Subjects. Study particpants were 72 students at Wayne State University who received course credit for their participation.

Materials. The materials were constructed from thirty-two noun-noun ambiguous words. Each ambiguous word was placed in the first clause of a two clause sentence. The first clause was constructed so as not to bias either meaning of the ambiguous word using the following procedure. Norms were collected in which subjects completed sentence fragments. These were then scored according to which meaning was chosen. Items retained for the study were those in which the meaning chosen occurred with the same proportion when the word was presented in a sentence fragment as when the word was presentd in isolation. The unambiguous control stimuli were formed by replacing the ambiguous word with a word related to its alternate readings. For example, the ambiguous word PUPILS was replaced with the unambiguous words CORNEA and STUDENTS. Targets were selected that were related to each of the meanings; in the example, these were EYE and SCHOOL. Unrelated targets were then selected that were matched to the related targets in length and frequency. In the example, GAME was the unrelated target matched with EYE and PART was the unrelated target matched with STUDENTS. Sample sentence are presented in Table 1.

Table 1
Sample Sentences and Targets

Bill examined his pupils (corneas) carefully this morning, because he thought that there was something wrong with his eyes.

Related Target: EYE
Unrelated Target: GAME

Bill examined his pupils (students) carefully this morning, because he wanted them to do well on the exam.

Related Target: SCHOOL
unrelated Target: PART

The experimental design was a 2 X 2 X 2 X 3 with the factors being Meaning (the meaning of the ambiguous word related to the target), Ambiguity (whether the sentence contained an ambiguous or unambiguous word, Relatedness (whether the target was related or unrelated to the word in the sentence, and Target presentation point (targets were presented immediately after the ambiguous word, 500 msec after the ambiguous word, and at the clause boundary which was usually three of four words after the ambiguous word. In order to keep subjects from focusing on the first clause, filler stimuli were included in the experiment in which target words were presented at various points in the sentence. Filler stimuli were also included in which the target was a non-word. The targets were non-words on half of the trials.

Procedure. Each subject listened to the sentences while seated at a CRT. Subjects made lexical decisions to the target stimuli using the keyboard of a microcomputer. On one-quarter of the trials the subject also answered a true-false comprehension question before beginning the next experimental item.

RESULT AND DISCUSSION

The data of interest are the mean lexical decision times for the related ambiguous, unrelated ambiguous and unrelated conditions which are presented in Table 2. In the immediate condition, lexical decisions to targets in the amiguous related and unambiguous related condition showed approximately the same amount of facilitation (39 and 35 msec., respectively) compared to their unrelated control conditions. A similar pattern obtained in the 500 msec condition with 54 msec of facilitation in the amiguous related condtion and 44 m sec in the unambiguous related condition. At the clause boundary, ambiguous related targets were facilitated by 7 msec compared to 37 msec in the unambiguous related condition.

Table 2

Lexical Decision Times and Facilitation Scores in msec

Condition	Target Presentation Point: Immediate	500 msec Delay	Clause Boundary
Ambiguous Related	846	739	779
Ambiguous Unrelated	885	793	786
Facilitation	39	54	7
Unambiguous Related	841	733	757
Unambiguous Unrelated	876	777	794
Facilitation	35	44	37

ANOVAS were conducted using both subject and item condition means. Meaning, Target Relatedness, and Ambiguity were crossed in both analyses. In addition there were eight Presentation lists, which were nested within subjects. Target presentation point was nested within subjects and crossed with items. The analysis revealed a significant effect of Target Relatedness both by subject $F(1, 48) = 27.89$, $p < .01$ and by item $F(1, 93) = 8.74$, $p < .01$. The Ambiguity by Relatedness and Ambiguity by Relatedness by Target Presentation Point interactions did not approach significance. Separate analyses at each presentation point revealed significant effects of target relatedness and no interaction between ambiguity and target relatedness in the immediate and 500 msec conditions. This is just the pattern of results consistent with multiple access. At the clause boundary condition, the effect of relatedness was only marginally significant by subject and by item and the relatedness by ambiguity interaction approached significance. This pattern obtained because only targets following unambiguous words showed facilitation compared to the unrelated condition.

The results in the immediate condition replicate previous research in demonstrating that multiple meanings of ambiguous words are initially accessed. In contrast to previous research with biasing contexts in which ambiguity resolution took place within 200 msec, the 500 msec condition continued to show a multiple access pattern. Thus multiple meanings continue to be available in the absence of biasing context. The results at the clause boundary indicated that ambiguity resolution had taken place by this point. In contrast to the immediate and 500 msec conditions, less facilitation obtained to targets in the ambiguous related condition than in the unambiguous condition. Although the amount of facilitation in the ambiguous related condition was small, a post hoc analysis based on our frequency norms indicated that targets related to the dominant reading of the ambiguous word were showing a significant amount of facilitation while targets related to subordinate readings were not being facilitated. In contrast, targets related to both dominant and subordinate readings were being significantly facilitated in both the immediate and 500 msec conditions.

References

Fodor, J.A., Bever, T.G., & Garrett, M. (1974). The Psychology of Language New York: McGraw-Hill.

Lucas, M.M. (1983). Lexical access during sentence comprehension: Context effect, frequency effects, and decision processes. Unpublished doctoral dissertation. University of Rochester.

Onifer, W. and Swinney, D.A. (1981). Accessing lexical ambiguities during sentence comprehension: Effects of frequency of meaning and contextual bias. Memory and Cognition, 9, 222-236.

Seidenberg, M.S., Tanenhaus, M.K., Leiman, J.M., and Bienkowski, M. (1982). Automatic access of the meanings of ambiguous words in context: Some limitations of knowledge-based processing. Cognitive Psychology, 14, 489-537.

Swinney, D.A. (1979). Lexical access during sentence comprehension: (Re) consideration of context effects. Journal of Verbal Learning and Verbal Behavior, 18, 645-659.

Tanenhaus, M.K. and Donnenwerth-Nolan, S. (in press) Syntactic context and lexical access. Quarterly Journal of Experimental Psychology.

Tanenhaus, M.K., Leiman, J.M., and Seidenberg, M.S. (1979) Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. Journal of Verbal Learning and Verbal Behavior, 18, 427-440.

LEVELS OF PROCESSING IN METAPHOR COMPREHENSION*

Janice Johnson
Institute of Human Learning, University of California, Berkeley

In a metaphor a topic is described in terms of a vehicle. The topic and vehicle terms generally refer to two diverse, conceptual or experiential, domains. Several theories of metaphor assume that metaphoric comprehension involves a mapping from vehicle to topic of vehicle properties or aspects. For example, Ortony (1979) proposes that in a metaphor highly salient attributes of the vehicle are matched with low salient attributes of the topic. The matched (i.e., shared) attributes need not be identical, but must have high similarity. Glucksberg, Gildea, and Bookin (1982) suggest that metaphor comprehension involves instantiation, in terms of the topic, of a small set of salient properties of the vehicle. Tourangeau and Sternberg (1982) criticize the notion that the topic and vehicle in a metaphor have matching (i.e., shared) attributes, yet still consider that metaphor comprehension requires a mapping of attributes between vehicle and topic. They propose that the attributes of the vehicle domain must be transformed to apply in the topic domain. In a similar vein, Verbrugge and McCarrell (1977) argue that metaphor comprehension involves a "novel schematization of the topic domain" (p. 494) in terms of transformational and structural invariants of the vehicle domain.

These approaches all propose some mapping from vehicle to topic of vehicle aspects. They differ in their definitions of the aspects and in the degree to which they see vehicle aspects as being transformed rather than matched in the topic. Combining and extending these approaches, I propose here a metaphor comprehension model which allows for varying degrees of transformation in the mapping of vehicle aspects, that is, different levels of accommodation of the vehicle aspects to the semantics of the topic. I formalize these levels in terms of different kinds of mapping processes.

A Semantic Mapping Model of Metaphor Comprehension

I propose that the semantic process of comprehending a metaphor involves selecting some facets or aspects of the vehicle that are potentially applicable to the topic, then mapping these facets to the topic to evaluate analytically the appropriateness of the mapping. The mapping is done by means of semantic combinators (i.e., mapping functors); these are semantic transformations that convert one or more semantic facets into other different facets--combinators can apply on topic or vehicle facets. In terms of structure, the topic and vehicle in a metaphor refer, in the subject, to knowledge representations I call obs (short for "object schemes"). An ob is a complex mental structure that stands for all the discriminative, manipulative, and functional aspects or facets of a distal object (Pascual-Leone, Goodman, Ammon, & Subelman, 1978); it is thus the mental representation of a thing in the environment. The notion of an ob is analogous to other notions of memory structure, such as "frame," "schema," and "prototype." Facets are the functional components (properties or relations) of an ob that emerge from goal-directed interaction with the object; facets are not properties inherent in the object itself, but are constraints the subject has experienced.

I propose that the semantic processing of metaphors takes place in successive moments. In a first moment of global processing/mapping, the semantic relation between topic and vehicle is investigated by way of shallow, more or less concrete, semantic content processing. Global processing can yield an adequate metaphor interpretation only if topic and vehicle share (low-level) content facets. A further moment of deeper, analytical processing/mapping involves analytical

elaboration or modification of the global meaning in light of the detailed meanings of both topic and vehicle. Analytical processing, more so than global mapping, is guided by and obedient to semantic constraints imposed by the topic; it thus represents an accommodation of the vehicle facets to the semantics of the topic ob. A first level of analytical processing involves modification of the global meaning; a second level may involve a movement from one topic facet to another. I will make the processing levels clearer when I describe semantic combinator kinds that are instances of the levels. In order to specify the semantic combinator kinds, I developed a method of coding metaphor interpretations.

Method

I collected metaphor interpretations from children and adults. I report here data for the adult subjects: 24 students at York University in Toronto. I interviewed subjects individually and asked them to interpret orally each of 19 metaphoric sentences. This paper reports results for 6 of the metaphors; these were constructed by combining, in a sentence frame of the form "____ was a ____", each of three vehicle nouns (rock, mirror, butterfly) with each of two topics (My sister, My shirt) to form sentences such as "My sister was a mirror," "My shirt was a rock," etc. Subjects were encouraged to give as many interpretations as they could for each item.

Coding of metaphor interpretations. According to the processing model, in interpreting a metaphor the subject maps facets from vehicle to topic by means of some semantic combinator. In coding a metaphor interpretation, one first infers the actual vehicle facet(s) that underlie the interpretation and then the kind of semantic combinator that must have applied on the facet(s) to generate the interpretation. The three main kinds of semantic combinators I propose (and the only ones space limits permit me to discuss here) are the Identity, Analogy, and Predicate combinators. Identity and Analogy are types of between-obs combinators that map facets from vehicle to topic.

The Identity semantic combinator is an instance of the initial, global level of metaphor processing. In an Identity mapping the subject finds a facet in the vehicle ob that has (or could have) the same name and semantic definition in the topic ob and does a direct mapping of the facet from vehicle to topic. The facet is mapped without any change in meaning. An example is the following response to the sentence "My sister was a rock": "Maybe she felt to the physical touch very hard." The rock facet used is "hardness", the defining statement of which could be <<rocks do not change shape under the application of external physical force>>. Here the subject selects a salient facet of the rock ob and maps it to the sister ob without changing the sense of the facet. For a response to be scored as an Identity the mapped facet(s) must be compatible with the semantics of the topic ob. A second example is the following response to "My shirt was a mirror": "It could actually be a mirror--made out of some kind of material that would actually reflect." This response is based on one or both of two mirror facets: a facet corresponding to the optical "image" produced by the mirror (<<a mirror gives back a reproduction or likeness in two dimensions of whatever is in front of it>>) and a facet corresponding to the mirror's ability to reflect light (i.e., its "shininess").

The Analogy combinator is an instance of the first level of analytical metaphor processing. In an Analogy mapping the facet(s) emerging from global processing undergo a change in sense as they apply from vehicle to topic. The change in sense represents an accommodation of the vehicle facet(s) to the semantics of the topic. An example is the following response to "My sister was a rock": "She was very firm and unyielding sort of like a rock . . . his sister is like a rock as far as the way she behaves or acts, like hard as a rock." Here the "hardness"

vehicle facet is accommodated to the topic ob through a process of constructive abstraction, whereby topic-relevant content is inserted into the vehicle facet structure (see definition of "hardness" rock facet above): "does not change shape" becomes "does not change behavior" and "external physical force" becomes something like "verbal instruction" or "psychological pressure." In an Analogy mapping, the vehicle facet and the (semantically different) topic facet it maps are related by way of a higher-level (i.e., generic) superfacet that subsumes the topic and vehicle senses. A second example is the following response to "My shirt was a mirror": "Maybe it would mean that you saw someone else with the same shirt as you." This response is based on the mirror "image" facet described above, but in this case the facet is mapped with a change in sense; that is, it is applied with the sense of resemblance rather than optical reproduction.

The Predicate is a type of within-obs combinator that applies within the topic ob following a between-obs mapping; it is an instance of the second level of analytical processing. The Predicate serves to express the result of a between-obs mapping in terms that closely conform to the pragmatics of the topic ob. To this end, the subject elaborates the initial mapping in terms of a concept or an instantiation that is relevant to the topic, but not to the vehicle. An example is the following response to "My sister was a rock": "Whenever I think of a rock I think of something hard, so maybe your sister is cold or unfriendly. You are not very close with your sister." Again, this response is based on Analogical mapping of the "hardness" rock facet (with the sense of non-responsiveness), but here the subject elaborates the initial mapping in terms of unfriendliness and psychological distance--concepts that are relevant for describing persons, but not rocks. In contrast to the Analogy, in a Predicate response the topic-relevant concept or instantiation is cued by the generic superfacet, but is not subsumed under it. Another Predicate example is the following response to "My shirt was a mirror": "My shirt was a reflection of myself, so if I had on a white shirt I'd be a conservative, and if I had on a wild shirt I would be a wild person." Here the shirt is, Analogically, an "image" of the wearer's personality, and the subject instantiates this Analogy in terms of types of shirts and personalities.

Identity, Analogy, and Predicate are three main kinds of semantic combinators that are instances (and most characteristic) of three proposed levels of metaphoric processing: global, analytical-1, and analytical-2. Elsewhere (Johnson & Pascual-Leone, 1984) I have described additional kinds of combinators and developmental data that support the validity and reliability of the coding method (see also Johnson, Fabian, & Pascual-Leone, in press). Here I use the notions of semantic combinators and vehicle facets to characterize adult processing of metaphors. In the Results I use the combinator names introduced above to refer to the processing levels; responses coded with other kinds of combinators are assimilated to the appropriate level.[1]

Results and Discussion

Level of processing. Subjects typically gave more than one interpretation for an item; the mean number of responses across subjects and items was 2.6 (the modal numbers of responses were 2 and 3). Of the total number of responses, 17% are at the global level (i.e., are of the Identity type), 27% are at the Analogy level, and 50% are at the Predicate level (6% are below the global level; e.g., responses that violate the reality constraints of the topic ob or that do not make a mapping from vehicle to topic). Thus, adults use all three processing levels when interpreting metaphors, but more often respond at the higher levels. Eighteen out of 24 subjects gave responses at all three levels.

As one would expect, the Identity level is used more often for items with the shirt topic (27%) than for those with sister (7%). Shirt shares more physical facets with the vehicle obs than does sister (i.e., Identity mappings are more likely to be compatible with the semantics of shirt). In order to say something meaningful about sister, in light of the vehicles, the vehicle facets must be transformed: For items with the sister topic, 33% of the responses are at the Analogy level and 53% are at the Predicate level; for shirt, 21% are Analogies and 46% are Predicates. "My shirt was a butterfly" has the highest rate of Identity responses (39%); there are a number of butterfly facets (e.g., colorful, light, soft) that are directly compatible with possible shirt facets. "My sister was a mirror" has the highest rate of Analogy responding (51%); the most frequently used mirror facet is the mirror "image"--subjects make numerous Analogies concerning resemblance in looks or behavior. All items yield a high rate of Predicate responding (ranging from 40-59%), but "My sister was a rock" has the highest rate; here the most frequently used vehicle facets refer to the hardness, strength, and immobility of rock, and in sister these aspects are transformed into Predicates expressing emotional coldness, strength of character, and stubbornness.

Vehicle facets. It is often proposed that facets that have high salience in the vehicle are selected for mapping to the topic (e.g., Ortony, 1979). Results of the current study show that some vehicle facets are more frequently used than others in the metaphor interpretations (one might characterize these frequently used facets as more salient), but that the topic also plays a role in the vehicle-facet selection. For example, I inferred 23 different facets of rock as underlying the 61 responses given to "My shirt was a rock." The most frequently used facets correspond to the hardness and heaviness of rocks; these facets are involved in 31% and 28% of the responses, respectively.[2] The next most used facets refer to the greyish color (11%) and rough texture (10%) of rocks. Of 26 rock facets inferred to account for the 71 responses to "My sister was a rock" (18 of these also inferred for the shirt item), the "hardness" facet is used most often and is involved in 48% of the responses. The next most used facets refer to the strength (23%), immobility (rocks do not themselves move--17%, and are difficult for people to move--18%), and changelessness (14%) of rocks. Thus, beyond what is likely the most salient facet of rock (i.e., "hardness"), different facets tend to be mapped to shirt than to sister; similar results obtain for the other vehicle obs.

Conclusions

The notion that the topic and vehicle in a metaphor share low-level content facets, which constitute the metaphoric ground, is probably true only for relatively trite metaphors (e.g., those based on immediate topic-vehicle resemblance). The ground is more likely to be facets shared at a higher level (i.e., superfacets), and thus one must propose some process whereby content facets selected from the vehicle are transformed into related facets in the topic. I formalize this process in terms of semantic combinators. The vehicle facets selected for mapping are likely to be ones that have been salient in the subject's construction of the vehicle ob (or that are made salient in some context). However, in interpreting a metaphor one must construe something meaningful about the topic, in light of one's knowledge of the vehicle; thus topic and vehicle interact, in that the selected facets in the vehicle must be transformable into pragmatically important aspects of the topic ob. One can represent metaphor interpretations in terms of vehicle content and the process by which this content is accommodated to the topic.

Notes

*Preparation of this paper was supported by a Social Sciences and Humanities Research Council of Canada postdoctoral fellowship. The research was supported in part by a Natural Sciences and Engineering Research Council of Canada grant to Dr. J. Pascual-Leone, with whom I developed many of the ideas expressed here.

[1]Note that the metaphoric processing levels are ordered because they are embedding and because of their cognitive-developmental difficulties which lead to their ordered emergence in development (see Johnson et al., in press; Johnson & Pascual-Leone, 1984).

[2]Responses can be based on more than one vehicle facet; thus, percentages do not add to 100%.

References

Glucksberg, S., Gildea, P., & Bookin, H. B. (1982). On understanding nonliteral speech: Can people ignore metaphors? Journal of Verbal Learning and Verbal Behavior, 21, 85-98.

Johnson, J., Fabian, V., & Pascual-Leone, J. (in press). Stage-bound mental-capacity constraints on language development: Investigations of subordinate conjunctions and metaphoric processing. In M. Moscato & G. Pieraut-Le Bonniec (Eds.), Ontogenese des processus psycholinguistiques et leur actualisation. Paris: Presses Universitaires de France.

Johnson, J., & Pascual-Leone, J. (1984). Level of processing and mental-attentional demand in metaphor comprehension: Their measurement and developmental validation. Manuscript submitted for publication.

Ortony, A. (1979). Beyond literal similarity. Psychological Review, 86, 161-180.

Pascual-Leone, J., Goodman, D., Ammon, P., & Subelman, I. (1978). Piagetian theory and neo-Piagetian analysis as psychological guides in education. In J. M. Gallagher & J. A. Easley (Eds.), Knowledge and development (Vol. 2). New York: Plenum Press.

Tourangeau, R., & Sternberg, R. J. (1982). Understanding and appreciating metaphors. Cognition, 11, 203-244.

Verbrugge, R. R., & McCarrell, N. S. (1977). Metaphoric comprehension: Studies in reminding and resembling. Cognitive Psychology, 9, 494-533.

The Interaction Between Working Memory and Units of Procedural Knowledge

Thomas A. Kanarski
Donald J. Foss
University of Texas at Austin

This study focuses on the interaction between units of procedural knowledge and working memory. Evidence is provided supporting the concept that procedural knowledge is stored in memory as modular units often referred to as subroutines. The paper also gives evidence supporting the notion that more than one working memory exists in cognitive processing.

To illustrate the idea of units or modules of procedural knowledge, consider a person who must eliminate information from a computer data base system. A reasonable description of the activity will break it into a sequence of chunks (mental or behavioral units) that vary in complexity and duration. For a particular data base management system, called OMNI, one such description is: 1) find the information; 2) mark the information for later elimination; 3) eliminate the marked information. These steps can be thought of as labels. "Find the information" would be an identifier for a group or packet of instructions to get the appropriate data displayed on the CRT. For the purposes of this paper, a packet of instructions is called a unit or module of procedural knowledge.

A module of procedural knowledge may use other units. In the example, the "find the information" module may use a module called "GET" (instructions to use the "GET" command in OMNI). In turn, the "GET" module may use other modules, which use still others, and so forth, until specific motor commands are issued. The more general module, "find the information," deals with a plan of action, the level at which this study focuses.

WORKING MEMORY

An assumption made in this study is that the units are processed in working memory areas that hold instruction groups for processing on a temporary basis (Baddeley and Hitch, 1974). This centralized setup greatly reduces processing overhead (Kanarski, 1984).

Two subroutine retrieval models of working memory are considered in this paper (Sternberg, Monsell, Knoll, and Wright, 1980). For a discussion of how a limited-capacity model (Baddeley and Hitch, 1974) and a competition model (Lashley, 1951; Wickelgren, 1969) of working

memory would interact with units of knowledge see Kanarski (1984). The first subroutine retrieval model (SRM-1) loads modules or subroutines into working memory as needed. The module is processed, then the next module is found and loaded into working memory. SRM-1 has been used to predict the rapid movement sequence of speech and typing at the level of motor commands (Sternberg, et al., 1980). The data also suggest that motor command modules are subject to either rapid decay or destructive reads.

A closer look at the plan-of-action level of processing suggests that the rapid loss of information in working memory may not be efficient. Unlike motor movements, the same module implementing some portion of the overall plan is very likely to be repeated. Using the example above, a person finding several items of information in the data base which are to be deleted may repeat the same command sequence several times to find all of the items. In this case, it would be more efficient to check the contents of working memory and determine if they are needed for the next round of processing (Kanarski, 1984). This type of working memory will be referred to as the subroutine retrieval model - type 2 (SRM-2). The primary difference between the two working memory models is whether information is subject to rapid loss (as in SRM-1) or not (as in SRM-2).

By having a person perform a task for which more than one method exists, it is possible to behaviorally differentiate the two working memory models. The example of eliminating information from a data base using the OMNI data base management system is such a task. The appropriate OMNI commands are "GET" (find the information), "DELETE" (mark the information for later elimination), and "WEED" (eliminate the information). Suppose the user had several items of information to eliminate from the data base. The user could "GET" the location of each item, "DELETE" (mark) each item, and then "WEED" all of the items. This is called a short cycle method and is denoted by GET / DELETE / WEED /. Alternatively, the user could "GET" one item and "DELETE" (mark) it. This is repeated until all of the items are found and marked. Then the user could "WEED" all of the items. This is called a medium cycle method and is represented by GET DELETE / WEED /. Using the last possible method, the user could "GET" one item, "DELETE" (mark) it, and then "WEED" it. This sequence, the long cycle method, is repeated for each item. This method is denoted by GET DELETE WEED /.

Suppose that each OMNI command is represented as a module of procedural knowledge in the user's memory. After processing, a module in the SRM-1 working memory is not available for further processing. If the module is to be used again, it must be found and loaded into working memory. In terms of the OMNI task, the time for the operator to restart a DELETE command in the short cycle method should be the same as the time to start the DELETE command after finishing the GET command in both the medium and long cycle methods.

In a SRM-2 working memory, a check is made to determine if the current module is required for further processing. If it is not needed, the proper module is found and loaded into working memory. If the current module is needed, processing can simply be restarted, bypassing the search and load processes. In terms of the OMNI task, it should be faster to restart the DELETE command in the short cycle method than to start the DELETE command after finishing the GET command in either the medium or long cycle methods. Also, the time to start the DELETE command in both the medium and long cycle methods should be the same. Both methods require that a new module (DELETE) be found and loaded before processing can continue.

METHODS

The subjects were thirty undergraduates at the University of Texas at Austin selected from introductory psychology courses and those responding to a newspaper advertisement. All of the subjects had little or no computer experience and they were all were able to touch type at least 30 WPM. Expert users were not used because they would have specialized task strategies that would confound the study.

The subjects were tested individually. After a typing test, the subject was given a modified version of the OMNI manual to read. The subject was told not to memorize the manual since it would be available during the experiment. The subject then attempted seven practice tasks. These could be accomplished using only one OMNI command.

The subject was then given seven experimental tasks. The tasks fell into one of three types and there were at least two possible methods to accomplish each task type. One task was to eliminate five items from the data base. The methods are described earlier in this paper. Another task was to add five items to the data base and maintain the alphabetical order of the data base. The commands are ADD (add an item to the end of the data base) and ORDER (get the data base in alphabetical order). This task has a short cycle method (ADD / ORDER /) and a long cycle method (ADD ORDER /). The last task was to change information of five items in the data base. The commands are GET (find the item) and CHANGE (change information in the item). This task also has a short cycle method (GET / CHANGE /) and a long cycle method (GET CHANGE /).

The instructions for each task stated how the task was to be accomplished without explicitly stating which commands were to be used. Thus, each subject performed each task type using the short, medium, and long cycle methods. That is, cycle length was the within-subject variable. The experimenter did not give the subject any help unless there was an equipment failure or the subject deviated from the task

instructions. The subjects were not told about either the experimental hypothesis or whether a task was practice or experimental.

The subjects were randomly assigned to one of two groups depending on the order that the tasks were presented. The tasks were interweaved as not to have a task of a particular type follow a task of the same type.

The commands of interest were DELETE, ADD, and CHANGE. Each of these commands occured in only one task type and were used the same number of times in each task. The dependent measure was the mean times before the third, fourth, and fifth use of a command of interest. The first and second times were not used because pilot studies indicated that there were large practice effects influencing the measure. By the third use of a command with a task, none of the subjects referred to either the manual or the task instructions.

RESULTS

Three ANOVAs were performed, one for each command of interest. If the last three times to the next use of a command could not be properly extracted, that subject's data were thrown out for that command only. This occurred once in each analysis.

In each ANOVA, the time to the next use of the command in the short cycle method was significantly less than that time in the long cycle method (see Table 1).

TASK	CYCLE LENGTH	MEAN TIME TO THE NEXT USE OF COMMAND (SECONDS)	F	DF	P
ADD	SHORT	2.65	5.86	1,27	< .05
	LONG	3.34			
CHANGE	SHORT	3.80	12.27	1,27	< .05
	LONG	4.88			
DELETE	SHORT	3.50	6.65	1,54	< .05
	MEDIUM	4.45			
	LONG	4.24			

TABLE 1

The order main effects and the order by cycle length interactions were not significant in any of the analyses. A planned comparison showed that the time to the next command in the short cycle method of the DELETE task was significantly less than both the medium and long cycle methods ($F = 13.70$, $p < .05$). Also, the medium and long cycle method times were not significantly different ($F < 1.0$).

DISCUSSION

The data clearly support the subroutine retrieval model in which the contents of working memory are not subject to rapid loss (SRM-2). The faster times to start commands of interest in the short cycle methods over the long cycle methods indicate that the current contents of working memory are checked. If the match is successful, as it would be in a short cycle method, processing the current contents simply recurs. If the match is not successful, as in the medium and long cycle methods, the proper module must be found and loaded into working memory before processing can continue.

SRM-2 also predicted that there would be no difference in the times to the next use of the DELETE command between the medium and long cycle methods. In both cases, the DELETE module was not in working memory. The same amount of time, on the average, was spent searching for and loading the module in both of the methods.

The assumption that procedural knowledge is packaged into modules simplifies computational theories of cognition. The results of this study were also predicted under this assumption. For a discussion of how non-modular procedural knowledge interacts with working memory see Kanarski (1984).

The data from this study and from Sternberg, et al. (1980) suggest that there exist at least two working memories. One is responsible for processing motor command modules. This working memory acts like SRM-1, that is, there is a rapid loss of information to prevent it from interfering with new information coming into working memory. The other working memory processes information at the plan-of-action level. This working memory retains data for possible continued processing. That there may be two, possibly more, working memories that have different properties is not unreasonable. Considering the tremendous replication of neural structures and localization of function in the brain, many working memory areas, each with specialized processing capbilities, are certainly possible (Rosenzweig and Leiman, 1982).

REFERENCES

Baddeley, A.D. and Hitch, G. (1974). Working memory. In G.H. Bower (Ed.), The psychology of learning and motivation, advances in research and theory, volume 8.

Kanarski, T.A. (1984). Units of knowledge and working memory: Support for a subroutine retrieval model of working memory. Unpublished master's thesis, University of Texas, Austin, Tx.

Lashley, K.S. (1951). The problem of serial order in behavior. In L.A. Jeffress (Ed.), Cerebral mechanisms in behavior.

Sternberg, S., Monsell, S., Knoll, R.L., and Wright, C.E. (1980). The latency and duration of rapid movement sequences: Comparisons of speech and typewriting. In R.A. Cole (Ed), Perception and production of fluent speech. Hillsdale, NJ: Lawrence Erlbaum Associates.

Wickelgren, W.A. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behavior. Psychological Review, 76, 1-15.

Lexical Access Using a Neural Network

Alan H. Kawamoto and James A. Anderson
Department of Psychology
Brown University
Providence, RI 02906

Introduction

To understand language, one must first be able to access items in an internal lexicon and retrieve the semantic properties of the token specified graphemically or phonemically. In recent years, a number of different models of this process have been proposed. These include Morton's logogen model (Morton, 1982), Marslen-Wilson and Tyler's interactive model (Marslen-Wilson and Tyler, 1978) and the McClelland and Rumelhart's interactive activation model (McClelland and Rumelhart, 1981).

One aspect of lexical retrieval that has received a great deal of attention recently is the problem of lexical disambiguation. Despite the fact that almost every common word is a homograph or homophone, we almost always access the appropriate one. Although syntactic, semantic, and pragmatic cues constrain the choice to the appropriate one, all meanings seem to be activated initially. A model of lexical memory must account for these properties. With the interest in natural language parsing by computers, a number of AI researchers have also pursued the problem of lexical disambiguation. Recent work by Hirst (1983) describes one recent approach which considers psychological data in the implementation and provides a review of recent AI attempts to resolve this problem.

The-Brain-State-in-a-Box

Neural network: The model presented here is part of a continuing effort of Anderson and his colleagues (for recent reviews, see Anderson et al., 1977; Anderson, 1983) to simulate aspects of memory and categorization using a network of neuron-like elements. The use of a large number of interacting elements functioning simultaneously reflects the large degree of parallelism found in the nervous system. This overcomes the inherent slowness of the individual components and the noisy operating environment. Although we do not make any claims regarding these elements as realistic manifestations of neurons, we do believe that the major constraints imposed by the nervous system have been taken into account. We assume that (1) nervous system activity can be represented as the simultaneous activity of a group of neurons, (2) activities of single neurons are coded by their firing frequency (above and below steady state levels) and bounded by a maximum and minimum level, (3) memory is distributed rather than localized, with each neuron participating in each memory trace, and (4) synapses associate activity in one element with another by incrementing connection weights by a proportion of the product of values dependent on pre- and post-synaptic activity.

In our system, learning results in modification of the synaptic weights coupling two neurons. The entire set of couplings is given by the matrix A, where an element a_{ij} is the synaptic weight coupling neuron i to neuron j. Unlike previous studies where learning occurred in an unsupervised environment, our current efforts are directed toward systems which learn with a "teacher." To begin a learning trial, a stimulus is chosen from the learning set described in the following section and scaled so none of the elements are saturated. The resulting activity pattern is presented to the network and successively iterated by the scheme

$$x_{t+1} = \text{BOUND}[(A + aI)x_t]$$

where a is a decay constant and
BOUND limits the activity.

The activity after tau iterations, x_{tau}, is compared with the desired output, X, provided by the "teacher." Rather than simply learning a proportion of the outer-product of x_{tau} (the product of each neuron with every other neuron), a proportion of the outer-product of $(X - x_{tau})$ is learned. This error-correcting scheme limits the amount of learning allowed on any given trial and as the current state approaches the desired state, less learning occurs. In fact, if the current state is equal to the desired state, no learning occurs.

Stimulus coding: Although a number of modelling attempts use non-overlapping stimulus representations, each neuron contributes to every stimulus representation. In the simulations below, each lexical entry is 64-dimensional and is formed by concatenating subvectors comprising its graphemic, phonemic, syntactic, and semantic (GPYS) fields. Each field is a 16-dimensional Walsh-Hadamard vector and each distinct value of a given field is represented by a unique Walsh-Hadamard vector. In these initial attempts, the six words shown in table 1 were learned. The 4 hex values are a shorthand notation where each value represents the corresponding Walsh-Hadarmard vector. Thus, as seen in the table, all nouns have identical values in the third field, and likewise for verbs. The only other case with identical values in the same field for more than a single lexical entry is the homograph *wind*. To simulate the different frequencies of occurrance in language, each stimulus is represented a different number of times in the learning set. *Desk* is the most frequent, and *agar* is the least. Furthermore, for the homograph *wind*, the noun will be regarded as the dominant homograph because of its greater frequency relative to the verb.

Simulation Results

In our simulations, we present part of a given lexical entry and allow the output of the network to be fed back until all elements reach saturation. We take the number of iterations required for all elements to saturate as a measure of reaction time (RT). In all cases, the graphemic field is presented with each element in this field fully saturated. In some cases, activity is also present in the syntactic or semantic fields.

Another method probes the semantic field and measures the activity of the current state relative to a number of different meanings. Because all the meanings are mutually orthogonal in the stimulus coding scheme used here, the dot product of the activity in this field with a particular meaning yields a measure of the degree to which that meaning is activated.

Lexical access: Two of the most important observations regarding retrieval from the lexicon are the effects of frequency and hints on RT. Both of these properties can be observed in Table 2. These results show the number of iterations required for the test stimulus to be correctly regenerated after 200, 500, 1000, 2000, and 5000 learning trials, as well as with a "hint."

The frequency effect is manifested in two ways. First, the greater the frequency, the sooner the word is correctly regenerated. We see that *desk* and *rant,* with relative frequencies of 4 and 3, respectively, are learned by the first 200 trials. *Lurk,* with a relative frequency of 2 is learned by 500 trials, and *rant,* with a relative frequency of 1, is not learned until 5000 trials. Furthermore, until RT reaches some asymptotic level (probably as a result of asymptotic learning), the greater the frequency of presentation, the faster the RT.

The second major property of lexical access, the decrease in RT with contextual cues, has also been simulated. To simulate contextual cues, the semantic field of the entry (with the magnitude of each element in the subvector equal to 0.5) is also presented initially. As seen in the last column of table 2, the presence of these cues decreased the RT for every word except *lurk.* In addition, we have found that as the input becomes more degraded (reversing the activity of a number of elements in the graphemic field), the RTs increase.

Lexical ambiguity: In our approach, ambiguous words are treated in the same fashion as all other words. However, use of the dot-product measure described above after each successive iteration allows the time-course of activation of the semantic field to be revealed. As in the properties of lexical access in general, both frequency and context affect which meaning of a homograph is accessed initially. With no context, the more frequent (dominant) homograph's meaning is initially accessed. With the appropriate contextual cue, the less frequent (subordinate) homograph's meaning is also accessed. As in the results of Simpson (1981), we

find that if the appropriate contextual cue is not large enough, the more dominant homograph's meaning is accessed.

However, recent studies reported by Swinney (1982) indicate that *both* meanings are initially activated, independent of context. As seen in figure 1a, with no context, both the dominant and subordinate meanings are activated initially. Even with a contextual cue biasing a particular interpretation, *both* meanings are still activated (see figures 1b and 1c). Even when the syntactic field is specified, again *both* appropriate and inappropriate meanings are initially activated as seen in figure 2.

Summary

In this study, we present a method of learning, storing, and retrieving stimuli constructed as lexical entries. Our formulation allows the different fields comprising a word to interact through coupling weights. It is this property which allows the reconstruction of the entire word from a part of the stimulus. Appropriate hints, implemented as partial activity in fields other than the graphemic one, decrease RTs. In addition, the same scheme used for unambiguous words is used for ambiguous ones. When presented with a homograph, the dominant one is accessed initially. However, when given sufficiently large cues, the subordinate homograph can also be accessed. Moreover, we have been able to show that despite conflicting cues for one of the homographs, *both* meanings are activated initially. We feel that this approach is quite promising and are currently exploring more realistic coding schemes and enlarging the lexicon.

Acknowledgements

Financial support for some of this work was provided by a grant from the National Science Foundation to J. A., administered by the Memory and Cognitive Processes section (Grant BNS-82-14728) and by the United States Office of Naval Research (Contract N00014-81-K-0136) to the Center for Neural Science, Brown University. We would like to thank the Center for Cognitive Sciences, Brown University, for computing facilities used in our simulations.

References

Anderson, J.A., 1983. Cognitive and psychological computation with neural models. I.E.E.E. Transactions on Systems, Man, and Cybernetics SMC-13: 799-814.

Anderson, J.A., J.W. Silverstein, S.A. Ritz, and R.S. Jones, 1977. Distinctive features, categoraical perception, and probability learning: Some applications of a neural model. Psychological Review 84, 413-451.

Hirst, G., 1983. Semantic interpretation against ambiguity. TR CS-83-25, Department of Computer Science, Brown University. Providence, RI.

McClelland, J.L. and D.E. Rumelhart, 1981. An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. Psychological Review 88, 375-397.

Simpson, G.B., 1981. Meaning dominance and semantic context in the processing of lexical ambiguity. Journal of Verbal Learning and Behavior 20: 120-136.

Swinney, D.A., 1982. 'The structure and time-course of information interaction during speech comprehension: Lexical segmentation, access, and interpretation'. in J. Mehler, E.C.T. Walker, and M. Garrett (Eds.), Perspectives on Mental Representation. Hillsdale, N.J.: Erlbaum.

Tyler L. and Marslen-Wilson W.D., 1982. 'Speech comprehension processes'. in J. Mehler, E.C.T. Walker, and M. Garrett (Eds.), Perspectives on Mental Representation. Hillsdale, N.J.: Erlbaum.

LEXICON

lexical entry				*code*	*rel. freq.*
WIND	\wind\	n.;	weather	4285	3
WIND	\wInd\	v.;	rotate	491C	2
DESK	\desk\	n.;	furniture	2D83	4
AGAR	\agar\	n.;	gelatin	E78A	1
RANT	\rant\	v.;	yell	9C1D	3
LURK	\lurk\	v.;	hide	CF1E	2

Table 1. Complete lexicon giving Walsh-Hadamard coding representation and relative frequency.

RTs AS A FUNCTION OF LEARNING

word	200	500	1000	2000	5000	hint*
WIND	85	33	19	17	20	11
DESK	32	15	11	11	11	10
AGAR	xx	xx	xx	xx	23	13
RANT	48	22	13	12	12	10
LURK	xx	77	18	11	11	11

xx error
* after 5000 learning trials

Table 2. RTs as a function of learning (200, 500, 1000, 2000, and 5000 learning trials, and effect of hints).

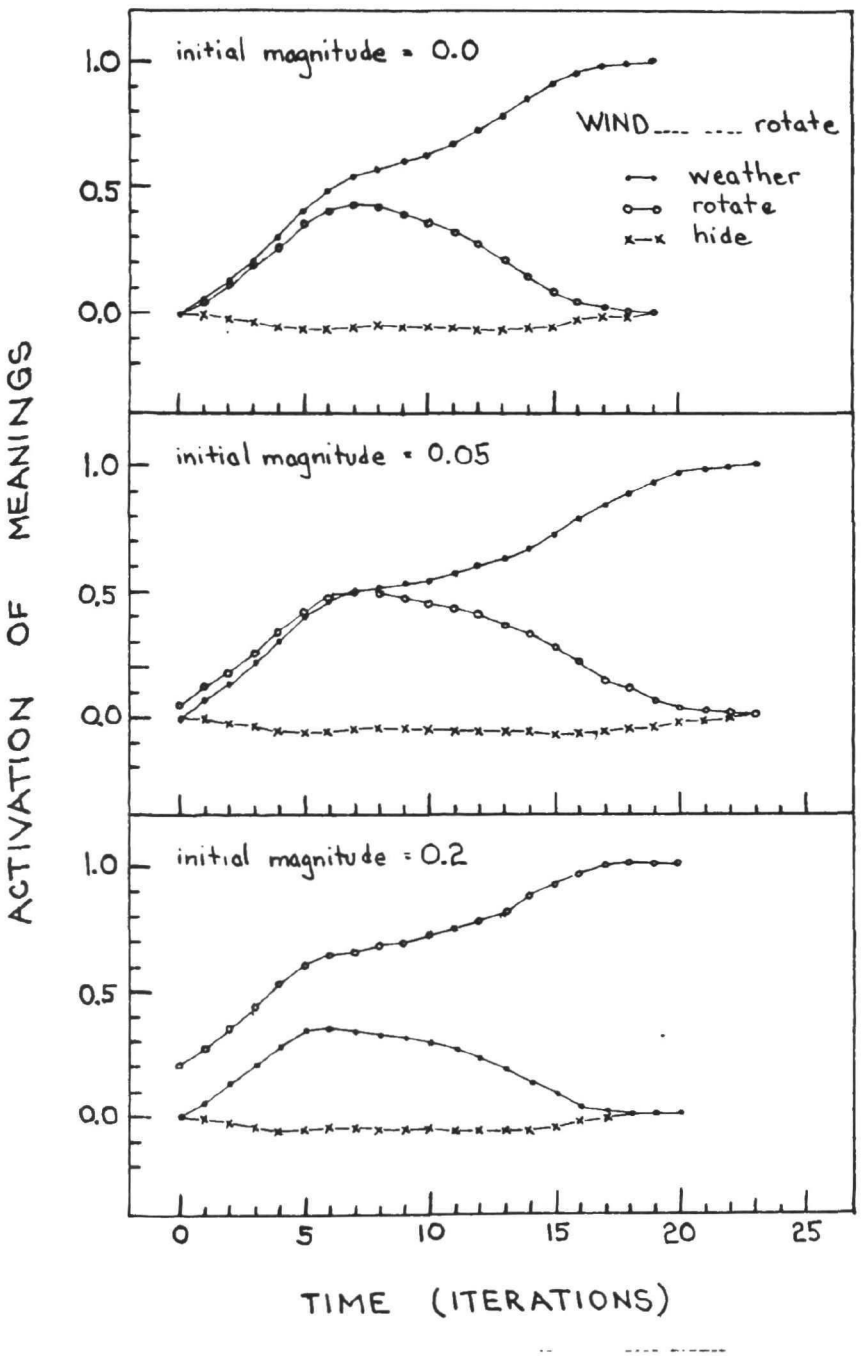


Fig. 1. Time course of activation of meanings with semantic cues. Magnitude of elements in the graphemic field are saturated, and the magnitudes of elements in the semantic field (rotate) are (a) 0.0, (b) 0.05, and (c) 0.2.

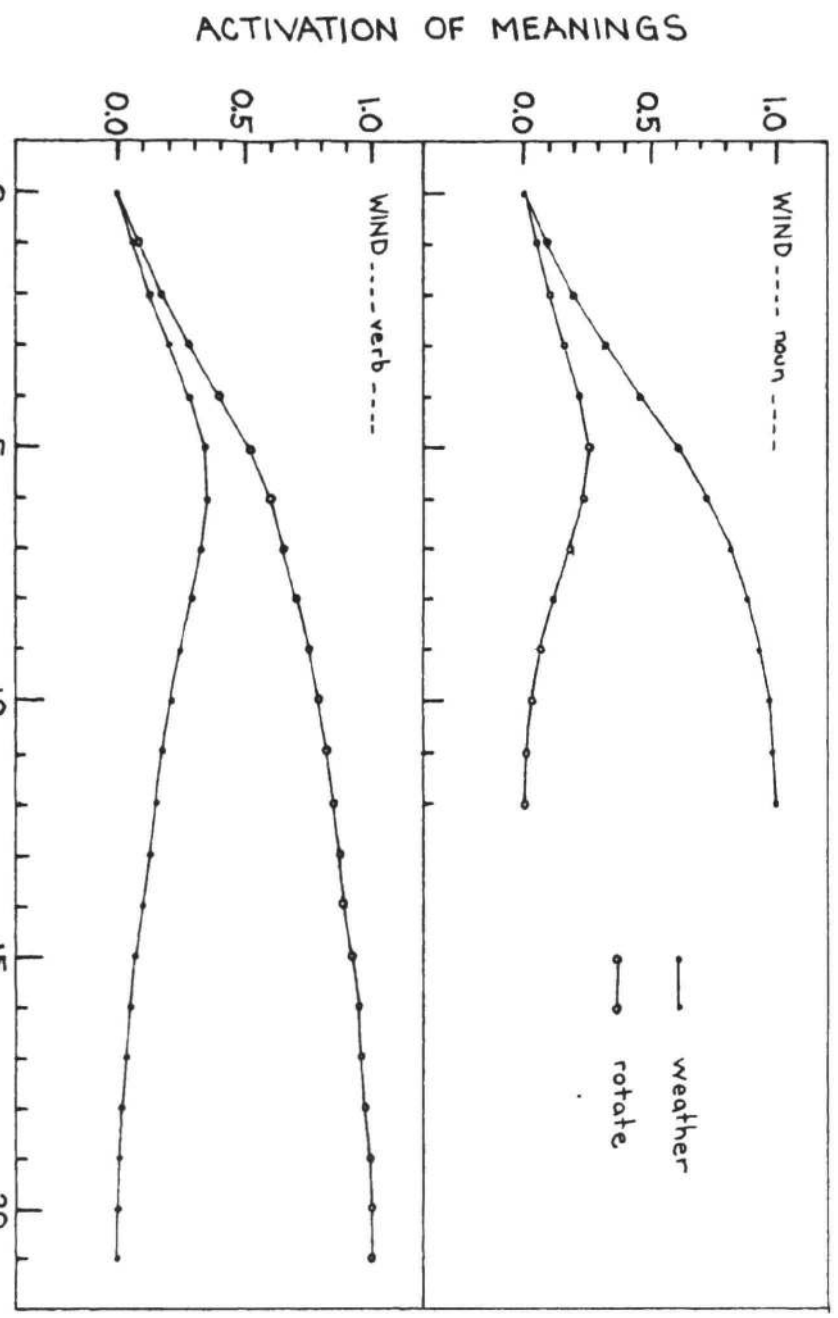


Fig. 2. Time course of activation of meanings with syntactic cues, graphemic field and form class (a) noun, and (b) verb.

Summarising the *Wall Street Journal*

Dana S. Kay and John B. Black

Summarization, whether it is used incidentally in text comprehension or intentionally as a study technique, is an important skill to acquire. Various models have been presented which attempt to describe this complex process of transforming detailed knowledge representations into concise, coherent summaries (Lehnert, 1981, and van Dijk and Kintsch, 1977). The majority of these models are based upon the study of individual summaries of narratives. The models were created by noting differential characteristics across summaries. This research, however, used people with average experience in summarization and thus, only a limited aspect of the summarization process was examined. A problem with this approach is that it does not allow one to study summarization as a cognitive skill with various levels of expertise.

The expert/novice paradigm has recently received a great deal of attention in fields such as Cognitive Psychology, Cognitive Science, and Artificial Intelligence. This paradigm is particularly useful in the study of the development of a given behavior. Brown and Day (1983) examined the differences in the summarization processes used by novices (young children) and experts (graduate students). Using variations on the macrorules of van Dijk and Kintsch (1977), they were able to trace the development of the summarization process from simple deletion of trivial and redundant information to more complex transformational rules of condensation.

The current study extends the development of summarization to the expert level and examines the characteristics of summaries after the acquisition of basic summarization strategies. These characteristics will include both content differences and processing differences that may be accounting for differences in summarization. The domain in which we looked at these characteristics is the summarization of *Wall Street Journal* articles. This domain is optimal because (1) expert summaries appear in the newspaper every day and (2) novices (college undergraduates) understand the content of articles, but lack advanced summarization knowledge.

We used six articles from the *Wall Street Journal*. These articles were chosen because (a) they were summarized on the front page of the paper and (b) they were written in a narrative style rather than the style of a stock report. The stories were about economic concerns such as bankruptcies and marketing changes. Subjects, who served as the novices in our experiment, were asked to give the story a title and write a one to two sentence summary of the story. The expert summaries were taken directly from the *Wall Street Journal* and were roughly the same length.

The titles generated by the novices were used to be sure that the subjects understood the main topic of the article. For all the stories, the titles were similar to the actual article titles. In some cases, the titles stated the main topic, while in other cases, the novices sensationalized the titles to sound like a newspaper article. The latter finding suggests that the subjects were using their knowledge of typical newspaper articles, though the articles were referred to as "stories" in the experiment.

To exemplify the novice/expert differences that were observed, we present a sample expert summary along with a two novice summaries. These summaries are for an article which discussed the possible failure of Osborne Computer and the situation that the company was currently facing.

Expert Summary

Osborne Computer faces possible failure, unless it finds a purchaser. The company is deeply in debt to suppliers, just furloughed nearly 80% of its employees and has halted computer production. One possible buyer, ITT, denies any involvement.

Novice Summaries

Osborne Computer Corp., a portable computer industry, is failing and looking for someone to acquire it and lend it money for debts. The company began as a prosperous, fast-growing company which was ruined when it tried to change the computer market and tangled with the jumbo companies such as IBM.

Osborne Computer Corp., being deeply in debt, faces failure unless it is acquired by a larger company, possibly ITT. Its downfall can be attributed to too rapid a climb, trying too much and pressure from a much larger competitor, IBM.

Before comparing the novice and expert summaries, we analyzed each type of summary and possible strategies that could account for the information that was selected from the article and put in the summary. It should be noted that each story generally presented a main event and several sub-events that either elaborated upon the main event or described past events relevant to the main event. There are three types of information reported in the expert summaries. The first information type is a concise statement of the main event. This statement is usually a condensed version of the first paragraph of the article. The second type of information is an elaboration of the event, with other information present in the article. Using the example above, this information refers to the second sentence in which Osborne's failure is defined by its debt, furlough of employees and halting of computer production. The final type of information present in the data refers to implications of the current situation. That is, statements about the future outcome of the event described. In the example above, this information is the denial of involvement by ITT.

For the novice summaries, only one of the experts' three types of information is present. Novices present the main topic of the story, but do not elaborate the event or note the implications of the event. Instead, they include information about the causes of the situation. Examples of this information can be seen in the second sentences of the novice summaries previously presented. This type of information was found in over one half of the novice summaries for each article. Thus suggesting that novices see causes of an event as more important than elaborations or implications of an event.

Having examined the content of each of the novice and expert summaries, we proposed algorithms that could account for these observed behaviors. It appears that experts decide on the main event of the article and then infer a possible goal that is active for this event. Using this goal, the expert follows the events associated with the goal and notes the success or failure of these events. That is, the expert seems to look at the future results of the situation.

The novice summarization process begins in the same manner as the expert process in that the current goal is inferred. However, rather than carrying the goal through to the possible implications, novices attempt to explain the goal by reporting other events that are causally linked to the active goal and present these events.

These content differences suggest that there are novice/expert distinctions in summarization even after they have acquired the basic structural strategies such as those presented in Brown and Day. These distinctions appear in the different selection and abstraction strategies used. When

selecting what to report in a summary, novices focus on the causes of the main topic, whereas, experts focus their attention on the possible outcomes of the event. In addition, experts present a more detailed representation of the event, rather than presenting a number of causally related events at less detailed levels.

The question which must now be answered is why does this difference occur? The explanation that we would like to give for the observed differences is that novices are viewing summarization as a process by which one attempts to put as much information as possible from the article into the constraints of the summary. As a result, they present less detailed versions of as many events as possible. On the other hand, experts see summaries as brief, coherent presentations of the most important event in the passage. Therefore, they present a more detailed account of the event and the implications of that event that might later be of importance.

However, there are other possible explanations for these results which should be noted. One explanation is that the experts in our experiment have had experience with the individual events and assume that the readers of their summaries have also been following the story. Thus, they do not see the past causally related events as important, but rather try to predict future events that the reader should be aware of. We are currently testing this explanation by giving novices a set of related articles that describe the entire progression of the event and asking for summaries of each article in the set. If the above explanation is true, then the summaries of the first articles in the set should be similar to the novice summaries above and the final summaries should be similar to the expert summaries.

Another explanation is that the novices and experts in our study had different goals. That is, the experts were trying to get someone to read the full article presented later in the paper and the novices were simply complying with our instructions and presenting a basic summary. This explanation, if true, suggests that past models of summary are incomplete in that the goals of the writer of the summary have not been considered. We plan to test this hypothesis by telling the novices that they are writing a summary for the Wall Street Journal and giving them an example of a summary written for the paper. If the goal explanation is correct, then the novice summaries should be more similar to the expert summaries.

This study was proposed as an exploratory experiment. From the data, we were able to generate an number of hypotheses about summarization and content differences that are present after the acquisition of general summarization rules. At this point, we are pursuing several of these hypotheses. In addition, we use verbal protocols of the summarization process to get a better picture of the process. The results that we have observed thus far suggest that summarization is more dynamic than previous models have suggested and needs further exploration of the goals in summarization as well as the plans used to achieve these goals.

Acknowledgements: This research was supported by a grant from the Systems Development Foundation.

References

Brown, A.L. & Day, J.D. (1983) Macrorules for summarizing texts: The development of expertise. *Journal of Verbal Learning and Verbal Behavior*, **22**, 1-14.

Lehnert, W.G. (1981) Plot units and narrative summarization. *Cognitive Science*, **5**, 293-331.

van Dijk, T.A. & Kintsch, W. (1977) Cognitive psychology and discourse: Recalling and summarizing stories. In W.U. Dressler (Ed.), *Trends in text-linguistics*. New York: De Gruyter.

THE ACQUISITION OF PROCEDURES FROM TEXT

David E. Kieras and Susan Bovair

Department of Psychology
University of Arizona, Tucson, AZ 85721

Quite often people must learn procedures from written instructions. In the context of the currently developing theory of procedural knowledge and cognitive skill, this task must involve the formation of production rules from the information available in text. This process has not been systematically explored; the results reported here provide an initial characterization. Two main conclusions will be presented. The first is that using a production rule representation can provide a very precise characterization of the difficulty of learning procedures. The second is that apparently there are powerful comprehension-like processes that operate very early in learning on declarative representations of production rules. This supplements Anderson's (1982) description of the acquisition of skill, in that much of the work of learning a procedure can take place before a procedural representation has been formed.

Our approach is to have subjects learn procedures for operating a simple piece of equipment by reading step-by-step instructions. By measuring the reading time, and the accuracy of execution of the procedure, we are able to essentially track the acquisition of individual rules. The procedures are related, so some transfer of training is possible. A major result is that this transfer can be predicted very well based on the production system representation for the procedures. This paper is highly condensed; full details can be found in Kieras and Bovair (in preparation).

METHOD

The subjects learned how to operate a device consisting of a simple control panel, which is described in Kieras and Bovair (in press, in preparation). The goal of operating the device was to get a certain indicator light to flash. Each procedure consisted of several steps, illustrated in Tables 1 and 2. Table 1 is the procedure for a "normal" situation, in which the device is operating properly. Table 2 is the procedure for a "malfunction" situation. The device could be in one of several malfunction states, in which some imaginary component of the device was not operating. Depending on the nature of the malfunction, the device either could be made to work by an alternate procedure, or could not. The final step in each procedure was to signal success or failure in getting the device to work. Note that this was a rote learning situation; the internal organization of the device was not taught to the subjects. Each subject learned a series of 10 such procedures in a fixed order. There were three different orders, chosen as described below, with a separate group of 20 subjects for each order.

Table 1
An Example of a "Normal" Procedure

If the command is to do the MA procedure, then do the following:

Step 1. Turn the SP switch to ON.

Step 2. Set the ES selector to MA.

Step 3. Press the FM button, and then release it.

Step 4. If the PF indicator flashes,
then notice that the operation is successful.
Step 5. When the PF indicator stops flashing, set the ES selector to N.

Step 6. Turn the SP switch to OFF.

Step 7. If the operation was successful,
then type "S" for success.
Step 8. Procedure is finished.

Table 2
An Example of a "Malfunction" Procedure

If the command is to do the MA procedure, then do the following:

Step 1. Turn the SP switch to ON.

Step 2. Set the ES selector to MA.

Step 3. Press the FM button, and then release it.

Step 4. If the PF indicator does not flash,
then notice that there is a malfunction.
Step 5. If the EB indicator is on, and the MA indicator is off,
then notice that the malfunction might be compensated for.
Step 6. Set the ES selector to SA.

Step 7. Press the FS button, and then release it.

Step 8. If the PF indicator does not flash,
then notice that the malfunction can not be compensated for.
Step 9. Set the ES selector to N.

Step 10. Turn the SP switch to OFF.

Step 11. If the malfunction could not be compensated for,
then type "N" for not compensated.
Step 12. Procedure is finished.

To learn each procedure, the subjects first read a set of step-by-step instructions for the procedure, such as those in Tables 1 and 2, and then attempted to execute the procedure on the device. If they made an error, they were immediately informed, and then began to read the instructions again. They were required to execute the procedure correctly three times in a row before proceding to the next procedure. The data recorded were the reading time on each step of the instructions, the accuracy of each step while executing the procedure, and the speed and accuracy of a final retention test, which will not be discussed here.

THEORETICAL ANALYSIS

The step-by-step instructions exemplified in Tables 1 and 2 were prepared so that each sentence in the instructions appeared to correspond to a single production rule, one for each step or action (internal or external) involved in the procedure. Each procedure could then be expressed as a series of production rules. Table 3 provides an example corresponding to Table 1. The syntax of these rules is very simple and will not be discussed here. See Kieras and Polson (in press) for a full description of the production system notation, along with a description of the user-device interaction simulation that was used to test the production rules for accuracy. The system's working memory contains descriptions of either GOALS, or NOTES, which consist of non-goal items concerning processes underway, the environment, or specifications of the tasks to be accomplished.

In earlier work with this device (see Kieras and Bovair, 1983) it was noticed that the time required to learn the procedures under rote conditions varied over a very wide range. Obvious variables like the number of steps in the procedure, or serial order, could not explain this variation. Rather, the explanation appeared to lie in the order in which the procedures were learned, and the relation between the steps in a new procedure and those that the subjects had already learned. A transfer process was defined to explain how this transfer of training would work in terms of production rules, and formalized as a LISP program.

The transfer process compares the production rules for a new procedure with the production rules for all the procedures that have already been learned. Each rule in the new procedure can then be placed in one of three categories: The rule is identical to a previously learned rule, or the rule is completely new compared to the already learned rules, or it is generalizable with an old rule. The generalizable category requires some explanation: If the rule from a new procedure is similar to an already known rule, differing in only one term in the description of a goal or note in working memory, then this term in the old rule can be replaced with a "wild card," which matches any term in memory, and the rule from the new procedure can be discarded.

Table 3
Example of Production Rules

```
(MA-N-START
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (NOT (TEST-GOAL DO ??? STEP)))
THEN ((ADD-GOAL DO SP-ON STEP)) )

(MA-N-SP-ON
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO SP-ON STEP))
THEN ((OPERATE-CONTROL *SP ON)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO SP-ON STEP)
      (ADD-GOAL DO ES-SELECT STEP)) )

(MA-N-ES-SELECT
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO ES-SELECT STEP))
THEN ((OPERATE-CONTROL *ESS MA)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO ES-SELECT STEP)
      (ADD-GOAL DO FM-PUSH STEP)) )

(MA-N-FM-PUSH
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO FM-PUSH STEP))
THEN ((OPERATE-CONTROL *FM PUSH)
      (WAIT-FOR-DEVICE)
      (OPERATE-CONTROL *FM RELEASED)
      (DELETE-GOAL DO FM-PUSH STEP)
      (ADD-GOAL DO PFI-CHECK STEP)) )

(MA-N-PFI-CHECK
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO PFI-CHECK STEP)
        (LOOK *PFI FLASHING))
THEN ((ADD-NOTE OPERATION SUCCESSFUL)
      (DELETE-GOAL DO PFI-CHECK STEP)
      (ADD-GOAL DO ES-N STEP)) )

(MA-N-ES-N
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO ES-N STEP)
        (LOOK *PFI OFF))
THEN ((OPERATE-CONTROL *ESS N)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO ES-N STEP)
      (ADD-GOAL DO SP-OFF STEP)) )

(MA-N-SP-OFF
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO SP-OFF STEP))
THEN ((OPERATE-CONTROL *SP OFF)
      (WAIT-FOR-DEVICE)
      (DELETE-GOAL DO SP-OFF STEP)
      (ADD-GOAL DO TAP STEP)) )

(MA-N-TAP
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO TAP STEP)
        (TEST-NOTE OPERATION SUCCESSFUL))
THEN ((DELETE-NOTE OPERATION SUCCESSFUL)
      (ADD-NOTE TYPE S-FOR SUCCESS)
      (DELETE-GOAL DO TAP STEP)
      (ADD-GOAL DO FINISH STEP)) )

(MA-N-FINISHED
IF (AND (TEST-GOAL DO MA PROCEDURE)
        (TEST-GOAL DO FINISH STEP)
        (TEST-NOTE TYPE S-FOR SUCCESS))
THEN ((DELETE-NOTE TYPE S-FOR SUCCESS)
      (DELETE-GOAL DO FINISH STEP)
      (DELETE-GOAL DO MA PROCEDURE)) )
```

The assumption is that the only rules that require substantial effort to learn are the completely new ones; the identical and generalizable rules should be very easy to learn, since all or almost all of their content is already known. Thus, the number of new rules in a procedure should be closely related to the difficulty of learning the procedure. In the data reported in the rote condition in Kieras and Bovair (1983) the number of new rules in a procedure accounts for 79% of the variance among the mean training times for the 10 procedures, supporting the value of the production system analysis of transfer in the learning of procedures.

By using three different training orders, this study was designed to get a more comprehensive set of data on the relation of the production rule representation to transfer of training. The three different training orders were chosen by analyzing the production rule sets for each procedure using the transfer process program, and selecting training orders that produced substantial variation in the number of new rules in each procedure, and also the number of new rules in each serial position in the training order.

RESULTS

Training Time

The total training time for a procedure is defined as starting when a subject begins the first reading of the first sentence of the instructions, until the last step of the last attempted execution. The training times for each subject on each procedure in the three training order conditions (a total of 600 observations) were analyzed with multiple regression in terms of the transfer status of the rules in each procedure. Figure 1 shows the predicted and observed mean times and the final regression equation. The most important predictor variable was the number of new rules in each procedure (NEW), which alone could account for 69% of the variance. The partial regression coefficient for NEW is substantially larger than those for identical (OLD) rules and generalizable rules (GEN), which were very similar. In addition, there were other effects, notably some learning-to-learn effects (FIRST and ORDER), and an apparent "overload" effect (C2FIRST), in which the first procedure in the second training order condition was very complex and took an extremely long amount of time to learn. Details appear in Kieras and Bovair (in preparation). Despite these other effects, however, the production system variables provided by the transfer process explain the training times very well; in fact, the number of new rules is a better predictor than the subjects' individual means!

Thus, by analyzing the procedures in terms of production rules, and the relations between them, it is possible to account for the difficulty of the learning the procedures with great precision.

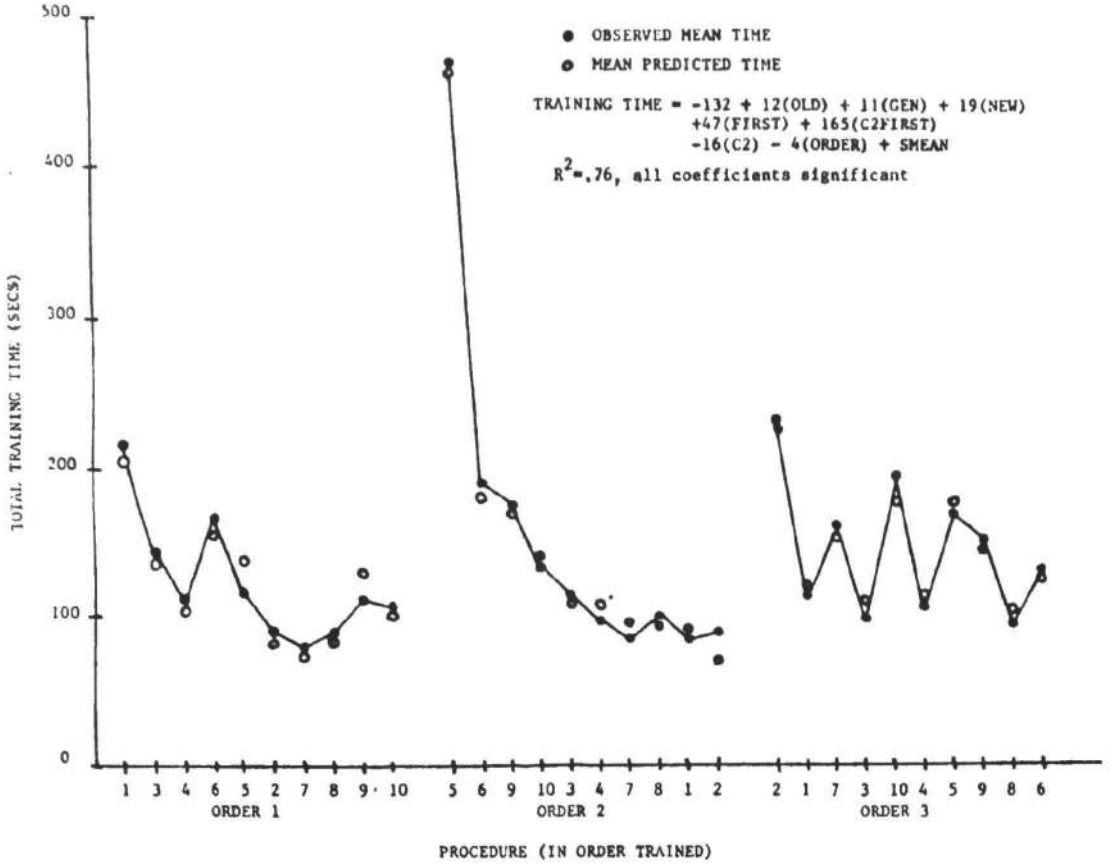


Figure 1. Predicted and observed mean training times.

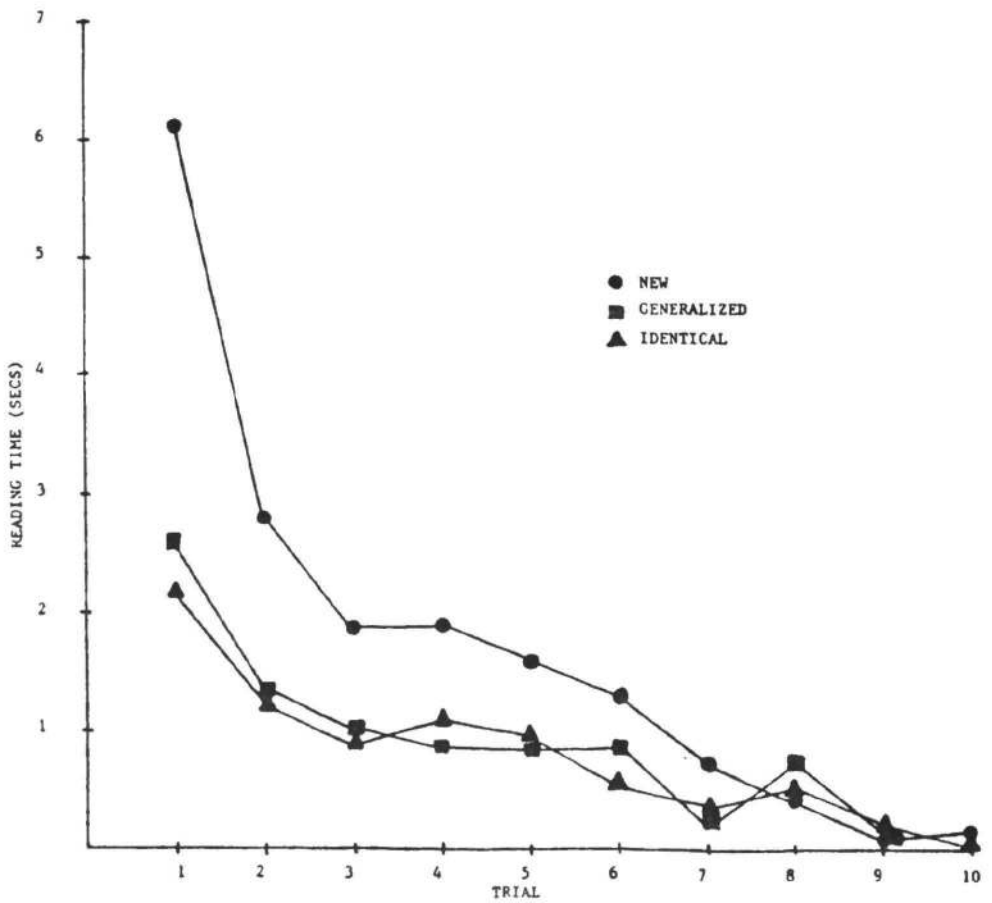


Figure 2. Sentence reading times.

Reading Time

The time required to read each sentence of the instructions, was averaged over procedures, but classified by training trial (e.g. first reading, second reading, etc.) and by the transfer status of the corresponding production rules. Figure 2 shows these means. The key point is simple. There was a substantial difference in the reading times for instruction steps depending on the transfer status of the corresponding production rule. The reading times for generalizable and identical rules were almost identical, but reading times for new rules were much longer. A key result is that this difference appears on the first reading, meaning that subjects can immediately distinguish whether a sentence describing a step corresponds to a new rule or an old one, and govern their reading and study time accordingly.

CONCLUSIONS

A basic conclusion is that production rules, as a way to represent procedural knowledge, can provide a detailed account of learning. This supports the approach suggested by Kieras and Polson (in press) who suggest that the production-rule theory of skill acquisition is useful for practical applications. That there are other phenomena involved, such as the "overload" described above, is clarified by the production system analysis as well.

These results present a puzzle for the theory of skill acquisition as formulated by Anderson (1982). The transfer process defined here has many similarities to some of Anderson's compilation and tuning processes. However, his processes are defined in terms of operations on procedural representations. These are constructed as a by-product of the activity of general interpretive procedures that are driven by an initial declarative encoding. However, in these results, rules are being compared, modified, and constructed very rapidly, and apparently before they exist in a procedural form. As Figure 2 shows, a generalization process can apparently occur on the first reading, and is almost as fast as recognizing an identical rule. Although there is no rigorous basis at this time for saying so, it seems that these aspects of the results are not reasonably subsumed under Anderson's compilation and tuning processes.

Instead, perhaps the work of relating new and old rules is done by processes similar to those proposed for macroprocessing in comprehension (e.g. Kieras, 1982), which can compare, modify, and construct complex propositional representations while reading is going on. Thus, subjects translate the instruction sentence into a declarative representation of a complete production rule, which can then be related to other such representations. As in Anderson's proposals, this declarative representation would be interpreted by a general procedure for following instructions, and the procedural form of the rules would eventually be formed. However, correct execution of the procedure would begin when the declarative rule set has been successfully encoded, and the time

required to do so would depend on how much use could be made of previously learned rule representations. Thus, when procedures are acquired from text, comprehension-like processes can play a major role early in learning, leaving the compilation and tuning processes to govern learning once the initial declarative form of the rules is in place.

ACKNOWLEDGEMENT

This research was supported by the Office of Naval Research Personnel and Training Research Group, under Contract Number N00014-81-C-0699, Contract Authority Identification Number NR 667-453.

REFERENCES

Anderson, J. R. (1982). Acquisition of cognitive skill. Psychological Review, 89, 369-406.

Kieras, D. E. (1982). A model of reader strategy for abstracting main ideas from simple technical prose. Text, 2, 47-82.

Kieras, D. E., & Bovair, S. (1983). The role of a mental model in learning to operate a device (Technical Report No. 13 UARZ/DP/TR-83/ONR-13). University of Arizona, Department of Psychology.

Kieras, D. E., & Bovair, S. (in press). The role of a mental model in learning to operate a device. Cognitive Science.

Kieras, D. E., & Bovair, S. (in preparation). The acquisition of procedures from text (Technical Report). University of Arizona, Department of Psychology.

Kieras, D. E., & Polson, P. G. (In press). An approach to the formal analysis of user complexity. International Journal of Man-Machine Studies.

Pre-schooler's solution of problems with ambiguous sub-goals

David Klahr

Carnegie-Mellon University

27 February 1984

Abstract

Children (4 to 6 years old) were presented with problems requiring from 4 to 7 moves for solution. The problems were constructed such that the subgoal ordering was ambiguous. Children's performance was consistent with a generate and test strategy that had a 2-move lookahead for a goal state, a no-backup constraint, and some partial evaluation of progress toward the goal.

When faced with problems in unfamiliar domains, adults draw on a small repertoire of processes called "weak methods", including generate-and-test, heuristic search, means-ends analysis, hill-climbing, and planning (Newell, 1969; Laird & Newell, 1983a). The weak methods are usually inadequate and inefficient compared to knowledge-rich, problem-specific methods. Nevertheless, they are extremely general, and they often provide the only basis for intelligent action. Young children also use rudimentary forms of weak methods requiring the use of subgoals, such as means-ends analysis (Klahr, 1978; Klahr & Robinson, 1981, Spitz & Borys, 1984; Spitz, Webster, & Borys, 1982).

Klahr & Robinson (1981) presented pre-school children with two variants of the Tower of Hanoi (TOH). They found that performance declined when subgoal ordering was not self-evident. On the standard "tower-ending" problems, in which all the objects are stacked on a single peg, it is clear that the bottom-most object must get to the goal peg first, then the second from the bottom, and so on. This subgoal sequence is apparent even though the exact move sequence necessary to achieve it is not. On these problems, half of the 6-year-old subjects could solve 6-move problems, and even 5-year olds were able to solve 4-move problems most of the time.

On "flat-ending" problems, in which each peg has one object on it, the proportion of 5- and 6-year-olds who could reliably plan at least four moves ahead dropped from 81% to 40%. Flat-ending problems do not have an obvious order in which disks reach their goal pegs. When the surface form of the problem does not suggest an unambiguous ordering of subgoals, then children have a difficult time applying MEA. Instead, they must use an even weaker one of the weak methods.

This study further investigates how pre-school children behave when confronted with such ambiguous subgoal problems. We address the following questions:

* Do children move haphazardly when subgoals are ambiguous?

* Do children avoid unnecessary backup?

* Do children advance directly toward a goal once it becomes "visible"?

* Are children reluctant to move away from a goal temporarily in order to ultimately reach it?

* Are they easily led down "garden paths"?

The Dog-Cat-Mouse Puzzle

The Dog-Cat-Mouse (DCM) puzzle consists of three toy animals and three toy foods that "belong" to the animals (a bone, a fish and a piece of cheese) arranged on the game-board illustrated in Figure 1. The board has four grooves running parallel to each side of the square, and a diagonal groove between the upper left and lower right corners of the square formed by the four outside grooves. The animals can be moved along the groves, and the foods can be fastened to and unfastened from each of the four corners.

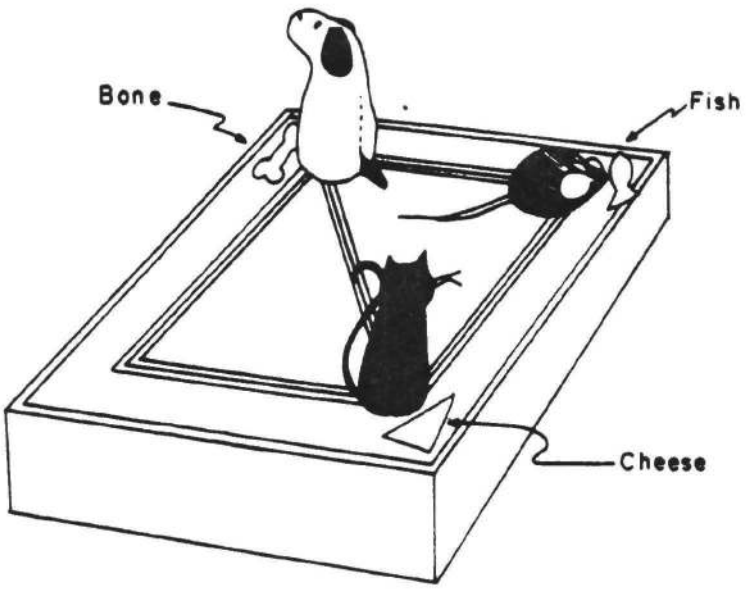


Figure 1 The apparatus for the Dog-Cat-Mouse problem. Each animal must be moved to its favorite food: the dog to the bone, the cat to the fish, and the mouse to the cheese.

A problem consists of an initial state -- indicated by some arrangement of the animals and a final state -- indicated by some arrangement of the foods.

Problem Set

The state space for the DCM puzzle is illustrated in Figure 2[1]. Each node represents a legal configuration. The label on each arc corresponds to the animal that was moved to get from one state to its neighbor.

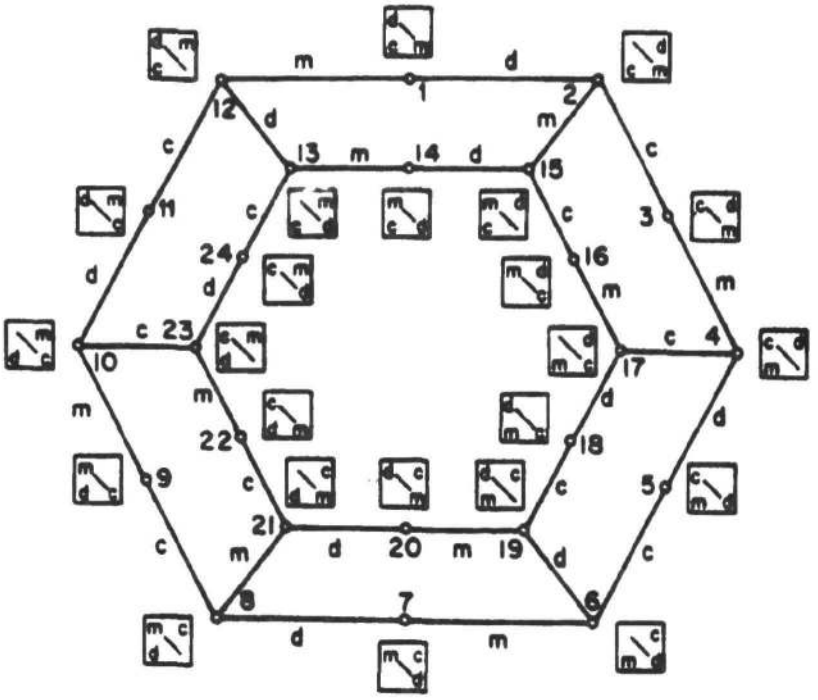


Figure 2 State space for the DCM problem. Each node represents a unique configuration of the three animals.

Several properties of the state space are relevant to our subsequent discussion:

* Rotation problems have both initial and final states on the same hexagon -- either the inner or the outer. They have minimum paths that do not use the diagonal of the game board (Examples: 1-5, 23-17). Permutation problems have initial and final states on different hexagons, and require the use of the diagonal. These problems start and end with different permutations of the three animals, and the permutation order can be changed only by using the diagonal (Examples 1-15, 22-3).

* Permutation problems generally have several minimum paths. For example, the minimum path from node 1 to node 19 could cross from the outer to the inner loop at nodes 2, 4 or 6.

* If we abstract over the specific identity of the pieces, then there are only two types of nodes: those with open diagonals, having three adjacent states (e.g., 2, 4, 19, 21), and those with closed diagonals, having two adjacent states (e.g., 1, 3, 18, 20). At an open node, there are three possible moves; at a closed node, there are two possible moves.

Problem selection

Problems were designed to vary path length (from 4 to 7), type of initial node (open or closed diagonal), and problem type (permutation or rotation). The eight problems selected are listed in the bottom section of Table 1.[2] In addition, four three-move training problems were used. They are shown at the top of Table 1.

Table 1: Problem Set

		Initial State	Goal State	Path Length	Initial Node	Problem Type
Training Set	T1	1	4	3	closed	rotation
	T2	7	22	3	closed	permutation
	T3	12	9	3	open	rotation
	T4	2	17	3	open	permutation
Problem Set	1	17	21	4	open	rotation
	2	18	8	4	closed	permutation
	3	11	20	5	closed	permutation
	4	10	5	5	open	rotation
	5	13	19	6	open	rotation
	6	24	18	6	closed	rotation
	7	14	7	7	closed	permutation
	8	15	8	7	open	permutation

Method

Subjects

Thirty-nine predominantly middle-class children, ranging in age from 45 to 70 months old, completed this experiment.

Procedure

Problems were presented in the order shown in Table 1. Children were told a cover story about animals who wanted to get to their favorite food. Children were given two chances to produce a minimum path solution to each problem. If a problem was solved in the minimum number of moves, then the next problem in the sequence was presented. If it was solved in more than the minimum number, or if it had not been solved after twice the minimum number of moves had been made, then it was presented a second time. Regardless of whether the second trial produced the minimum path, a longer solution path, or no solution, the next problem in the sequence was then presented.

Results

For each trial, subjects were assigned a 0/1 score based on the number of moves they made. If the number of moves made on any trial was greater than two more than the minimum path, then the score was 0, otherwise it was 1. Note that the scoring is somewhat more lenient than the actual procedure used to decide whether or not to repeat a trial. Two extra moves were allowed because they correspond to common minor errors in this particular state space:

* On rotation problems, if subjects unnecessarily use the diagonal to move from the inner to outer hexagon, and then move back to the correct hexagon, they will make two extra moves.

* If subjects make a false start, but immediately correct it, then they will make two extra moves.

Each subject was assigned a score based on the percentage of passes (1s) -- on either the first or second presentation of each problem -- across the eight problems. Each problem was assigned a score based on the proportion of subjects passing it. The scores, ranked by subject performance and problem difficulty, are shown in Table 2.

Subjects' performance varied widely: from maximum scores for the three best subjects to almost total failure for the worst subject. Problem difficulty also varied widely: from nearly all subjects passing the easiest problem to about two-thirds of the subjects failing the hardest problems.

The most important result shown in Table 2 is the rank order of the problems. Recall that the problems varied in path length from 4 moves (problems 1 and 2) to 7 moves (problems 7 and 8). Path length is a poor predictor of problem difficulty (r = .34). (See also the solid line in Figure 3.) The two easiest problems are also the two shortest, but even though they both have a path length of 4, there is a 20% difference in the proportion of subjects passing them. The next two easiest problems are the two longest (7 moves). The four hardest problems are intermediate in path length, and within that set, there is a large difference between the pairs with the same path length.

In the following analysis, we will show how path length, solution strategy and the structure of the problem space interact to produce this pattern of results.

Strategic analysis

How might children attempt to solve these problems? In this section we will describe a basic strategy and compare it to the subjects' performance. Then we will propose several variations on that strategy, and show that none of them fit the data as well as the basic strategy.

Consider the following procedure -- called Strat2 -- for making moves in the

Table 2: Pass/fail scores (by second trial) for all subjects on all problems. Ranked from best to worst subject and easiest to hardest problem.

Problem Number	2	1	8	7	3	5	6	4	
Subject									mean
4	1	1	1	1	1	1	1	1	1.0
12	1	1	1	1	1	1	1	1	1.0
26	1	1	1	1	1	1	1	1	1.0
18	1	1	1	1	1	1	0	1	.88
20	1	1	1	1	1	1	1	0	.88
17	1	1	1	1	1	1	*	0	.85
5	1	1	1	1	1	0	1	0	.75
11	1	1	1	1	1	1	0	0	.75
14	1	1	1	1	0	1	1	0	.75
33	1	1	1	1	0	1	1	0	.75
3	1	1	1	1	0	1	0	1	.75
36	1	1	1	1	0	1	0	1	.75
27	1	1	1	1	1	0	0	1	.75
2	1	1	1	0	1	0	1	1	.75
16	1	1	1	0	1	1	0	1	.75
7	1	1	1	1	0	1	0	0	.63
39	1	0	1	1	0	1	1	0	.63
56	1	0	1	1	1	0	1	0	.63
41	1	0	1	1	1	0	0	1	.63
51	1	1	0	0	1	0	1	1	.63
1	0	0	1	1	1	0	0	1	.50
13	1	1	1	1	0	0	0	0	.50
19	1	1	1	1	0	0	0	0	.50
50	1	0	1	0	1	1	0	0	.50
9	1	1	1	0	0	0	1	0	.50
29	1	1	1	0	1	0	0	0	.50
6	1	1	0	0	0	1	0	1	.50
30	1	1	0	0	1	*	0	0	.43
15	0	0	1	1	0	1	0	0	.38
25	1	0	0	1	0	1	0	0	.38
31	1	0	0	1	0	1	0	0	.38
43	1	1	1	0	0	0	0	0	.38
45	1	1	0	0	0	0	1	0	.38
8	1	1	0	0	1	0	0	0	.38
40	1	1	0	0	1	0	0	0	.38
10	1	1	0	0	0	0	0	0	.25
55	1	1	0	0	0	0	0	0	.25
24	1	0	0	0	0	0	0	0	.13
42	1	0	0	0	0	0	0	0	.13
Problem mean	.95	.74	.69	.59	.51	.50	.34	.33	

DCM state space:

1. If there is a two-move sequence that can reach the goal state, then make it, otherwise:

2. Generate all candidate moves: (all legal moves, except the piece just moved.)

3. If there is more than one candidate, choose randomly.

4. Go to step 1.

This is a simple generate-and-test strategy, with two constraints: a) Two-move lookahead to the goal state. The lookahead has a very simple evaluation function: the state is either the goal state or it is not. No partial evaluations are made (such as the number of pieces in their goal positions.) b) No immediate backup. In the DCM puzzle, a constraint against moving the same piece twice is equivalent to a prohibition on immediate backup.

We can determine the probability that Strat2 would discover a minimum path solution for each problem by computing the compound probabilities that it will stay on a minimum path.

By applying this analysis to each of the eight problems, we can compare the probability that Strat2 would pass each problem with the subjects' actual performance. For each of the problems in Table 3 -- ranked from easiest to hardest -- we have listed the problem number, the initial and final states, the path length, the probability that Strat2 would find the minimum path solution on a single trial, the probability that Strat2 would be successful if it were given two chances to find the minimum path, and, in the final column, the proportion of subjects passing each problem by the second trial.

Table 3: Subject performance and Model performance

Problem Number	States Initial	Final	Path Length	Strat2 p	Strat2 $2p - p^2$	Proportion Passing
2	18 --->	8	4	.500	.750	.95
1	17 --->	21	4	.333	.556	.74
8	15 --->	8	7	.500	.750	.69
7	14 --->	7	7	.500	.750	.59
3	11 --->	20	5	.375	.609	.51
5	13 --->	19	6	.333	.556	50
6	24 --->	18	6	.250	.440	.34
4	10 --->	5	5	.167	.310	.33

A plot of both the model's and the subjects' likelihood of success for each problem, is shown in Figure 3. Strat2 explains almost 60% of the variance in problem difficulty (r = .767, t = 2.9, df = 7, $p < .05$). If we eliminate the two 4-move problems, which were much easier for the subjects than for the model, then the correlation between Strat2 and subject performance is r = .95 (t=6.02, df=5, $p < .01$).

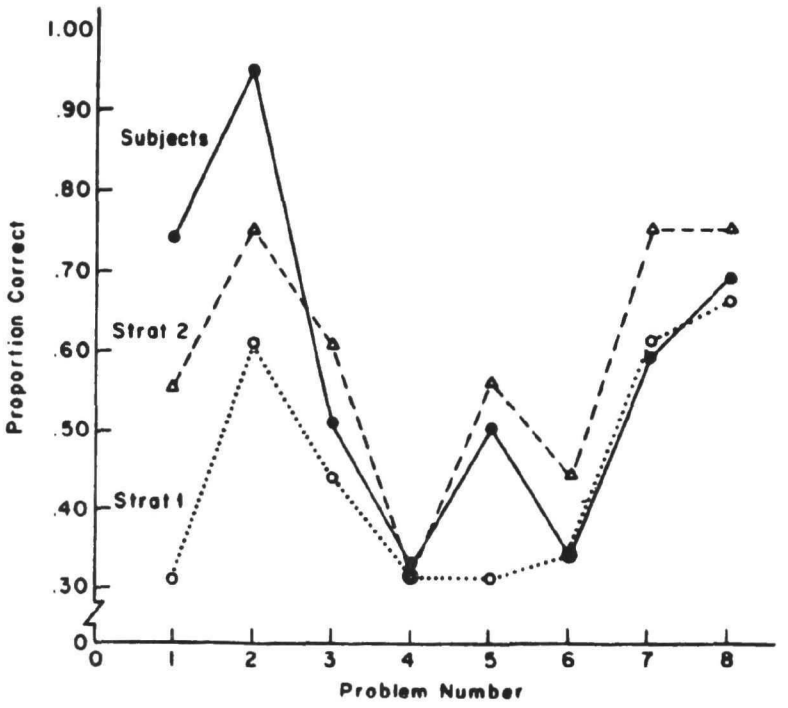


Figure 3 Probability of passing a problem by the second trial for subjects (solid line), Strat2 (dashed line) and Strat1 (dotted line).

Strat2 can be characterized as a random walk through the state space with two constraints: no immediate backup, and a two-step lookahead for the goal state. We can ask two questions about these constraints. First, how well do subjects adhere to them? Second, how important are they?

The No-Backup Constraint. Children's compliance with the no-backup constraint was assessed by counting the number of times - over both successful and unsuccessful trials - that they moved the same piece twice in succession. Overall, there was a violation rate of 11%. If moves were made without the constraint, we would expect 33% of moves to be double moves.

Removing the no-backup constraint from Strat2 substantially reduces the probability of solution. All the two-way branches become three-way branches,

and all the direct connections (e.g., 11-10 in problem 7) become binary nodes. Even with two-move lookahead, such a model would perform far below the average solution rates for our subjects.

Depth of Lookahead. Compliance with the two-move lookahead constraint was assessed by computing the proportion of trials in which subjects move to the goal directly from states that are n moves distant from the goal. Figure 4 shows the actual proportion of minimum path solutions as a function of distance from the goal. Also shown are the proportions predicted by Strat2 and by a random move generator.

Strat2's two-move lookahead predicts perfect performance from up to 2 moves away from a goal, and then a sharp decline. Subject performance is indeed quite good at 2 moves away, but it remains high (nearly 90%) for 3 moves away, rather than dropping as predicted. In fact, about 40% of the subjects exhibited perfect performance once they were 3 moves away from the goal.

Given this relatively good performance from 3 moves away, it is reasonable to consider an alternative to Strat2 that differs only in having three-move, rather than two-move lookahead to the goal. Strat3 would produce very high likelihoods of success within two trials, ranging from .97 and .94 for problems 8 and 7, to lows of .56 for problems 1, 4, and 5. Not only does Strat3 produce unacceptably high solution rates, but also, it only explains about 5% of the variance in subjects' solution rates.

If we degrade the two-move lookahead to a one-move lookahead, then we get a model that explains only 26% of the variance.

All-or-none Evaluation. Associated with Strat2's two-move lookahead is an all-or-none evaluation function. If the children were using a partial evaluation function that was sensitive to some -- but not all -- of the pieces being in their goal positions, then we should see two kinds of biases in their move patterns. One bias would show up as a tendency to favor moves -- early in the solution -- that increase the number of pieces in their goal locations. For example, in Problem 2 (18 --> 8), a first move of the cat increases the evaluation function, while moving the dog does not. The dog is also off the minimum path. Over all trials and all subjects, on this problem, the cat was moved 81% of the time. Even more revealing are the "garden path" problems. In Problem 4 (10 --> 5), the minimum path move is the mouse, which does not increase the evaluation function. Only the cat increases the partial evaluation function, and it is preferred on 66% of the trials, even though it is off the minimum path. Similarly, on Problem 5 (13 -->19), the non-minimum move of the dog is preferred on 61% of the trials.

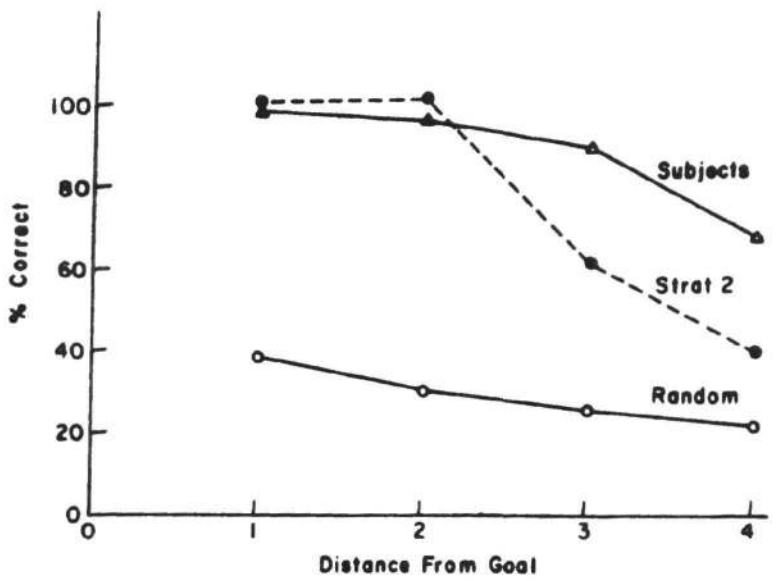


Figure 4 Proportion of minimum path solutions from n moves away, for Strat2, subjects and a random model.

The other bias would be a reluctance to remove pieces from their goal locations -- to reduce the value of a partial evaluation function. This can be assessed on Problem 3 (11-->20), where the minimum path sequence requires that the dog be temporarily removed from its goal position. On 65% of all trials with Problem 3, subjects preferred to move the cat rather than the dog, even though this took them off the minimum path. The all-or-none evaluation function in Strat2 understates the sensitivity of children to partially correct solutions.

Summary of Strategic Analysis

Strat2 explains almost 60% of the variance over all problems and 95% of the variance over the six most difficult problems. Strategies that vary the depth of the lookahead do not do as well. Strat1 explains 26% of the variance, and Strat3 only 5%. Elimination of no-backup from Strat2 yields unacceptably low solution rates.

Strat2 slightly understates children's abilities in two respects. First, the children appear to be capable of some partial evaluation, whereas Strat2 is not. Second, once they are only 3 moves away from the goal state, the children are more likely to find a minimum path solution than is Strat2. Nevertheless, within the space of

plausible alternative strategies explored here, Strat2 provides the best account of how children solve problems with ambiguous subgoals.

Discussion

Piaget (1976) concludes from his observations of 5- and 6-year-old children solving conventional TOH problems that they are unable to plan and that "There is ... a systematic primacy of the trial-and-error procedure over any attempt at deduction, and no cognizance of any correct solution arrived at by chance." (p. 291). In contrast, studies of pre-schoolers solving a modified version of the TOH (Klahr & Robinson, 1981) show that, although the amount of planning they can do is limited, the procedures they use are highly similar to adult forms.

In this investigation, pre-schoolers were presented with problems having ambiguous subgoals. We discovered that here too, Piaget's characterization does not do justice to young children's abilities. First, as described earlier, even the random component of Strat2 is highly constrained. The avoidance of double moves reveals a rudimentary knowledge about thoroughly useless actions that is not conveyed by the "trial-and-error" view. Second, solutions are not really "arrived at by chance", since there is a lookahead to the goal state, and little deviation from the minimum path, once it is in sight. Third, children use partial evaluations of nearly correct states to guide their choice of moves.

Acknowledgements

This work was supported in part by grants from the National Science Foundation (BSN81-12743) and The Spencer Foundation. Thanks to the parents, the teachers, and particularly the children of the CMU Children's School for their enthusiastic cooperation, and to Margaret Kinsky for her artistry, advice and assistance in creating materials, running the experiments and analyzing the results. Author's address: Department of Psychology, Carnegie-Mellon University, Pittsburgh, PA 15213.

References

Klahr, D. (1978). Goal formation, planning, and learning by pre-school problem solvers, or: 'My socks are in the dryer'. In R.S. Siegler (Ed.), **Children's thinking: What develops?** (pp. 181-212). Hillsdale, NJ: Lawrence Erlbaum Associates.

Klahr, D., & Robinson, M. (1981). Formal assessment of problem solving and planning processes in preschool children. **Cognitive Psychology**, 13, 113-148.

Laird, J.E. & Newell, A. (1983a). **A universal weak method.** Technical Report, Computer Science Department, Carnegie-Mellon University.

Newell, A. (1969). Heuristic programming: Ill-structured problems. In J. Aronofsky

(Ed.), **Progress in operations research, III** New York: Wiley.

Piaget, J. (1976). **The grasp of consciousness.** Cambridge, MA: Harvard University Press.

Spitz, H.H. & Borys, S.V. . (1984 [in press]). Depth of search: How far can the retarded search through an internally represented problem space. In P.H. Brooks, R. Sperber, & C. McCauley (Eds.), **Learning, cognition and mental retardation** Hillsdale: Lawrence Erlbaum Associates.

Spitz, H.H., Webster, N.A., & Borys, S.V. (1982). Further studies of the Tower of Hanoi problem-solving performance of retarded young adults and nonretarded children. **Developmental Psychology**, 18(6), 922-930.

Notes

[1]The DCM puzzle is nearly identical to the "depth-of-search" puzzle first described by Spitz & Borys (1984).

[2]The problem shown in Figure 1 is Number 3 in Table 1.

EXPERIENCE AND PROBLEM SOLVING: A FRAMEWORK

Janet L. Kolodner
Robert L. Simpson, Jr.

School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

ABSTRACT

Most research into problem solving has considered each problem to be solved as a unique event. Our observations lead us to conclude that much of the problem solving people do is based on previous experience. Analogy to previous similar problems helps in solving new problems, and each problem solving experience contributes to the knowledge available for later problem solving. This paper presents a framework for those components of problem solving which rely on previous experience. The processes involved and the organization of experience which supports those processes are considered. Examples are drawn from two problem domains: diagnosis and treatment of mood disorders and plan selection for resolution of disputes.

1. EXPERIENCE'S ROLES IN PROBLEM SOLVING

Problem-solving is a widely-studied area in both psychology and artificial intelligence (e.g., [1],[5],[7]). Yet, with rare exceptions [6], there is little study of experience's role in the process. Our observations have led us to believe that experience plays two important roles in problem solving [3]: First, **experience contributes to refinement and modification of reasoning processes.** Successful experiences reinforce already-known rules or previous hypotheses, while failures require analysis of the reasoning and knowledge used originally, and modification of faulty rules and knowledge. Experience's second role is equally important. **Individual experiences act as exemplars upon which to base later decisions.** Analogy to previous cases serves to guide and focus later decision making. An example from medicine illustrates our claims:

> Dr. X sees a patient who shows classic signs of Major Depression. She has previously been diagnosed as Depressive, and was treated in a mental hospital with antidepressants. She was sickly as a child, has had a drinking problem, and has had some unexplained physical illnesses. Dr. X concludes that she is suffering from Major Depression, Recurrent, without Melancholia and treats her with antidepressants. They seem to work, but the woman comes back complaining of additional major physical disorders. Taking a further history, the doctor finds that her unexplained medical problems have been numerous. Realizing that this is an important consideration, he makes a second diagnosis of Somatization Disorder (adapted from [11], case #125).

As a result of this case, Dr. X should learn that it is important to consider medical history in choosing predominant clinical features, and that Depression can camouflage Somatization Disorder. Using the first fact, he can

---

This research is supported in part by NSF Grant No. IST-8116892 and in part by the AFIT. The views expressed are solely those of the authors. Thanks to Dr. Robert M. Kolodner, Dana Eckart, and Katia Sycara-Cyranski.

refine his rules for choosing predominant clinical features. The relationship between Depression and Somatization Disorder will be helpful in diagnosing and treating later cases. To illustrate experience's role in providing exemplars, consider Dr. X's capabilities upon seeing a second patient diagnosed for Major Depression who also has unexplained medical problems. We expect him to transfer his knowledge from the previous case to the new one and consider whether the new patient might also have Somatization Disorder.

In building a framework for problem solving which includes experience, we must consider a number of issues:

1. Which reasoning processes use experience?
2. What knowledge is available as a result of experience?
3. How is experiential knowledge integrated into reasoning processes?
4. How does experience change the structure of knowledge in memory?

We are studying these problems in two domains: the common-sense resolution of disputes [10], and the diagnosis and treatment of mood disorders [3].

2. EXPERIENCE CONTRIBUTES TO LEARNING

We begin by considering the reasoning processes which rely on experience. We identify two experiential processes whose primary purpose is refinement and modification of reasoning processes and domain-specific knowledge: **similarity-triggered generalization** and **failure-triggered explanation.**

Similarity-triggered generalization [2] occurs when two cases already classified in the same way share additional features not accounted for by the classification. In that case, a new concept described by the shared features is created. It is a generalization of the cases and a specialization of the original classification. Thus, if most of the cases a doctor has seen in which the patient is diagnosed for Major Depression and has heart problems respond to the same treatment, then a generalization can be made that this medication is good for treating Major Depressives with heart problems. Generalization of this sort can be thought of as confirming hypotheses that might have been made on the basis of one example.

When a hypothesis is violated, or a piece of knowledge (e.g., rule) fails to work as expected, failure-triggered explanation occurs [3], [9]. An explanation for the failure is found, and the failed piece of knowledge is modified. This is illustrated in the psychiatric example above. In general, tracking down a failure and explaining it are hard problems. As we shall see later however, experience can play a role in this process if a failure is reminiscent of a previous one. A third type of learning is the integration of a new case into memory's already existing structures. This is discussed in section 4.

3. EXPERIENCE CONTRIBUTES EXEMPLARS FOR ANALOGY

A second set of experiential processes transfer knowledge from a previous case to a current one. We call the process by which this happens <u>similarity-triggered analogical reasoning</u>. When a new case is reminiscent of previous cases, those cases are used as exemplars to aid in evaluation of the new case. A prerequisite for analogical reasoning is the capability of remembering appropriate previous episodes. This will be discussed in section 4. For now, it will suffice to say that a previous episode can be recalled if it is classified similarly to the current one and has similar features not predicted by

that classification. An attempt to recall previous similar cases occurs each time new features of the current case are discovered. In general, particular past experiences called to mind by a current problem can be useful in any of the following problem solving tasks:

1. They can aid in problem classification by predicting additional features to be investigated or by pointing out alternative classifications.
2. They can help in planning by suggesting procedures or courses of action to be followed or avoided, or by suggesting a means of implementing a plan.
3. They can suggest an explanation and a means of recovery from failure.

We saw experience functioning as an aid in classification earlier when Dr. X diagnosed his second case of Major Depression combined with unexplained physical problems. While the first time he had to wait for the treatment to fail to make the secondary diagnosis of Somatization Disorder, he has an exemplar to base his diagnosis on the second time he sees such a patient.

Experience is useful in plan selection in several important ways. First a previous case can suggest a plan for problem resolution or one to be avoided (e.g., a previous treatment that worked or didn't work in a similar case). Analogical reasoning is also useful during plan selection in evaluating potential plans and in choosing between alternatives. The process involves simulating the results of alternative treatments or courses of action and evaluating them in light of previous experience. Simulation of alternatives provides hypothetical situations similar to previous ones. The success or failure of previous attempts at implementing the same plan under similar conditions provides a metric for evaluation of a potential course of action. We see this use of analogy quite often in prescribing treatment. This process is related to Schank's intentional reminding [9] and Wilensky's [12] Projector. Experience can also be helpful in choosing the means for implementing a selected plan (similar to Mostow's [4] operationalization). Any particular plan that is selected for resolution of a problem might be applied in several ways. Application of the common-sense plan "one cuts, the other chooses",for example requires differentiating between the party which will do the cutting and which will do the choosing.

Experience, as part of follow-up, aids with explanation of and recovery from failures [10]. Upon failure recognition, the reasoner attempts to recall a similar previous error. Features available for such recall include the original ones plus those associated with the failure. A previous similar failure may provide an explanation which can be applied in diagnosing the error in the current case. It may also suggest a plan for error recovery.

The following scenerio shows multiple uses of analogy in solving a complex problem. Consider a common-sense reasoner reading in the paper about the dispute between Egypt and Israel over possession of the Sinai. She knows something about the Korean War and the recent dispute between the US and Panama that resulted in the US giving back economic and political but not military control of the canal to Panama. Initial consideration of the Sinai dispute causes reminding of the Korean War since both involve disputes over land, both are competitive, and in neither can the conflict be resolved completely for both sides. Based on this reminding, she predicts that Israel and Egypt will divide the Sinai equally. She later reads that this advice was given and rejected by

both sides. Considering that "divide equally" failed, she is reminded of the time her daughters were quarrelling over an orange. She had suggested that they divide it equally, and they had rejected that, since one wanted to use the entire peel for a cake. Realizing that she hadn't taken their real goals into account, she then suggested that they divide it agreeably — one take the peel, the other the fruit. This reminding provides the suggestion that failures sometimes occur because the goals of the disputants are misunderstood. She therefore attempts an alternate understanding of Israel and Egypt's goals. Considering that Israel wants the Sinai as a military buffer zone in support of national security, and that Egypt wants the land for national integrity, she can now reconsider the conflict as a political dispute with concordant goals. Further reasoning from the orange dispute suggests that "agreeable division" based on the real goals of the disputants is appropriate. This causes reminding of the Panama dispute since it is political with similar goals and named plan. The analogy made possible through this reminding allows operationalization of the "agreeable division" plan. Using the settlement between Panama and the US as a guide, the US is replaced by Israel (the party currently in control of the object) and Panama is replaced by Egypt (the party who used to own it and wants it back). As was the case in the Panama Canal agreement, the prediction is made that Egypt will get economic and political control of the Sinai, while its normal right of military control will be denied.

4. ENCODING AND ORGANIZING EXPERIENCE

A prerequisite for learning from and using experience is the capability of retrieving relevant past experiences applicable to a new situation. The memory structure we propose is based on generalized episodes [2], [8]. These structures hold generalized knowledge compiled from the experiences they organize, and individual experiences are indexed in these structures according to their differences. When two experiences differ from the generalized episode in the same way, a collision, which we call "reminding" [2], [8] occurs. Predictions based on the first episode can be used to analyze the new one (analogy). Similarities between the two episodes can be compiled to form a new memory schema with the structure just described (generalization).

The organization provides a way of locating exemplars to use in evaluating a new case. The process which allows analogical "reminding" is a traversal procedure. When a new case is encountered, appropriate generalized episodes are chosen for it. Features which differentiate a new case from others in the same generalized episode are extracted from it and indices associated with those features are traversed. In the process, the new case collides with previous cases already indexed in memory. It is those cases which are now available for further evaluation. New cases are added to memory by the same process.

Cases are indexed in memory by their differentiating features and also by failures which occur in the course of processing. This allows learning and reminding on the basis of failure. If blame can be assigned for a failure, the case is indexed by those features which caused the failure. When a second similar situation is encountered, the marker serves as an index to a failed episode. If a solution was found to the first failure situation, it can be applied to the second so that the failure won't happen again. When blame has not been assigned, a marker denoting the difference between the failed episode and others is left, again serving as an index when a similar situation is encountered. In this case, a procedure to be avoided will be found.

In the psychiatric domain, for example, diagnostic categories (e.g., Major Depression) act as generalized episodes. The medical example above is differentiated from other cases of Major Depression by (among other things) (1) the fact that there were unexplained physical disorders in addition to those symptoms considered in the original diagnosis and (2) treatment failed in that the patient seemed cured of depression but complained of additional physical disorders.

5. WHERE DO WE GO FROM HERE?

In this paper, we have attempted to provide a framework for experience's role in problem solving. We have named processes which use experience and suggested a memory organization in support of those processes. We have not, however, stated exactly how each of the experiential processes work. We are currently investigating these processes in the two domains cited. Our memory structures, too, need considerably more work. In particular, we must specify the types of features appropriate for indexing and the allowable types of classification structures. Finally, we might be criticized for not taking into account how experiential reasoning interacts with causal reasoning. Each of the problems presented represents an important research area. It is only through investigation of each that we can discover how experiential and more traditionally considered forms of reasoning combine.

6. REFERENCES

[1] Hayes-Roth, B. (1980). Human planning processes. Rep. NO. R-2870-ONR, Rand Corp., Santa Monica, CA.

[2] Kolodner, J. (1983). Maintaining Memory Organization in a Dynamic Long Term Memory. Cognitive Science, Vol. 7.

[3] Kolodner, J. L. & Kolodner, R. M. (1983). An Algorithm for Diagnosis Based on Analysis of Previous Cases, in Proceedings of MEDCOMP 83.

[4] Mostow, D. J. (1983). Machine transformation of advice into a heuristic search procedure. In R. Michalski, J. Carbonell and T. Mitchell (Eds.) Machine Learning: An Artificial Intelligence Approach. Palo Alto: Tioga Publishing Co.

[5] Newell, A. and Simon, H. (1972). Human Problem Solving. Englewood Cliffs, NJ.: Prentice-Hall.

[6] Ross, B. (1982). Remindings and their effects in learning a cognitive skill. Cognitive and Instructional Sciences Series CIS-19. Palo Alto, CA.: Xerox Palo Alto Research Centers.

[7] Sacerdoti, E. (1975). A Structure for Plans and Behavior. Amsterdam: Elsevier North-Holland, 1977.

[8] Schank, R. (1980). Language and memory. In D. Norman (Ed.) Perspectives on Cognitive Science. Norwood NJ: Ablex Publishing Co.

[9] Schank, R. (1982). Dynamic Memory. Cambridge: Cambridge University Press.

[10] Simpson, R. (1984). Strategies for retrieval and prediction in an advisory system: A research proposal. Tech Report GIT-ICS-84/03, Georgia Institute of Technology, School of Information and Computer Science, Atlanta GA.

[11] Spitzer, R., Skodol, A., Gibbon, M., and Williams, J. (1980). DSM-III Case Book. American Psychiatric Association, Washington, D.C.

[12] Wilensky, R. (1983). Planning and Understanding: A Computational Approach To Human Reasoning. Reading, MA.: Addison-Wesley Publishing Co.

Cognitive Architectures and Principles of Behavior[1]

Pat Langley
Stellan Ohlsson
Robert Thibadeau
Robert Walter
The Robotics Institute
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213 USA

Introduction: Taxonomies and Principles

Taxonomies play an important role in emerging fields of science, since they identify significant dimensions along which the entities studied by those fields can differ. Yet one must eventually move beyond simple taxonomies to formulate *principles* that relate these dimensions of variation to observed behavior. The emerging field of Cognitive Science is concerned with the behavior of intelligent entities, both human and artificial. However, Cognitive Science is notably lacking in both taxonomies for the structure of intelligent systems, and in principles which relate such structures to intelligent behavior. In this paper, we describe an evolving taxonomy of cognitive architectures, and propose some initial principles based on this taxonomy.

Cognitive Science and its sister discipline, Artificial Intelligence, have generally been empirical sciences, in that they have spent considerable time collecting *examples* of intelligent behavior, through experiments with humans and through constructing simple intelligent artifacts. This work has been worthwhile and should continue, but eventually we must begin to develop *theories* of intelligence that cover not only human intelligence, but cognitive behavior in general. Since different intelligent entities may rely on different cognitive mechanisms, Ohlsson [1] has proposed that a general theory of intelligence must be concerned with the *relation* between such mechanisms and the form of intelligent behavior that results.

In science, a researcher often limits his attention to ensure progress, and in this case we have focused on the class of cognitive architectures known as *production systems*. Production systems were first proposed as models of the human information processing architecture by Newell and Simon [2]. Since that time, they have been used to simulate a variety of intelligent behavior, ranging from problem solving to natural language understanding to cognitive development. Production system schemes have a number of features that make them attractive candidates for cognitive architectures, independent of their value as models of human behavior. For instance, they seem to be a viable compromise between the stimulus-response approach of behaviorism, and the goal-driven approach of cognitive psychology. In addition, the relative independence of the condition-action rules making up a production system program lends itself to modeling the learning process, since interaction between new and old components will be minimal.

Dimensions of Production System Architectures

Our research goal has been to identify the significant dimensions along which production system architectures may vary. This has resulted in a formalism, PRISM2, which specifies a set of such dimensions, along with possible values for each of these dimensions. Using this formalism, one may succinctly describe architectures that have been explored by other researchers, as well as architectures that have never before been examined. This allows comparison of the differences between various architectures, and should facilitate communication between researchers in the area.

Since we are concerned with the relation between architectures and intelligent behavior, we have implemented PRISM2 in such a way that one can "run" an architecture in conjunction with a particular production system program. Thus, PRISM2 has some of the characteristics of a programming language,

---

[1]This research was supported by Contract N00014-83-K-0074 from the Office of Naval Research. We would like to thank David Nicholas, Robert Neches, and Rolf Pfeifer for their assistance in the early stages of our work on PRISM2.

though it actually defines an entire *class* of production system languages. In this context, a given architecture can be viewed as providing "free" control structure that need not be specified explicitly in the program itself. Let us examine the dimensions of architectural variation supported by the PRISM2 formalism:

- *The structure of memory.* PRISM2 provides for multiple declarative and production memories, allowing "flat" production system schemes, hierarchically organized systems, or more exotic control structures. Moreover, each memory may have different characteristics. E.g., PRISM2 allows arbitrary numeric attributes (such as strength, activation, and affect) to be associated with each memory.
- *Decay and forgetting.* Production system architectures differ in the manner in which memory elements decay over time, and in the details of the forgetting process. In PRISM2, elements in a given memory may decay by a fixed amount on every cycle, or as a function of the number of elements entering memory. The formalism supports a number of alternate decay and forgetting methods.
- *Retrieval by spreading activation.* Production system architectures differ in the manner by which they retrieve forgotten elements through spreading activation. E.g., activation may decay according to different functions as it spreads through adjacent elements, ceiling effects may occur, and the threshold below which activation may not spread can differ. PRISM2 supports a variety of constraints on spreading activation.
- *The match process.* Production system architectures differ in their matching abilities, and PRISM2 supports a variety of different matching styles. E.g., conditions may match against embedded structures, find sequences of symbols, and match against lists as though they were sets. In addition, the user may require one-to-one mappings between conditions and memory, or allow many-to-one matches to occur.
- *Conflict resolution.* Production system architectures differ in the conflict resolution methods they employ for selecting among competing instantiations of rules. Three relevant dimensions of conflict resolution are:
  - *Ordering strategies.* The architecture orders instantiations of productions along some dimensions, such as recency of matched elements, or specificity of the matched rules.
  - *Selection strategies.* The architecture selects one or more instantiations based on the resulting ordering; e.g., the best instantiation may be selected, or all those above a certain threshold may be chosen.
  - *Refraction strategies.* The architecture may remove some instantiations permanently; e.g., it may remove all instantiations that applied on the last cycle, or all instantiations currently in the conflict set.

  The PRISM2 formalism supports many different combinations of ordering, selection, and refraction strategies for implementing alternate conflict resolution methods.
- *Learning methods.* Production system architectures differ in the processes they use to learn new condition-action rules. Common methods include:
  - *Discrimination learning*, in which errors lead to more specific rules based on differences between positive instances and negative instances of the errorful production.
  - *Generalization learning*, in which specific productions with similar structures lead to more general rules based on features common to the original rules.
  - *Composition*, in which two or more rules that tend to fire in sequence are combined into a more complex production that leads to the same results.
  - *Proceduralization*, in which a specific version of a general rule is based on the current instantiation of that production.

  PRISM2 supports each of these learning methods, as well as providing the ability to modify the detailed characteristics of each method. The learning methods may be used in conjunction or in isolation.

The basic organizing principle for specifying PRISM2 architectures is the architectural *template*. Different templates are available for specifying the characteristics of declarative memories, production memories, and action side functions responsible for adding new elements, decay and forgetting, spreading activation, and learning new rules. Each template has an associated set of *parameters*, whose values determine the exact behavior of that component of the system.

For example, the spreading activation template contains three main parameters. The first of these, **spread-from-element**, contains a list of steps taken when activation spreads out from a memory element; this might include actions such as dividing the available activation by the number of symbols in the element and causing this activation to decay by a certain amount. The second parameter, **spread-through-symbol**, specifies a list of steps taken as activation spreads through a symbol contained in a memory element. This can include actions such as dividing up the activation by the number of other memory elements containing this symbol, leading to a form of the fan effect. Finally, the parameter **spread-to-element** specifies a list of actions carried out when activation spreads to a new memory element. This may include tests for whether to continue spreading activation, as well as constraints on the amount retained by the element, leading to a ceiling effect.

Towards Principles of Intelligent Behavior

The flexibility of the PRISM2 framework has allowed us to experiment with alternate production system architectures. To date, most of our experiments have involved alternate conflict resolution schemes and different methods for learning new productions, and we shall draw our examples of principles from these areas. While we would not claim that the PRISM2 formalism was necessary for generating these principles – in fact, there was a strong analytical component in both cases – it has certainly helped us in clarifying and testing our ideas. In our future work, we hope to examine the relation between other dimensions and behavior, and would welcome any other research with similar goals.

The first example relates conflict resolution strategies to the notion of *search*. In recent years, a number of production system models have been implemented in which the rules play the part of operators for moving through some problem space [3, 4]. Within this framework, the conflict resolution strategy used by the system determines the form of search it carries out. For example, Young [5] has proposed the following principle:

- *Recency-based conflict resolution schemes lead to depth-first search behavior with automatic backtracking.*

To be more specific, depth-first search behavior results when the architecture prefers instantiations matching against elements added to memory more recently, and when the single best instantiation is then selected for application. Automatic backtracking also results, provided that refraction removes applied instantiations from the conflict set. This relation has been known informally among production system users for years, but we believe it is important to note its status as a basic principle of cognitive architectures.

Since other conflict resolution schemes are possible, an obvious question is whether other search strategies arise from alternate architectures. Another popular conflict resolution scheme allows all instantiations to apply in parallel, unless they have been applied on an earlier cycle [6]. Upon reflection, this strategy leads to a second well-known search method – breadth-first search. Thus, we can formulate a second principle relating architecture to behavior:

- *Conflict resolution schemes involving parallel firings lead to breadth-first search behavior.*

In this case, no backtracking is required, and refraction is used only to keep instantiations used earlier from applying again, since there are no other constraints on the selection process. Presumably, other relations between conflict resolution and search methods exist, and these will be discovered as researchers further explore of the space of architectures.

Our second example involves the area of learning methods. In particular, we have been concerned with the distinction between *discrimination* learning, in which one moves from general rules to more specific ones, and *generalization* learning, in which one moves from specific rules to more general ones. The most common form of discrimination involves creating some variant of an existing rule containing additional conditions. Since such a learning system begins with simple rules and generates more complex ones only when errors of commission occur, we arrive at the principle:

- *Given two rules of different complexity, a discrimination-based learning system will master the simpler rule before the more complex one.*

In contrast, generalization involves the opposite process of creating a production with fewer conditions than an existing rule. Since such a learning system begins with complex rules and generates simpler ones only

when errors of omission occur, we arrive at another principle:

- *Given two rules of different complexity, a generalization-based learning system will master the more complex rule before the simpler one.*

These complementary principles relate learning methods to the rate at which rules of different complexity will be mastered. Although details are absent, even such global statements can be very useful. For instance, empirical studies of human language acquisition suggest that more complex function words are mastered later than simpler ones, suggesting that discrimination learning is a more likely explanation than generalization [7].

Discussion

In the preceding pages, we examined some principles that relate characteristics of the architecture – conflict resolution methods and learning mechanisms – to aspects of intelligent behavior – search strategies and rates of mastery. These principles are explanatory in the sense that they account for behavior in terms of underlying components, just as physical principles account for observed phenomena in terms of inferred properties. However, note that one cannot begin to formulate such principles until one has some ideas about the nature of the components underlying behavior. This is the reason we have focused on developing PRISM2, a formalism which allows us to explore the space of cognitive architectures, to represent the differences between architectures explicitly, and to actually test specific production system architectures in particular domains. The dimensions of variation supported by PRISM2 provide us with a taxonomy of architectural types, which we can then use in formulating principles of behavior.

Admittedly, the principles we have examined are only a beginning, and we are far from a complete theory that relates architectural components to aspects of intelligence. Our principles were intended mainly as examples of relations that one can express using the PRISM2 formalism, and as examples of what we believe should be a more common goal in our developing field. In fact, some readers may disagree with the principles themselves [8], and we would welcome suggestions for modifications and improvements. However, we hope to have convinced the reader that Cognitive Science is ready to begin formulating such principles, and that other researchers will join us in identifying the varieties of cognitive architectures, and in the more long range search for a general theory of intelligent behavior.

References

1. Ohlsson, S. "Mechanisms, behaviors, principles: A time for examples." *AISB Quarterly 48* (1983), 24-25.

2. Newell, A. and Simon, H. A.. *Human Problem Solving.* Prentice-Hall, Inc., Englewood Cliffs, N.J., 1972.

3. Ohlsson, S. A constrained mechanism for procedural learning. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983, pp. 426-428.

4. Laird, J. E. and Newell, A. A universal weak method: Summary of results. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983, pp. 771-773.

5. Young, R. M. Architecture-directed processing. Proceedings of the Fourth Annual Conference of the Cognitive Science Society, 1982, pp. 164-166.

6. Thibadeau, R., Just, M. A., and Carpenter, P. A. "A model of the time course and content of reading." *Cognitive Science 6* (1982), 157-203.

7. Langley, P. "Language acquisition through error recovery." *Cognition and Brain Theory 5* (1982), 211-255.

8. Bundy, A. "Superficial principles: An analysis of a behavioural law." *AISB Quarterly 49* (1983), 20-22.

A Psychologically Plausible Representation for Reasoning about Knowledge*

Anthony S. Maida

Program in Cognitive Science, T4
UC Berkeley
Berkeley, CA 94702

Richard B. Millward

Center for Cognitive Science,
Box 1911
Brown University
Providence, RI 02912

1. INTRODUCTION.

Designing a computer program to reason about the knowledge states of cognitive agents is a difficult matter. To ask that this reasoning be done in a human-like way is even more difficult. This paper describes a psychologically plausible representation and algorithm for representing and processing information about other cognitive agents' knowledge states.

1.1. The Fregean Approach to Reasoning about Knowledge

We are concerned with keeping track of coreferent terms in a memory which contains assertions about the knowledge states of cognitive agents. The situation can be illustrated by McCarthy's (1979) "telephone number problem" in which there are two phone numbers, Mike's phone number and Mary's phone number, which are coreferent. It follows that if a person, say Ed, dials Mike's phone number, he also dials Mary's phone number. However, if he knows Mike's phone number, it does not follow that he knows Mary's phone number. The Fregean approach (e.g., McCarthy, 1979) toward solving this problem is to claim that the phrase "Mike's phone number" means its referent, in normal contexts, but means something else (called the "sense") in contexts involving knowledge.

Creary (1979) and Barnden (1983) have shown that use of the Fregean approach requires more than just sense and reference; it requires a hierarchy of concepts. For example, in sentences (1)-(3) below, the phrase "Mike's phone number" must be represented as: 1) the number itself; 2) the concept of the number; and, 3) the concept of the concept of the number. For each embedding in a knowledge context, we must go one level deeper in the concept hierarchy.

(1) Ed dials Mike's phone number.
(2) Pat knows Mike's phone number.
(3) Tony knows that Pat knows Mike's phone number.

2. COGNITIVE SIMULATION APPROACH

We now describe another approach to this problem. Any concept that a human is capable of contemplating at the level of introspection should be representable as a node, or some equivalent kind of cognitive unit, in a simulation's memory. We will interchangeably call these

The authors acknowledge Nigel Ward for careful criticisms of an earlier draft of this paper. The first author was supported by the A.P. Sloan Foundation.

nodes "cognitive units" or "mental OBJECTs," to signify that they correlate with, or represent, manipulable objects of thought in a human mind. There should be no mental OBJECTs which represent concepts, ideas, or distinctions that humans do not use. These premises are a version of the so-called Uniqueness Principle described in Maida and Shapiro (1982). According to this principle, then, the phrase "Mike's phone number" should not be mapped into three distinct mental OBJECTs or cognitive units in sentences (1)-(3) unless there is introspective evidence that humans make this same kind of distinction. The remainder of this paper describes how to make this simple representation work.

We must augment this representation with processing that manages knowledge states. This involves two components. One is a scheme to maintain canonical names for equivalence classes of coreferent OBJECTs. The other is a scheme to maintain knowledge contexts for the mental OBJECTs. Each knowledge context will associate a mental OBJECT with a canonical name. Assertions will be stored with the canonical name selected by the current knowledge context.

3. THE KNOWLEDGE CONTEXT ALGORITHM.

The Knowledge Context Algorithm processes assertions of knowing with respect to the substitution of coreferent terms. The algorithm handles nested knowing such as the assertion: "Tony knows that Pat knows Mike's telephone number." The algorithm draws all valid inferences which follow strictly from the substitution of coreferent terms while drawing no invalid inferences (cf Maida, 1984).

3.1. Assigning Knowledge Contexts to Mental OBJECTs.

Expression (4) below makes reference to three concepts which will be represented as distinct mental OBJECTs. They are Tony, Pat, and Mike's-phone-number.

(4) (know-that Tony (know-value-of Pat Mike's-phone-number))

We must assign a knowledge context to each of these OBJECTs. An OBJECT in an assertion acquires its knowledge context from two sources. They are: 1) the OBJECT's knowledge context within the assertion (i.e., whether it is the second argument of a "know-that" or "know-value-of"), and 2) who happens to believe the assertion. Assuming the assertion resides in the top level of the system's memory, the knowledge contexts for each of the objects in the assertion appear in Table 1.

Table 1

Knowledge Contexts Assigned to the
OBJECTs appearing in Expression (4)

OBJECT	Knowledge Context
Tony	System
Pat	System-Tony
Mike's-phone-number	System-Tony-Pat

A hyphenated context such as "System-Tony" should be interpreted as the system's knowledge of Tony's knowledge.

3.2. Context Relative Equivalence Classes

Multiple cognitive units can turn out to be coreferent. Sets of OBJECTs known to be coreferent create equivalence classes. This scheme as it stands however can lead to a serious cross-referencing problem because multiple facts about a single real-world object can be sprinkled among any of the OBJECTs in the equivalence class.

We use a technique first employed by McAllester (1980) which involves assigning canonical-names (c-names) to equivalence classes. We shall augment McAllester's scheme with a manager for knowledge contexts. Henceforth, a mental OBJECT in an equivalence class will have a canonical name with respect to its knowledge context. Figure 1 depicts a situation in which the system knows the coreference of the three units in the set {the Morning Star, the Evening Star, Venus}, whereas it knows that Pat knows the coreference of only two of these, namely the units in the set {the Morning Star, the Evening Star}. Note that the unit representing the Morning Star has a canonical name which depends on the currently active knowledge context, i.e., in this case, whether it is the system's knowledge or the system's knowledge of Pat's knowledge.

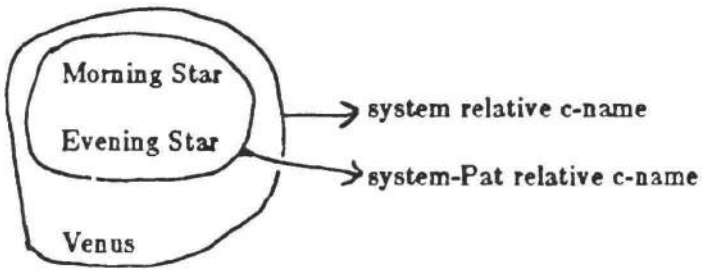


Figure 1

A OBJECT can be in Multiple Context-Relative Equivalence Classes each having their own Canonical Name.

3.3. Processing with the Knowledge Context Algorithm

Consider the units representing "Mike's phone number" in (5a) and (5b).

(5a) Pat dials Mike's phone number.
(5b) Pat knows Mike's phone number.

For a given OBJECT, information about it is stored with a canonical name that depends on the OBJECT's current knowledge context. In sentence (5a) the knowledge context for the OBJECT representing Mike's phone number is **system** and its canonical name in that context will necessarily be the same as the canonical name for the data base OBJECT representing Mary's phone number. Since extensional information pertaining to either Mike's or Mary's phone number will be stored with this canonical name, inferences which follow from substitution of coreferent terms are made implicitly by the storage scheme. However, in sentence (5b) the relevant knowledge context is **system-Pat** and the canonical name for the OBJECT representing Mike's phone number in this context will only be the same as the one for the OBJECT representing Mary's phone number if the system has an assertion residing in its data base asserting that Pat knows the phone numbers are coreferent. If the assertion resides in memory, the inference goes through, otherwise not; these are exactly the performance characteristics that we want.

3.4. Related Psychological Evidence

Anderson (1978) studied the question of what happens when two structures in human memory, previously believed to correspond to distinct real-world entities, are learned to be coreferent. When the subject learns of the coreference of distinct cognitive units, he or she gradually

migrates the factual information from one unit to the other, abandoning one unit. The unit previously used the most dominates. The subject's choice, based upon amount of previous use, is an arbitrary choice from a semantic or conceptual standpoint.

4. CONCLUSION

We presented a method of representing information about the knowledge states of other cognitive agents which is psychologically more plausible than the Fregean method. Purely Fregean methods for representing knowledge about knowledge make more distinctions than a human thinking about the same problem would make. The present method makes exactly the right number of distinctions.

References

Anderson, J.R. The processing of referring expressions within a semantic network. In TINLAP-2, 1978, 51-56.

Barnden, J.A. Intensions as such: an outline. In IJCAI-83, 1983, 280-286.

Creary, L.G. Propositinal attitudes: Fregean representation and simulative reasoning. In IJCAI-79, 1979, 176-181.

Maida, A.S. Selecting a humanly understandable representation for reasoning about knowledge. To appear, International Journal of Man Machine Studies, 1984.

Maida, A.S. & Shapiro, S.C. Intensional concepts in propositional semantic networks. Cognitive Science, 1982, 6, 291-330.

McAllester, D. The use of equality in deduction and knowledge representation. AI-TR-550, MIT Artificial Intelligence Lab, 1980.

McCarthy, J. First order theories of individual concepts and propositions. In J. Hayes & D. Michie (Eds.) Machine Intelligence 9, New York: Halsted Press, 1979.

TUTORIAL GOALS AND STRATEGIES IN THE INSTRUCTION OF PROGRAMMING SKILLS

Jean McKendree, Brian J. Reiser, and John R. Anderson

Advanced Computer Tutoring Project
Department of Psychology
Carnegie-Mellon University
Pittsburgh, PA 15213

Current research on Intelligent Computer-Assisted Instruction (ICAI) has largely focused on modelling expert domain knowledge, student's knowledge, and error diagnosis techniques. In most current ICAI systems, tutorial strategies are implicit only in the architecture of the system, and cannot be dynamically chosen during the instructional conversation with the student. While learning theories may suggest efficient strategies for the learner, they have little to offer the instructor in guiding effective presentation of material or in making strategic decisions regarding tutorial methods.

In this paper we examine the range, situational determinants, and effectiveness of tutoral strategies. Our analysis of tutoring strategies is based on recordings of fourteen undergraduates being individually tutored on the LISP programming language. The students had no prior experience with LISP; approximately three-fourths had prior programming experience with BASIC or Pascal. Five psychology and computer science students who had prior experience as teaching assistants or private tutors for LISP served as tutors in this experiment. Each student read a text describing basic LISP functions and syntax and worked through a set of problems with the tutor's assistance. Tutors were told only to help the students solve the problems in the text, and were free to use whatever techniques they found appropriate. All terminal interactions were recorded and the sessions were video-taped.

We characterize the tutorial interactions in terms of the tutorial goals (e.g. clarify misconception, guide problem-solving, promote exploration, elaborate knowledge), the strategy or plan enacted to achieve the goal (e.g. analogy, restructuring a problem, reminding, statement of facts), and the problem-solving context in which the interaction occured. This characterization indicates the manner in which tutors dynamically devise an individually tailored curriculum. Most of the interactions were directed at two major tutorial goals. First, tutors provided information to clarify student misconceptions. In

these situations, the student exhibits a misunderstanding or slip in the execution of a procedure. Examples are using a synonym for the desired function (ADD vs PLUS), typing an incorrect number of parentheses or choosing an incorrect combining function. The strategies chosen by the tutor varied with the student's past performance and the current state of problem solving. The strategies observed included:

* **Analogy.** A typical problem with instruction in a new domain is the student's lack of conceptual structure to apply to the abstract ideas presented in an attempt to construct new rules. By presenting and then guiding application of an analogy, tutors aid the student in developing a useful model for the new domain.

* **Fact provision.** When an error is judged to be either a slip or a non-serious misconception, tutors provide the necessary information directly in order to facilitate problem solving or to lessen the student's memory load. Typical examples of inputs eliciting this response are the generation of a synonym for a function name or unbalanced parentheses.

* **Reminding.** Knowledge learned in previous episodes is often unavailable to the student for immediate recall in a different context. Tutors remind students of previous episodes to encourage them to apply knowledge learned in those contexts to the new problem.

Second, tutors often set new goals for the student within the current problem. Although students may possess the necessary procedures for solving components of the problem, they are often unable to partition the problem into manageable pieces. Our tutors tended to provide hints or suggestions concerning the next goal in the problem-solving, rather than simply providing the next step in the problem and encouraging the students to continue. Furthermore, hints were preferred to stating rules in general. The three primary strategies used within this goal context were:

* **Decomposition.** Often at the beginning of a problem, the student will flounder, seemingly overwhelmed. Tutors offer direction by focusing on a subpart of the current problem either by explicitly suggesting a goal or by asking leading questions.

* **Reminding.** Tutors use previous episodes to guide the solution path as they do in trying to clarify a misunderstanding. However, in these cases, the reminding is used to prompt the student to recall previous solutions in order to construct a parallel plan for the current problem.

* **Simpler problem.** Students may become fixated on a particular point or intimidated by a complex problem. Here, tutors generate a simpler problem to

be solved that contains the essential features. The student, perhaps with the tutor's guidance, can then apply this solution in the new context.

In addition, tutors reinforce correct concepts, elaborate knowledge and promote active exploration. These strategies are less prevalent in our tutorial transactions, but help to vary the tutor-student interactions and to enrich the knowledge provided for the students.

Finally, we discuss why these tutorial strategies are effective within the ACT* theory of learning (Anderson, 1983). While it may seem that providing facts and rules at appropriate times could be the most helpful interaction provided by a tutor, ACT* offers some insight as to why these strategies are less effective than the "indirect" strategies preferred by our tutors.

* **Setting goals.** Our tutors tended to provide hints or suggestions concerning the next goal in the problem-solving, rather than providing either an applicable rule or the result of that rule. The problem with referring to general rules is that students may not have the conceptual vocabulary to correctly represent the salient problem features involved in the rules. When the tutor intervenes by setting the next goal, the student may more easily access a weakly encoded rule and thereby strengthen it through execution in the appropriate goal context. Furthermore, an opportunity is offered for incorrectly encoded rules to be accessed and debugged within the current context. If the student were given the next result they may simply accept it and never access their weak or incorrect rule. Similarly, if the tutor had simply presented the general rule that was applicable in the current context, the student could encode it declaratively, missing an opportunity to strengthen an existing but currently inaccessible rule.

* **Guided Generalization.** Our tutors often redefined the problem for the student by selecting simpler problems with the same essential features. These cases enable the student to access a rule which they have already acquired and to apply it to the new context. This allows the learning mechanisms to generalize from the instances and to construct a more generally applicable rule.

In fact, these "indirect" strategies lead to faster learning rates and better performance for our subjects than do the standard pedagogical environments (reading texts, attending lectures, working problems). We believe these techniques enable the student to access and therefore to modify or generalize existing rules whereas factual information presented by the tutor is likely to be encoded declaratively or in an overly specific form.

A Problem Space Perspective on the Development of Children's Understanding of Gears

Kathleen E. Metz

Carnegie-Mellon University

Abstract

This paper investigates two contexts of children's developing knowledge of the physical world: (1) the macro-context of different age cohorts (8-9 years versus 11-12 years); and (2) the micro-context of a one-hour experimental session. Twenty subjects were video-taped, constructing goal-states for a task involving gears. Four distinct systemic approaches or problem spaces were identified: (a) Euclidean, (b) Kinematic, (c) Dynamic, and (d) Topological. The Arithmetic Modifier, effecting a numerical characterization of a problem space, can operate on any of the four. Cross-age, there was the substantial overlap of initial problem space employed, and minimal overlap of final problem space. This frequency of adaptive shift in problem space, strongly and positively correlated with age, suggests that, when confronted with an unfamiliar task domain, the capacity to recognize a problem space as inappropriate and to evoke another more adequate problem space appears to be a component of the answer to the classic question, "What develops?"

Introduction

It is proposed here that the Information Processing construct of problem space is a potentially powerful conceptual and representational tool for the investigation of children's thinking, particularly well-suited to the critical analysis of the systemic view of understanding and change. This research project uses the problem space construct to compare different aged children's capacity for understanding a physical task domain with which they are not familiar, and their capacity for adaptive change. The physical task domain chosen for this study is gears; in particular their models of relative directionality.

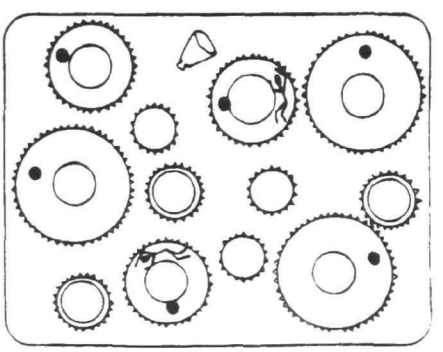
An instructionless experimental procedure has been developed, so as to enable the ecological investigation of systems of understanding and possible systemic changes. The procedure was derived from a line of research recently developed in Geneva by Bärbel Inhelder and her team, utilizing micro-analysis of conceptual change occurring in the context of children's problem-solving (**c.f.** Blanchet, 1977; Inhelder **et al.**, 1976).

Method

Subjects were drawn randomly from the population of graduates of a university laboratory preschool, who still lived in the area. The population is predominately middle and upper-middle SES.

The set of materials given to the subject consists of 12 gears, 3 each of 4 different sizes, a Velcro board, and a knob. The gears can be easily attached or removed from the board, by means of a Velcro adhesive on the inner circle of the gear-back. Two of the gears have tape on them, with a drawing of a man in the course of somersaulting. Turning a marked gear counterclockwise gives the appearence of a man somersaulting head-first; turning clockwise, of somersaulting feet-first. The knob can be placed in any of 6 of the 12 gears, including both of the marked gears.

Figure 1 · The Materials



The instructions to the subject are: " There are a bunch of these things. Two of these have men on them. Lots of them don't. The men can do head-somersaults like this. Or feet-first somersaults like this. The game is to make something, using any of these things you like, so that when you turn the knob (See, the knob can fit into any of these holes) both men do head-first somersaults. Make something, using any of these things you like, so that when you turn the knob, this man and this man are both doing head-first somersaults. Please think outloud. " After the child has built one successful construction, the experimenter extends the task:" Good! Now the game is to see how many **different** ways there are to get the two men to do head-first somersaults. Here's a pen and some paper. Use them to keep track of the ways you find. Remember! It's important to think outloud!"

The simplest and most efficient means of resolving the task entails the consideration of only one relation found within the gear-configuration in a state of no-motion: the parity or non-parity of gear-elements between the marked gears. When there is an odd number of gear-elements between the marked gears (and no other connections with an even number of gear-elements), then the two marked gears will turn in the same direction. Other approaches, such as the abstraction of patterns of relative motion or the calculation of the directional effects of pushes across pathways of transmission of movement, although less efficient, offer progressively more adequate models of the phenomenon of relative directionality of gear movement, and alternative paths to goal attainment.

Results and Discussion

The results are organized into two levels: first, a description of each problem space, as summarized from the coding criteria (criteria, developed across three pilot studies, by the gradual refining of the match between data and models); and second, a cross-age comparison of range and distribution of problem spaces, and the frequency of adaptive problem space shift. In the coding of the protocols, the trained raters' agreement was 91.3 %.

1. The four problem spaces

The Euclidean Problem Space

The Euclidean Problem Space is composed of elements and relations of Euclidean geometry, such as size of gear elements, particular alignment among gear placements or positioning on the board surface, gears-configuration shape, and symmetries. Each of these is irrelevant to the attainment of the goal.

A particularly interesting and common type of error is the use of symmetry as a means to achieve correspondence of displacements. Two strategies were based on symmetry. One entailed symmetrical matching of men-orientations. These subjects vascillate between mirror and slide symmetry, convinced that the correct symmetrical relation between men figures (in addition to the correct positioning of one element relative to the other) should solve the problem. In the second type, the subject tries to attain the goal by means of the bi-laterally symmetrical placement of the marked gears in a bi-laterally symmetrical gears-configuration. It is hypothesized that visual symmetry is one primitive heuristic employed by young physics-naive subjects, seeking to create identity of actions (as in this task) or equilibrium (as Inhelder and Piaget (1958) reported of their youngest subjects in their balance beam task).

The Kinematic Problem Space

The Kinematic Problem Space entails the enactment of a new data base, gears in motion. This focus on motion is manifested by extensive motion study, above and beyond that necessary to evaluate constructions as failures or successes (**e.g.** an examination of motion of non-marked as well as marked gears; setting gears into motion with only one or with no marked gears on the board; continuing to turn the gear construction, even after it has been evaluated as a success or failure).

The conceptual framework consists of these motions, and secondarily, the placements that effect them, defined either in terms of a Euclidean relation (**i.e.** the particular alignment of each element relative to the others) or Topological relations (more simply, which elements are touching). This data base enables the abstraction of goal- relevant kinematic relations and patterns.

In contrast to the Euclidean Problem Space, the Kinematic Problem Space is a fundamentally goal-appropriate conceptual framework. The sphere of relative motions of all gears, marked and

nonmarked, is an effective way of observing one's evolving constructions, of abstracting relations and patterns, and formulating constraints for gear-constructions. A weakness of the space is that these relations and patterns among the motions remain arbitrary empirical observations; i.e., they do not transcend the descriptive.

The Dynamic Problem Space

In the Dynamic Problem Space, on the basis of such entities as agents and patients, and pathways of transmission of movement, inferences are formulated concerning how objects **act** upon other objects, so as to effect particular patterns of displacements. Subjects conceptualize the turning of the knob as creating a force that is transmitted across the device, along the pathways of transmission of movement.

As the Dynamic framework involves inferring the sequence of the displacements across each pathway, it can impose significant demands on STM, particularly when the subject does not linguistically tag the directionalities. The space's strength is the initial explanation it offers of the phenomenon.

The Topological Problem Space

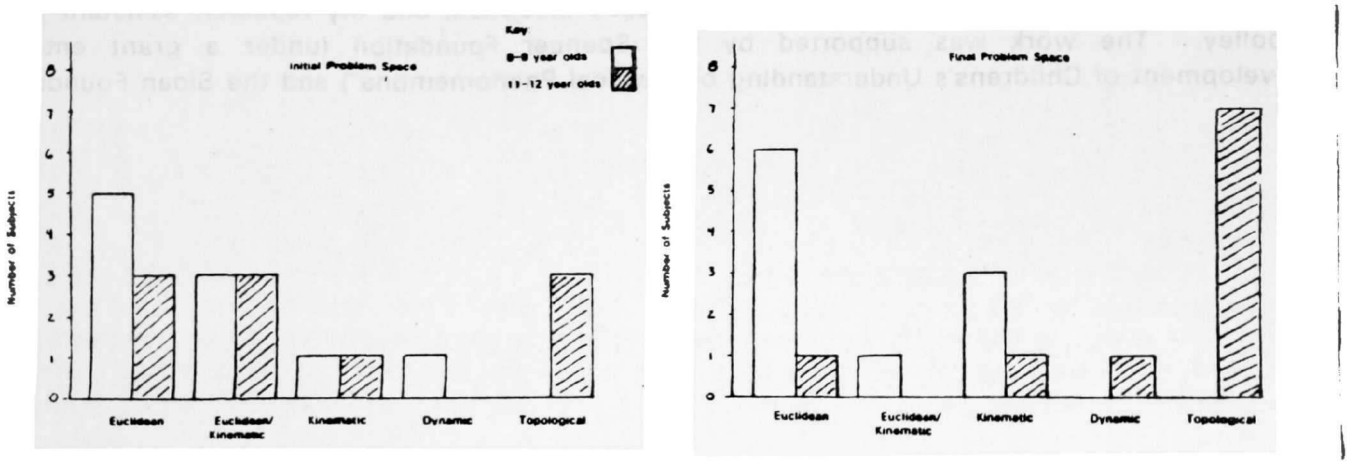
The conceptual framework of this problem space is based upon Topological Geometry . It is similar to the Euclidean, in that it is a static geometric framework, but different in that many distinctions within the Euclidean framework are not considered differences in the Topological. In the Topological, the subject assumes only connectedness of gears is relevant to the goal criterion, and ignores particular alignments among gears, their sizes, and the visual gestalt the gear construction may form.

The primary strength of the Topological Problem Space is that it facilitates the highly efficient enumeration of the complete set of possibilities (as defined by the conceptual system). The primary problem with the Topological Problem Space is the arbitrary quality of understanding of relative directionality, as manifested by the common confusion concerning which elements to count, the subjects' spontaneous descriptors of the odd/even rule (**e.g.**as the "trick" or the way the Experimenter "fixed the game"), and subjects' inability to offer explanations of the phenomenon.

The Arithmetic Modifier

The Arithmetic Modifier does not affect the conceptualization of the task domain, nor the heuristics, apart from the numerical characterization of the units of meaning, as defined by the semantics and syntax of that particular problem space. The benefits of such formalism are comparatively obvious, i.e. greater efficiency of task resolution or ease in identifying patterns. The primary liability identified in this data set is the dissociation of the arithmetic from the semantic and syntactic referent.

Cross-age Comparison of Initial and Final Problem Spaces



Cross-age comparison of range and distribution of problem spaces, and frequency of problem space shift

There is substantial cross-age overlap in initial problem space (See Figure 2.) With the exception of the first 5 episodes of one 8-9 subject, **all** of the 8-9's began the task in the Euclidean or Kinematic problem spaces, or some utilization of both (in vascillation or combination); 70% of the 11-12's did as well.

The cross-age overlap of final problem space is much smaller (See Figure 2). Seventy percent of the 11-12's end the task in the Topological space, a space that no 8-9 employed under the conditions of this experiment. One 11-12 ended the task operating confidently in the Dynamic Problem Space, a space no 8-9 was able to sustain. **All** of the 8-9's ended the task in the Euclidean or the Kinematic, or some utilization of both, as compared to only 20% of the 11-12's.

One intriguing cross-age difference embedded in the comparison of initial and final problem spaces is the frequency of adaptive change of problem space. There is a strong tendancy among the 8-9's towards absence of problem space shift. Conversely, there is a strong trend among the 11-12's who do not begin in an adequate space, towards adaptive change of problem space.

Conclusions

The dramatic cross-age difference in range was the eventual high frequency of the Topological Space among the older subjects, and its complete absence amoung the younger subjects. It is hypothesized that the reason that the Topological framework, documented in the literature as the most developmentally primitive geometry (**e.g.**Piaget, Inhelder & Szeminska, 1960) is the last employed is that its usage here implies the recognition of highly salient Euclidean features as irrelevant, and hence constitutes a less perceptually-bound, more abstract choice.

Second, the high frequency of adaptive problem space shift among the older subjects suggests that the abilities to recognize a space as inappropriate and to evoke a more appropriate one are components to the answer to the classic developmental question of "What develops?".

Finally, the study documents that children's understanding of physical phenomena can be described in terms of systems or paradigmatic approaches, and that the development of understanding can be represented in terms of changes in these systems.

References

Blanchet,A. 1977. La construction et l'équilibre du mobile. **Archives de Psychologie,** 45 (173) 29-52

Inhelder, B. & Piaget, J., 1958, **The growth of logical thinking from childhood to adolescence**.N.Y.:Basic Books

Inhelder, B., Ackermann-Valladao, E, Blanchet, A., Karmiloff-Smith, A. Kilcher-Hagedorn, H., Montangero, R. & Robert, M., 1976. Des structures cognitives aux procédures de découverte. **Archives de Psychologie**,44, (171), 57-72

Piaget, J., Inhelder, B. & Szeminska, A..1960 **The child's conception of geometry**. London: Routledge and Kegan Paul.

Acknowledgements I would like to thank Herbert Simon and James Greeno, for numerous discussions that had a substantial impact on the study, Alex Blanchet and Barbel Inhelder for suggestions and critical comments at the project's inception, and my research assistant Jacqueline Woolley. The work was supported by the Spencer Foundation (under a grant entitled "The Development of Childrens's Understanding of Physical Phenomemona") and the Sloan Foundation.

Context Dependencies in Features Used to Evaluate States

Donald Mitchell

Northwestern University

Problem solvers must consider the situational factors which influence the predictiveness of features used to make judgments. In an analysis of positions from Othello games, we found that the predictive validity of many features was dependent on the stage of the game and the skill of the players. This finding supports the use of context-sensitive weights for individual features in evaluation functions -- such as the application coefficients proposed by Ackley and Berliner (1983). In addition, our research provides a method for discovering these dependencies and for testing the general validity of features.

Ackley and Berliner (1983) define two important components of game playing programs: reasoning and judgment. In their words, "reasoning...is the process of imagining the environment to be other than it is (p.3)," and "judgment... is the process of forming an interpretation of the environment with respect to a goal (p.3)." In most programs, reasoning is the search algorithm and judgment is the evaluation function. Most game-playing research has focused on the reasoning process rather than the judgment process. Wilkins (1979) and Berliner (1983) are exceptions to this rule.

Psychological studies of expert-novice differences (deGroot, 1965, Simon and Chase, 1973) have found that experts and novices use similar search processes but different evaluation functions. Because of experience, experts recognize the best moves to examine and make more accurate evaluations of the outcomes.

Because most research has focused on the reasoning process, evaluation functions are usually developed on the basis of intuitions without formal analysis of reliability or validity. Most evaluation functions follow the same general format Shannon described in 1949: they are linear combinations of individual features. Ackley and Berliner (1983) detail some of the weaknesses of this simple approach. These weaknesses include the blemish effect which is an artifact of non-continuous functions, and the boundary effect which occurs at the extreme values of a function.

Wilkins' (1979) research on chess and Ackley and Berliner's (1983) on backgammon represent two significant attempts to analyze the judgment process. Our research involved the game of Othello which is the second most popular board game in Japan. The game is played on an 8 by 8 board of uniform color. The pieces are round disks which are white on one side and black on the other. Figure 1a shows the starting position. Black always starts the game. A legal move (see figure 1) consists of placing a disk on the board so that the disk captures at least one of the opponent's disks. To capture a disk, the moving player's new disk must sandwich one or more of the opponent's disks between it and at least one of the moving player's other disks without any empty squares between them. All disks so sandwiched are captured and are turned over to reveal the mover's color. If a player does not have a legal move, then it becomes the opponent's turn. The players alternate turns until neither player has a legal move. The winner is the player with the most pieces at the end of the game. Unlike chess, the final difference in material (the number of pieces for each player) contributes to player rankings.

For games such as Othello or chess, knowledge of the impor-

tant features is not sufficient to make accurate judgments. The validity of some features depends on the value of other features or on the stage of the game (e.g. early, middle, and late). A good judgment process must be sensitive to these dependencies. Ackley and Berliner (1983) used the term application coefficient to refer to weighting functions which correct for these interactions between features and context. Our research provides empirical evidence that these weighting functions should be sensitive to not only the stage of game but also the skill of the players.

Procedure

There were four major steps in our study of feature importance: 1) the identification of relevant features; 2) the use of the features to evaluate positions from games between experts and games between novices; 3) the development of an "omniscient" evaluation for each position (the external criterion); and 4) the correlation of the individual features with the external criterion.

We collected 29 features from three Othello programs and from articles by Othello experts. The programs are Odin by Peter Frey, Iago by Paul Rosenbloom (1981), and Brand by Anders Kierulf (1982). Jonathan Cerf, who was the national and world champion Othello player, said of these programs, "In my opinion the top programs...are now equal (if not superior) to the best human players. (p.16, 1981)". Because of the skill of these programs, their features seemed appropriate to this research.

We applied the 29 evaluation features to positions from 135 expert games and 131 novice games. Three positions were used from each game: early (16 pieces on the board), middle (32 pieces), and late (48 pieces). After deleting duplicate positions, there were 401 expert and 393 novice positions.

We considered three sources for the independent, omniscient evaluation of each position: 1) human experts' opinions, 2) program evaluations based on a lookahead search of sufficient depth and knowledge that it could beat most experts (the best search being a complete end-game search), and 3) the actual outcome of each game. Because of difficulties in collecting experts' opinions, the first approach was not used. We did use the actual game outcome and two search estimates -- Odin's 8 ply search evaluation and a complete, end-game search applied to the late position from each game (16 ply).

Results

We correlated each feature with the external criterion. Any position in which a feature did not apply was excluded from the feature's correlation. If there were fewer than 100 positions included in a correlation, then the number is shown in parentheses on the figures. The correlations were derived separately for each game stage and skill level. Therefore, there were six correlations for each feature representing the two skill levels and three game periods.

Overall, for experts, most features' predictiveness increased as the game progressed, but for novices, the predictiveness remained constant or decreased. Comparing the two skill

endencies

levels, most features were better predictors for novices' positions than for experts' positions. The predictiveness of all features was affected by the skill of the players or the stage of the game. Five examples of different effects are discussed below.

Figure 2a presents the correlations for a feature which becomes more predictive as the game progresses. This feature is the number of moves to the squares immediately adjacent to the corner. Because the corners are the most important squares on the board and because a square can only be captured if the opponent has a piece next to it, players usually avoid moves to the squares adjacent to corners. Accordingly, most programs negatively weight these moves. Contrary to expectation, our results data show that the ability (not necessarily the action) to capture these squares is positively correlated with success and that this relationship becomes stronger as the game progresses.

Figure 2b shows a feature whose correlation increases as the game progresses and is more positive for novices than for experts. The feature is the number of edge pieces which cannot be immediately captured. Edge pieces are considered important because they cannot be captured in as many directions as the rest of the pieces. Several Othello programs positively weight this feature. As figure 2b shows, this feature positively relates to position strength only for novices' late game positions. For experts' positions early in the game, the feature is a significant, negative predictor, but as the game progresses, the correlation increases to zero. For novices, the feature is a significant, negative predictor early in the game and crosses over the zero correlation to become a significant, positive predictor late in the game. Throughout the game, the correlation is more negative for experts than for novices.

Figure 2c shows the correlations for a feature which is more positively predictive for novices than for experts. The feature is the number of pieces which can never be captured. This feature is important because the goal in Othello is to have the most pieces at the end of the game. As figure 2c shows, the number of uncapturable pieces is more predictive in novices's games than in experts' games. In games between experts, there were no uncapturable pieces until late in the game and then the correlations were not as large as those for novices' games. It is interesting to note that when uncapturable pieces were present early in the novices' game, the player who had the uncapturable pieces won the game.

Figure 3a demonstrates an interaction between player skill and game period. The feature is the total number of pieces for each player. Most novices seem to believe that maximizing pieces is a good strategy. As figure 3a shows, the number of pieces is negatively correlated with the strength of a position. At the beginning of the game, the number of pieces is only predictive for novices. In the middle of the game, the number of pieces is predictive for both groups of players, and at the end of the game, the number of pieces is predictive only for for experts.

Figure 3b demonstrates a different type of interaction between skill level and stage of game. The feature is the number of pieces occupying the edge squares immediately adjacent to an

empty corner. This feature differs from the feature in figure 2a in that this feature counts pieces rather than moves, ignores the squares around occupied corners, and ignores the square diagonally adjacent to the corners. The intuition behind this feature is that these squares provide a method of attack for the opponent to capture the corner. For novices, this feature is always important, but for experts its importance decreases sharply as the game progresses. An intuitive explanation for this difference between skill levels is that an expert knows when placing a piece on one of these squares is not dangerous but a novice does not.

Conclusion

Our data demonstrate that the skill of players and the stage of game influence the ability of features to evaluate positions. A plausible explanation for this effect is that the feature definitions do not adequately account for exceptions. One method to deal with these exceptions is to use the application coefficients proposed by Ackley and Berliner (1983). The correlations provide the data necessary to determine which features need application coefficients and how to construct the coefficients. A second approach is to make the feature definitions more sophisticated by adding rules to handle exceptions. This approach requires greater knowledge of the game.

Some of the features we analyzed tried to account for exceptions. For most of these features, the extra rules resulted in problems with boundary and blemish effects. In addition, because most of the added sophistication is based on intuitions, the simpler features were often better predictors than the complex ones. For example, in addition to the feature depicted in figure 3b, we had a feature which looked for over 30 types of patterns which might occur around a corner to estimate how likely it was that a player would capture the corner. This feature was less effective than the simpler count of occupied edge squares next to empty corners (figure 3b).

We are not trying to imply that features should not be made more sensitive to exceptions by adding sophistication, but to point out the difficulty of properly accounting for all the situational variables which can influence the validity of a feature. Using correlations provides the information needed to determine which features should be made more sophisticated and for testing the changes. Even if a feature cannot be made sensitive to exceptions, application coefficients can improve the average predictiveness of the feature.

References

Ackley, D. & Hans Berliner. The QBKG System: Knowledge Representation for Producing and Explaining Judgments, Carnegie-Mellon Univ., Department of Computer Science, CMU-CS-83-116, March, 1983.

Cerf, Jonathan. Machine vs. Machine, Othello Quarterly, 1981, 3(1), 12-16.

deGroot, Adriaan D. Thought and Choice in Chess, Mouton: The Hague, 1965.

Hasegawa, Goro. How to Win at Othello, Jove Publications: New York, 1977.

Kierulf, Anders. Brand 2.3 -- an Othello program, Informatik ETH Zurich, 1982.

Rosenbloom, P. A World-Championship-Level Othello Program, CMU-CS-81-137, Carnegie Mellon University, Department of Computer Science, August 1981.

Shannon, Claude. Programming a Computer for Playing Chess, Philosophical Magazine, 1950, 41, 256-275.

Simon, H. & W. Chase. Skill in Chess, American Scientist, 1973, 61(4), 394-403.

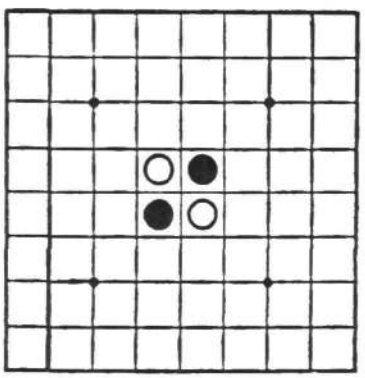
Wilkins, D. Using Patterns and Plans to Solve Problems and Control Search, Stanford Artificial Intelligence Laboratory Memo AIM-329, July 1979.

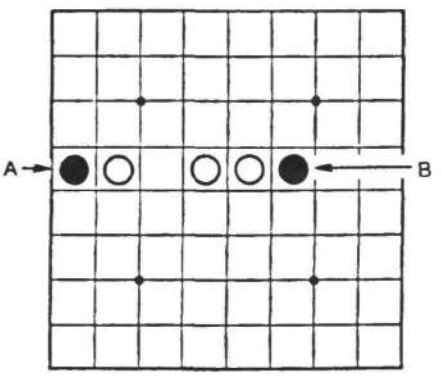
Figure Captions

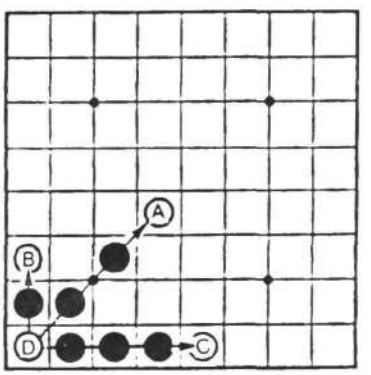
Figure 1. a) The starting position in Othello. b) an example of a legal move -- a move to D (the corner) would capture all the squares which the arrows pass through, c) an example of an illegal move -- black cannot move to A because there are empty squares between A and B. (Note: Pieces are left out of b and c to make the figures easier to read. These positions are not possible in a game. From, Hasegawa, 1977)

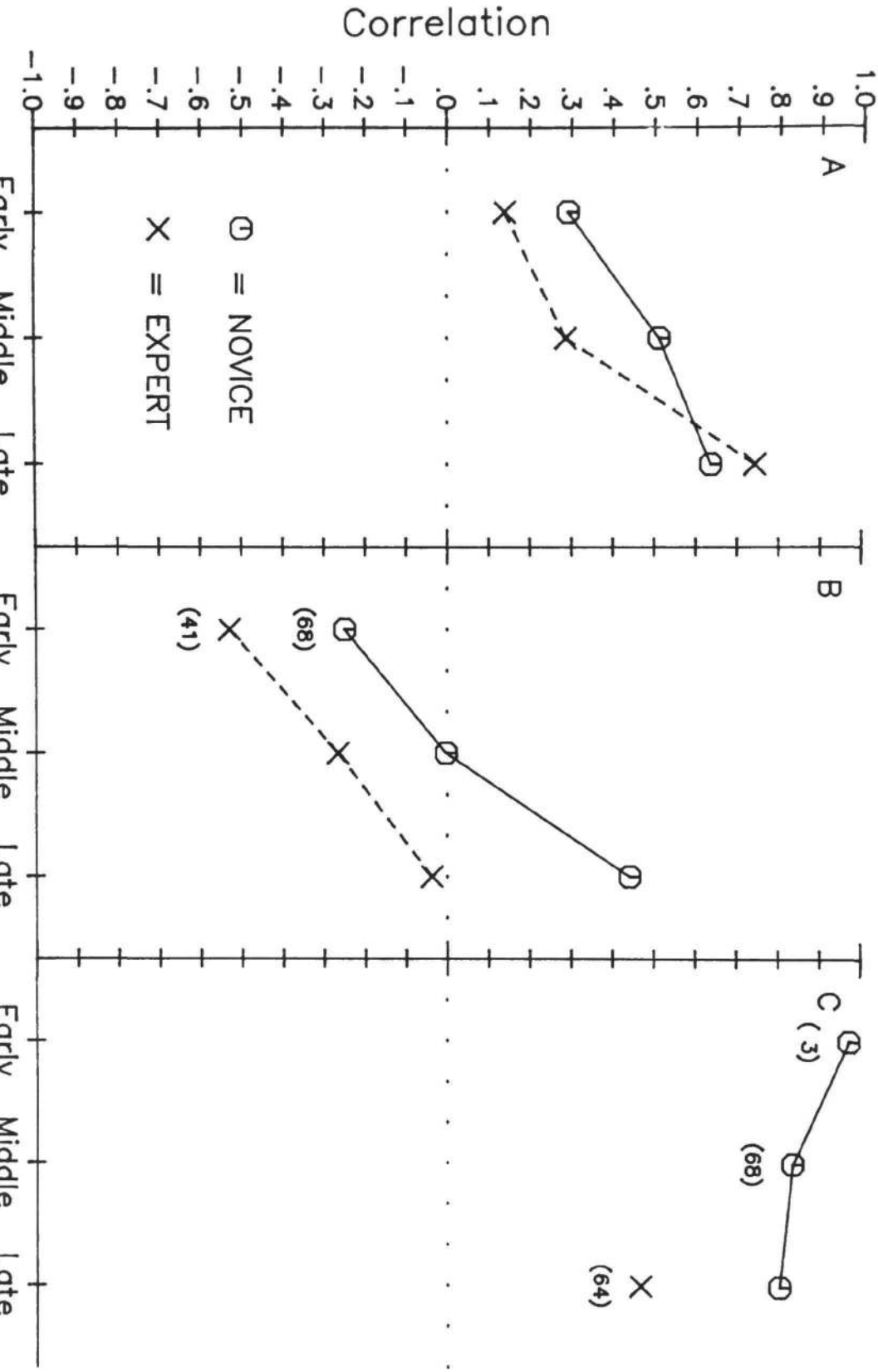
Figure 2. Correlation of evaluation features with an external measure of position strength. a) Stage of game effect -- number of legal moves to squares next to corners, b) Stage of game and skill level effect -- number of edge pieces, c) Skill level effect -- number of pieces which can never be captured.

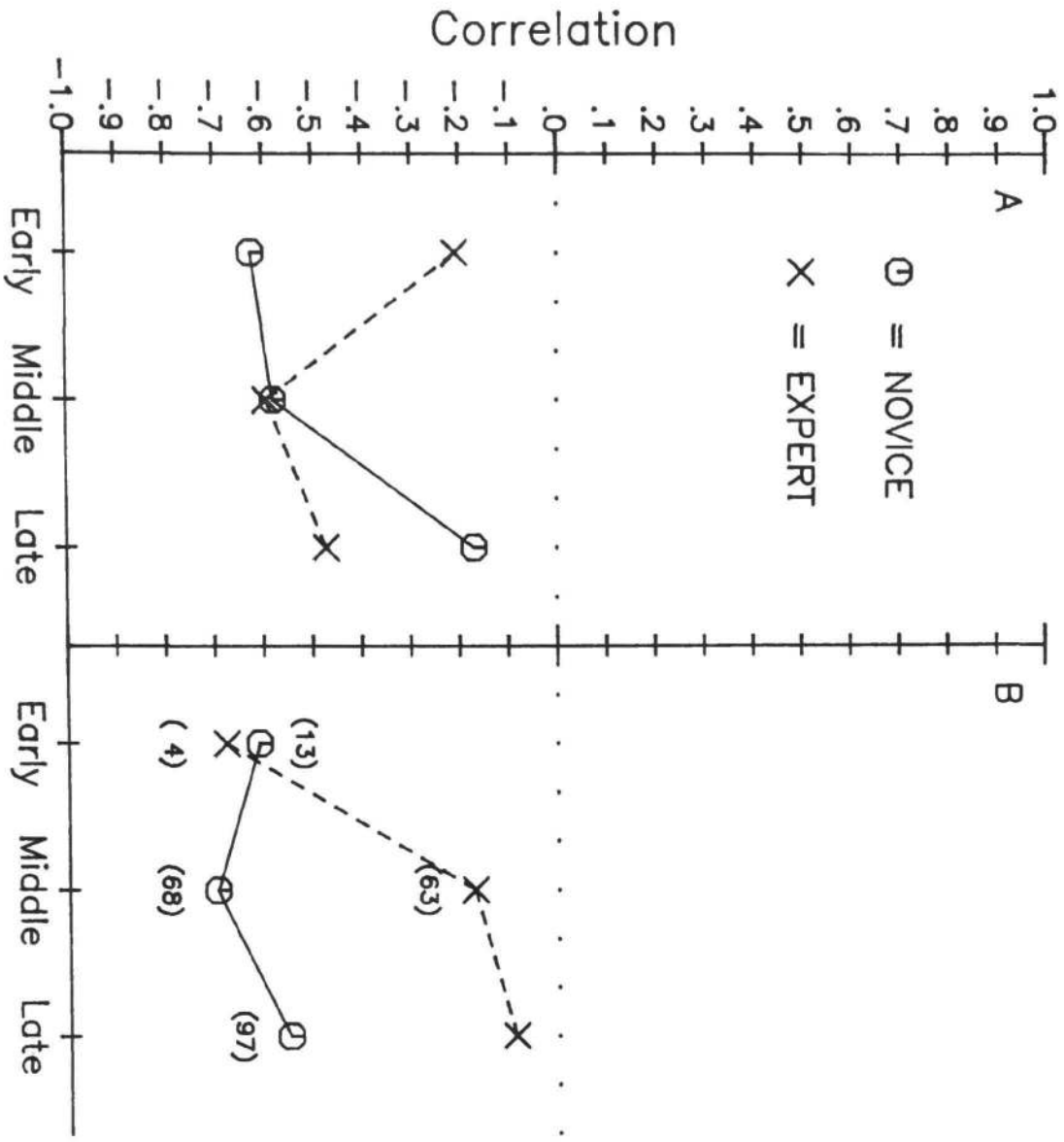
Figure 3. Correlations demonstrating interactions between skill level and stage of game. a) number of pieces for each player, b) number of pieces on edge squares next to empty corners.




A
B


A
B
D
C


Correlation
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
.0
-.1
-.2
-.3
-.4
-.5
-.6
-.7
-.8
-.9
-1.0
A
B
C
Early
Middle
Late
Early
Middle
Late
Early
Middle
Late
O = NOVICE
X = EXPERT
(3)
(68)
(64)
(68)
(41)
STAGE OF GAME


Correlation
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
.0
-.1
-.2
-.3
-.4
-.5
-.6
-.7
-.8
-.9
-1.0
A
B
O = NOVICE
X = EXPERT
Early
Middle
Late
Early
Middle
Late
(13)
(4)
(63)
(68)
(97)
STAGE OF GAME

TOWARD A THEORY OF PROGRAMMING SCHEMES

Jane Terry Nutter
Tulane University

Introduction

People have been writing programs and program documentation for about forty years now. For just as long, people have had to read these program texts. Because programming languages were initially designed for machines, not people, understanding programs presents special problems which persist despite the move toward high level languages and structured programming. With program documentation, the problem is perhaps worse: while programming languages become progressively more "English like", documentation has been moving in the opposite direction. Pseudo code and graphical representations are replacing natural language explanations, suggesting that understanding programs may not be particularly like understanding familiar natural language texts at all.

The fundamentally dynamic nature of programs supports this suggestion. Ordinary natural language texts certainly include dynamic aspects (verbs, for instance!), but nouns provide a stable element which helps anchor meaning. Programs conspicuously lack nouns. A difference of this magnitude must affect how we understand them. Yet understanding program texts must also resemble understanding natural language texts, since learning to deal with programs draws on previously developed skills, some of them reading skills. I believe that understanding in general is guided by abstract concepts, and that program understanding shares this trait. But in the case of program understanding, the concepts in question are abstract patterns for dynamic activities, which I call programming schemes.

Schemes and Representations

A programming scheme is an abstract structure which captures the essential features of a dynamic process to solve some problem. These schemes contain the conceptual information of what a particular chunk of code does. But because they are abstract, they are never directly present in any particular programming text. A particular instantiation in some concrete form is a scheme representation. The underlying patterns (schemes) provide templates by which people understand programs containing embodiments of them (scheme representations).

Researchers in natural language understanding have long believed that abstract cognitive patterns guide text understanding. Schank and Abelson (1977) hold that scripts govern story understanding. Schank (1982) proposes that memory organization packets (MOPS) and thematic organization points (TOPS) underlie memory and cognition. Similarly Chase and Simon (1973), Shneiderman (1976) and Adelson (1981) argue that experts use meaningful abstract formations in their domain of expertise to recall things they were given to memorize.

Researchers have also begun to investigate the role of schemes or something like them in programmer behavior. Soloway has undertaken extensive work in this direction (see e.g. Soloway and Woolf, 1981; Soloway, Ehrlich, Bonar and

Greenspan, 1982; Ehrlich and Soloway, 1982; and Soloway, Ehrlich, and Gold, 1983). Further evidence that abstract constructs govern programming behavior can be found in work by Weiser (1982), Curtis and Sheppard's group (see e.g. Curtis, Sheppard, Milliman, Borst, and Love, 1979 and Sheppard, Milliman, and Love, 1979), and Magel (1982). Yet even Soloway's extensive work leaves schemes themselves largely unanalyzed. We cannot use programming schemes to explain how people understand program texts unless we understand programming schemes. To date, no clear account of what programming schemes are, how they are related to one another, how they are related to their representations, or how they are identified has been given. Hence we have neither a theory to unify these results nor a model for predicting human interactions with particular schemes or their representations.

Why a Theory?

An analysis of programming schemes offers several benefits. First, it should produce a clearer characterization of schemes. To say that a programming scheme is an abstract process template is suggestive, but little more. What properties of programming schemes distinguish them from other abstract concepts and objects?

Second, it should clarify the kinds of possible relationships among schemes. For instance, schemes can contain other schemes: the "merge and sort" scheme contains the "file merge" scheme and the "file sort" scheme. But the "merge and sort" scheme is not an instance of the "file merge" scheme, nor is the "file merge" an instance of "merge and sort". However, the schemes "multiplication by repeated addition", "division by repeated subtraction", and "exponentiation by repeated multiplication" do seem to be instances of a more abstract scheme, namely "perform one operation by repeating another". Hence schemes can be related to one another in at least two ways: containment and instantiation. What other relationships or interactions among schemes need investigating?

Third, the theory should provide insight on how schemes are "tied" to scheme representations. The classes of features which are important can to some extent be identified from an abstract point of view. What follows in the rest of this paper represents a first step in this direction. A theoretical analysis should look further into these and other issues.

Preliminary Issues of Scheme/Representation Relationships

At least four issues arise when considering how schemes are related to their representations. First, what properties of a representation identify it as representing a particular scheme? I call these the identifying features of a scheme representation. Second, what features do people use to identify schemes from representations? These triggers may not be the same as, or even among, the identifying features. Third, what features are readers conscious of when considering code? These objects of attention may be neither identifying features nor triggers. Finally, how effectively does a particular representation reflect its associated scheme? I call this issue representational fidelity.

Understanding texts involves trying to figure out what the author meant. For program texts, this means determining what schemes the author had in mind. Documentation cannot eliminate this need to "look into the author's head",

since it too contains representations, not schemes. Identifying schemes in program texts requires matching features of the representations to possible schemes. And since programmers make mistakes, the representation may not "reflect" the programmer's intent. When it fails to, two questions arise. First, what does the representation represent? Second, and more basically, what does "understanding the text" now mean? If debuggers use schemes to understand "bad" program texts, there must be some independent criteria which link representations with schemes. These criteria are the identifying features of scheme representations.

But we only need independent criteria when something goes wrong. When reading "normal" or "good" texts, there is reason to believe that people use more direct "cues" to recognize schemes. What is required here is an account of how human readers interact with the various features of a representation, identifying or otherwise. (For more on triggers, see Hassell and Nutter, 1984.)

However, triggers need not always be objects of attention. In a directed prestudy experiment (Hassell and Lind, 1983; Hassell, Lind and Rice, 1983), 40% of subjects asked to identify the lines of pseudo code that constituted a "running sum" loop left out the loop construct itself! Thus it appears that readers may not be aware of essential parts of a scheme, i.e., that they take certain parts so much for granted that they do not spontaneously recall them when asked to do so. But in associated one-on-one studies in which subjects observed debugging code were asked to describe out loud what they were doing, the loop construct played a key triggering role. Thus triggers and objects of awareness need not correspond.

While some issues of understandability lie in the psychological realm, some relate to how "good" particular representations are. Informal evidence abounds that some representations reflect their function relatively clearly, while others obscure it with remarkable success. A few studies have compared the effectiveness of particular modes of representation (see e.g. Sheppard, Kruesi, and Curtis, 1981), but we still know almost nothing about which aspects of representations contribute to representational fidelity.

Conclusion

This paper constitutes a proposal for investigations into the nature and role of programming schemes and their representations. Interest in schemes or some similar construct has emerged from a number of different groups and has already motivated substantial empirical work. But without a clearer notion of what schemes are and of how they are related to their representations, these studies must rest on shaky ground. An improved foundational analysis offers enhanced understanding of the role abstract knowledge plays in program understanding, which can then be exploited both to suggest the role of related abstract knowledge in other kinds of understanding and to provide rich new directions for future research.

References

Adelson, B. Problem solving and the development of abstract categories in programming languages, Memory and Cognition v 9, 1981.

Chase, W.C. and H. Simon. Perception in chess, Cognitive Psychology v 4, 1973.

Curtis, B., S.B. Sheppard, P. Milliman, M.A. Borst, and T. Love. Measuring the psychological complexity of software maintenance tasks with the Halstead and McCabe metrics, IEEE Transactions on Software Engineering v 5, 1979.

Ehrlich, K. and E. Soloway. An empirical investigation of the tacit plan knowledge in programing, Human Factors in Computer Systems, Ablex Inc., 1982.

Hassell, J. and J. Lind. Programming plans as advance organizers and their use in improving programmer debugging performance, AEDS Twenty-First Annual Convention Proceedings, Portland, OR, 1983.

Hassell, J., J. Lind, and J. Rice. Using plan knowledge in debugging: an empirical investigation, Tulane University Technical Report 83-102, 1983.

Hassell, J. and J.T. Nutter. Programming schemes: their role in program understanding and maintenance, AEDS Twenty-Second Annual Convention Proceedings, Washington, DC, 1984.

Magel, K. A theory of small program complexity, ACM SIGPLAN Notices v 17, 1982.

Schank, R.C. and R. Abelson. Scripts, Plans, Goals and Understanding, Lawrence Erlbaum Associates (Hillsdale, NJ) 1977.

Schank, R.C. Remembering and memory organization: an introduction to MOPS. In Strategies for Natural Language Processing, W.G. Lehnert and M.H. Ringle, eds., Lawrence Erlbaum Associates (Hillsdale, NJ) 1982.

Sheppard, S.B., P. Milliman, and T. Love. Human factors experiments on modern coding practices, Computer v 12, 1979.

Sheppard, S.B., E. Kruesi, and B. Curtis. The effects of symbology and spatial arrangement of software specifications in a coding task, Proceedings of the Fifth International Conference on Software Engineering, 1981.

Shneiderman, B. Exploratory experiments in programmer behavior, International Journal of Computer and Information Sciences, 1976.

Soloway, E. and B. Woolf. Problems, plans, and programs, ACM SIGCSE Bulletin v 12, 1981.

Soloway, E., K. Ehrlich, J. Bonar, and J. Greenspan. What do novices know about programming? In B. Shneiderman and A. Badre, eds., Directions in Human-Computer Interaction, Ablex Publishing Co., 1982.

Soloway, E., K. Ehrlich, and E. Gold. Reading a program is like reading a story (well, almost), Proceedings of the Fifth Annual Conference of the Cognitive Science Society, 1983.

Weiser, M. Programmers use slices when debugging, Communications of the ACM v 25, 1982.

Attentional heuristics in human thinking

Stellan Ohlsson
The Robotics Institute, Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

1. Introduction

Several mechanisms have been proposed to explain human thinking, such as heuristic search [4], restructuring [6], deduction [2], etc. In this paper I propose that the *allocation of attention* is the major determinant of the course of human thought processes. Furthermore, I argue that this idea is implied by two known properties of the human cognitive architecture, namely the limited capacity of working memory and the inability to inspect procedural knowledge. It follows that strategies for thinking must contain heuristics for how to allocate attention. In consequence, simulation models of human thinking cannot afford to ignore perceptual interaction with the environment. Several examples of attentional heuristics are discussed.

2. Attention and Architecture

The purpose of this section is to derive the importance of attention from principles about the human cognitive architecture. The following definitions set the stage for the argument. Let us use *cognitive unit* generically to cover concepts, propositions, mental images, hypotheses, frames, or any other declarative representational device that may be needed in a theory of thinking. Similarly, let *cognitive operation* stand for rules of inference, problem solving operators, or any other unit of procedural knowledge. Let a *stimulus unit* be any part of the environment that is represented by a single cognitive unit. Let *working memory* at time t be the set of cognitive units in the problem solver's awareness at time t. The process of (visually) *attending* to a stimulus unit consists of moving the eye to that unit, and creating the corresponding cognitive unit in working memory. Information *integration* is any process in which two or more cognitive units are combined to create a new cognitive unit, as when two or more propositions are used as premises from which a new proposition is inferred.

The first part of the argument makes use of the familiar architectural principle that *working memory can only hold a certain number of cognitive units at any one time*. The cause of this capacitly limitation is not important for present purposes. The present argument requires only that there is some limit on the number of units that can be added to working memory at any one time without loss of existing units. The following sequence of assertions connect working memory capacity with attention:

- For many problems, the given information consists of more units than can be held in working memory simultaneously; therefore, the thinker necessarily attends only to a subset of them at any one time.
- In order to solve a problem, cognitive units must be integrated; a task which requires no information integration is almost by definition not a thinking task.
- In order to be integrated, two units must be in working memory at the same time.
- Since the eye can only focus on one stimulus unit at a time, cognitive units will arrive sequentially in working memory.
- Therefore, the order in which stimulus units are attended determines which units can be integrated, and thereby affects the possibility of solving the problem.

For example, suppose that the integration of units A and B is a necessary step in the solution, and

that *A* is attended to. For integration to occur, *B* must become attended before *A* has been forgotten. If the number of units attended to between *A* and *B* is too large, *A* may be lost before *B* arrives in working memory, preventing or delaying the solution. If the problem consists of, say, 10 units, there are 10!, or more than 3.5 million different orders in which the elements can be attended to, assuming that each element is attended once. (If an element can be attended more than once, the number of different orders is larger.) Unless the thinker has some rules to guide the allocation of attention, he/she can search for a long time before *A* and *B* appear simultaneously in working memory.

In conclusion, the limited capacity of working memory implies that a thinker must have rules for which stimulus units to attend to when solving a particular type of problem, and in which order to attend to them. Such rules will be called ***attentional heuristics***. Examples will be given in the next section.

The next part of the argument makes use of a second familiar architectural principle: *Humans have limited knowledge about their knowledge; in particular, they do not know which cognitive operations they are capable of, nor do they know what the legal conditions or the outputs of those operations are.* In other words, procedural knowledge tends to be ***opaque***. The stock example of our inability to inspect procedural knowledge is our lack of insight into the grammatical knowledge that we use in understanding natural language. The opaqueness of inferential knowledge is confirmed by work on expert systems. Extracting rules from a human expert takes a long time, and requires many revisions of the rule set. Experts have little explicit knowledge of the inferential rules they possess.

The opaqueness of procedural knowledge has several effects. First, the thinker cannot anticipate what the result would be of applying a particular cognitive operation without executing it. Being unable to retrieve the operation and inspect its "code", the thinker cannot reason at a meta-level about the operation. Second, since the thinker does not know which cognitive operations he/she has available, he/she cannot choose to execute a particular operation. A system with opaque knowledge must evoke its operations in a data-driven fashion, each operation keeping a look-out for a cognitive unit which can serve as input, and going into action when one is found. In such a system, the application of operations is controlled by the allocation of attention.

The above argument is summarized in the following *principle of attentional control of thinking*:

- A human thinker cannot intentionally apply cognitive operations, he/she can only intentionally select which stimulus unit to attend to at any one moment in time. A new conclusion, problem solving step, insight, etc, may or may not flow from the attended information. As a result, the thinker's control over his/her cognitive activity is indirect: which operations are applied and in what order is a function of the attended information. Success in thinking requires that units are attended in such an order that the necessary integrations can occur.

According to this principle, skill in thinking consists in knowing what to look at, and when to look at it. The next section gives examples of heuristics which encode such knowledge.

3. Examples of attentional heuristics

3.1. Verbal reasoning

In an analysis of 60 think-aloud protocols from a verbal reasoning task with spatial content, Ohlsson [5] found that out of 2520 identifiable problem solving steps, 1255 or 47 % were steps in which a premise was read from the problem text. The following ten attentional heuristics were postulated to account for such steps:

1. Begin solving the problem by reading the first premise.
2. Begin by reading the question.
3. When all premises have been processed, then read the question.
4. When there is nothing else to do, read from the problem text.

5. When a new conclusion has been arrived, read from the problem text.
6. If the last premise read was premise *N* , then read premise *N + 1*.
7. Read a premise which has not been read yet.
8. Select an object about which an inference recently was made, and read a premise which mentions that object.
9. Select an object about which no inference has yet been made, and read a premise which mentions that object.
10. Select an object that is remembered as interesting, and read a premise which mentions that object.

The heuristics fall into three distinct classes: rules 1-3 deal with the first and last acts of reading, rules 4 and 5 determine *when* it is appropriate to read (rather than do something else), and rules 6-10 specify *what* to read.

Each subjects was modelled by a different subset of these attentional heuristics. For example, the behavior of one subject was modelled well by heuristics 1, 5, and 6, while another subject seemed to behave according to rules 2, 4, 7, and 8. In short, interindividual differences in the allocation of attention were clearly visible in the protocols.

3.2. Classical mechanics

In a study of problem solving in classical mechanics, Larkin and co-workers [3] found that a major difference between novice and expert problem solving was the order in which the equations of a problem were used. Novices tended to work backwards, from the desired quantity to the given ones, while experts tended to use the equations in a forward search fashion, going from the given to the desired quantitites. Re-formulating these strategies as attentional heuristics, we have:

1. Attend to equations which contain many known quantities.
2. Attend to equations containing the desired quantity.

The similarity between these two heuristics and heuristics 8 and 9 in the previous example should be noted. Rule 1 (and 8 above) represent a *chaining* tactic: having processed an object (premise, variable, etc.), the thinker looks for further information about that object; in so doing, he/she encounters other objects which are then processed; etc. Rule 2 (and 9 above) represent a *missing part* tactic: look for objects about which nothing is known yet. These heuristics seem to be very general. Also, they nicely illustrate the nature of heuristics: both are useful, although they give contradictory advice.

3.3. Other examples

In a simulation study of chess perception, Simon and Barenfeld [8] found that the eye-movements of chess players could be predicted from the number and character of the chess-relations entered into by a particular chess piece. The attentional heuristic of the players could be formulated as "look at pieces which enter into many important chess relations".

In his studies of children's conservation of physical quantitites, Piaget [1] hypothesized that children fail to conserve because they do not coordinate compensating changes in objects. The attentional heuristic non-conserving children lack might be formulated as "attend to all changed dimensions of the object to be judged".

In a study of problem solving in geometry, Ohlsson [7] found that success in finding proofs was dependent on attending to the right geometric objects. When confronted with a figure which contained several different triangles, the subjects had no heuristics for which triangle to attend to. However, one subject had the heuristic "when stuck, try to discover new geometric objects in the figure".

4. Discussion

To restate the main idea, success in thinking is often a matter of knowing where to look; once the right subset of the available information is attended to, it is often self-evident what step to take or which conclusion to draw. Conversely, if the right information is not attended to, achieving the solution may be impossible. Human thought is therefore mainly governed by attentional heuristics.

The validity of this conclusion can be expected to vary across task domains. In domains where all relevant problem information can be kept in mind simultaneously, attentional heuristics become less important, because no selection is involved. Similarly, in task domains where the thinker has explicit representations of his/her operations (eg mathematics), the latter need not be invoked in a data-driven fashion. On the other hand, in domains with information-rich displays and large amounts of irregular, intuitive, and informally acquired inferential knowledge, attentional heuristics can be expected to be the major determinant of behavior.

Attention allocation is not proposed here as an *alternative* to other mechanisms of thought. The theory of thinking certainly has to make room for mechanisms such as heuristic search, restructuring, deduction, analogy, etc. But such mechanisms are dominated by attention in the sense that they operate upon attended information, and they can only succeed to the extent that the right information has been attended.

Psychological theories often treat thinking separate from the perceptual-motor interaction with the problem situation. Simulation models of thinking usually assume that the problem has been encoded, and that all given information is available in working memory. But in task domains where attention allocation is the major determinant of thinking, simulation models cannot ignore perceptual interaction with the environment without ignoring a major part of the behavior to be simulated.

References

[1] Flavell, J. H. *The developmental psychology of Jean Piaget*. Princeton, NJ: Van Nostrand, 1963.

[2] Hagert, G. & Hanson, A. Logic modelling of cognitive reasoning. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, 1983.

[3] Larkin, J. H., McDermott, J., Simon, D. P., & Simon, H. A. Models of competence in solving physics problems.*Cognitive Science*, 1980, p(4), 317-345.

[4] Newell, A. & Simon, H. A. *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall, 1972.

[5] Ohlsson, S. Problem solving strategies in a spatial reasoning task. Uppsala Psychological Reports, No. 340, 1982.

[6] Ohlsson, S. Restructuring revisited. I. Summary and critique of the Gestalt theory of problem solving. *Scandinavian Journal of Psychology*, in press.

[7] Ohlsson, S. Restructuring revisited. III. Re-describing the problem situation as a heuristic in geometric problem solving. Uppsala Psychological Reports, No. 353, 1983.

[8] Simon, H. A. & Barenfeld, M. Information-processing analysis of perceptual processes in problem solving. *Psychological Review*, 1969, *76*, 473-483.

Learning to Program Recursion

Peter L. Pirolli, John R. Anderson, and Robert Farrell

Carnegie-Mellon University

Learning to program recursive functions in languages like LISP is notoriously difficult. Indeed, a primary mark of expertise in such languages is the ability to plan and code recursive functions. Recently, we have performed protocol studies of students learning to program recursion in LISP and LOGO as well as controlled experiments on learning recursion in a simple programming language. We have used the GRAPES production system model (Anderson, Farrell & Sauers, 1984) to address these results. GRAPES not only models programming performance but also learning by doing by the mechanism of **knowledge compilation**. Knowledge compilation summarizes extensive problem-solving operations into new compact production rules (see Anderson, Farrell, & Sauers, 1984; Neves & Anderson, 1981 for details).

Characteristics of Learning to Program Recursion

In Anderson, Pirolli and Farrell (1983), we hypothesized that initial performance and learning in recursive programming is primarily driven by learning from examples, since students have little relevant prior knowledge. We further hypothesized that such learning consists of two components. First, students can use the solution to a given example as an outline which must be modified in order to solve a current problem (**probelm-solving by analogy**). Second, learning (knowledge compilation) mechanisms can summarize these analogy operations into new problem-solving operators which can apply to future problems (**learning from analogy**).

To illustrate problem-solving by analogy and learning from analogy we present a GRAPES simulation of a subject, SS, in her initial encounters with coding recursive LISP functions. At the time, SS had about 15 hours of tutoring and programming experience in LISP. The first function that SS wrote was SETDIFF, which returned all the elements of the one list not contained in a second list. In coding SETDIFF, SS analogized from a textbook example function, INTERSECTION1, which returned all elements that were common to two input lists. The primary structure of SETDIFF and INTERSECTION1 consists of a series of if-then conditional statements (see Figure 1).

The GRAPES simulation (Figure 2) of SS when provided with a representation of the INTERSECTION1 code at multiple levels of abstraction, a specification of the SETDIFF relation and a goal to code SETDIFF used following analogy production:

```
P1: IF the goal is to write a function
       and there is a previous example
    THEN set as subgoals
         1) to compare the example to the function
         2) map the example's solution onto the current problem
```

P1 sets goals to first check the similarity of the specifications of INTERSECTION1 and SETDIFF and then map the code structure of INTERSECTION1 onto the SETDIFF code. A set of comparison productions then found that both INTERSECTION1 and SETDIFF take two input sets and can be code recursively.

Next, **structure-mapping** productions map conditional clauses from INTERSECTION1 to SETDIFF. SS gave clear evidence in her protocol of performing the same mapping. For each INTERSECTION1 conditional clause, SS (and GRAPES) mapped the condition of the clause onto SETDIFF and determined what action should take place. Like SS, GRAPES fluctuated the level of abstraction at which these conditional clauses were mapped. For instance, initially GRAPES attempted to map the condition "test if an element should be added to the result" When this

failed, it mapped a more literal translation: "test if the first element of the first list is a member of the second list".

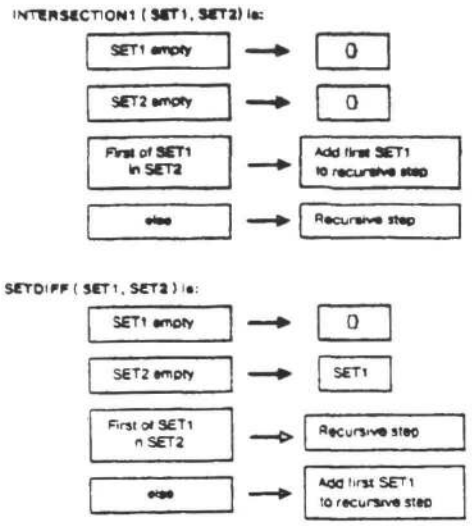


Figure 1: Abstract code specifications

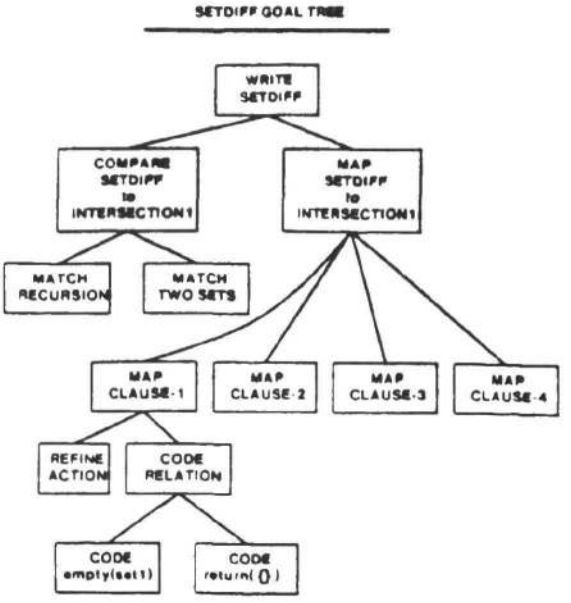


Figure 2: Part of the GRAPES solution for SETDIFF

A number of new GRAPES productions were produced by knowledge compilation in coding SETDIFF. However of primary interest is a rule which was produced by compiling the analogy processes:

```
C1:  IF the goal is to code
          a recursive relation on two sets SET1 and SET2
     THEN code a conditional and set as subgoals to
        1) Refine & code a clause
           to deal with the case when SET1 is empty
        2) Refine & code a clause
           to deal with the case when SET2 is empty
        3) Refine & code a clause
           to deal with the case when the first
           element of SET1 is a member of SET2
        4) Refine & code a clause
           to deal with the else case
```

In the SETDIFF example we see both problem-solving by analogy and learning by analogy. First, the solution of SETDIFF was heavily guided by the INTERSECTION1 example and second we see the acquisition of a new production, C1, summarizing this analogy process. This operator can only be successfully applied to a small domain of recursive functions. Both GRAPES and SS successfully used this production on the next function attempted, SUBSET, which determines whether one list is a subset of another by recursion. However, both SS and GRAPES had great difficulty in solving the next recursive function in the instructional series, POWERSET, which computes the set of all subsets of a set. This difficulty results from the inapplicability of production C1 in the POWERSET case. The analogy processes executed in SETDIFF were at a sufficiently abstract level that they generalized to SUBSET when compiled into C1. However

they were not abstract enough to generalize to POWERSET. This leads to the conjecture that transfer of learning from analogy is limited by the level of abstraction at which the analogy is initially carried out.

```
TO TUNNEL :X
SQUARE :X
IF :X = 50 THEN STOP
TUNNEL :X + 10
```

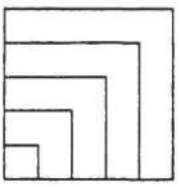


Figure 3: The TUNNEL function

```
TO CIRCLES :X
RCIRCLE :X
IF :X = 50 THEN STOP
CIRCLES :X + 10
```

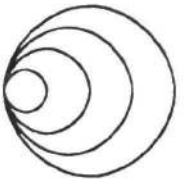


Figure 4: CIRCLES function

A further illustration of how the level of abstraction of an analogy impacts on generalization comes from a protocol of an eight year old student (J) coding her first recursive program in LOGO. The student's background consisted of a semester and a half of weekly LOGO lessons. J's first problem was a function, TUNNEL, which drew concentric squares on the computer screen (Figure 3). Her coding was guided by an example function CIRCLES which drew concentric circles (Figure 4). Unlike subject SS who basically mapped the conditional structure of INTERSECTION1 onto SETDIFF, subject J basically mapped the actual code of CIRCLES onto TUNNEL. The only portions of the CIRCLES code that J did not map at the literal level was the name of the function (CIRCLES changed to TUNNEL) and a subfunction called by the program (CIRCLES calls a circle drawing program while TUNNEL calls a square drawing program. The production rule produced by GRAPES by compilation of the CIRCLES - TUNNEL analogy is:

```
C2:  IF the goal is to code a figure
        and the function for figure is called <NAME>
        and a repeated subfigure of the figure is coded by <SUBFUNCTION>
     THEN write
          TO <NAME> :X
          <SUBFUNCTION> :X
          IF :X = 42 THEN STOP
          <NAME> :X + 10
```

In contrast to production C1, the condition of production C2 matches to relatively spurious program specifications and its actions largely specify literal code rather than an abstract plan. The fact that subject J had great difficulty writing subsequent recursive programs seems to corroborate the view that she had failed to learn any operators of even limited generality for coding recursion from this analogy episode.

The GRAPES simulations of programming recursion also suggest that learning to program recursion occurs by a process of piecemeal approximation. For example, the operator producing code for a tail-recursive call is compiled when coding SETDIFF, while an operator for combining the result of a recursive call with another result is not learned until GRAPES codes POWERSET. The protocols of subject SS support this analysis: after coding SETDIFF, SS has no problem coding a tail-recursive call in SUBSET, but does not initially know how to combine results of recursive calls with other results in POWERSET. However, after coding POWERSET, SS successfully wrote code combining recursive results with other results.

Conceptual Models of Recursion

Our protocols of LISP subjects suggest that novices typically view recursive functions with a "flow of control" mental model (see also Kahney, 1982). In attempting to plan code for a recursive function, subjects will frequently simulate or trace the flow of control of the function and its recursive calls. Using such a model generally leads to errors because such tracing is

often complicated and involves keeping track of many partial results. In addition, such a model does not readily map onto program generation since it is not specifically a model of how to write a recursive function: the flow of control model describes how already written functions are evaluated.

We have modelled in GRAPES what is arguably an ideal strategy for coding recursive functions. This strategy has the following "formal model" of code generation for recursive programs: "A recursive function consists of (a) one or more terminating cases in which a simple answer is returned and (b) one or more recursive cases in which the answer to the current problem is solved by assuming that the answer to a simpler version of the same problem (a recursive call) has been solved."

We hypothesized that subjects instructed with the formal model would learn to code recursive functions faster than those using the flow of control model. We conducted an experiment in which subjects initially learned the basic functions, predicates, conditionals, and definitional forms of a simple programming language modelled after LISP. One group of subjects was then introduced to recursive functions using the formal model, another group's instructions for recursion emphasized the flow of control model. All subjects were then presented with four recursive program specifications (one at a time) for which they had to write code. Subjects were re-presented with specifications until they had written a correct program for each specification. The formal model group took significantly less time (**M** = 3444 seconds) than the flow of control group (**M** = 5116 seconds) to code all functions correctly.

Implications for Intelligent Tutoring Systems

Our efforts to design intelligent computer-aided instruction (ICAI) system for programming (and especially recursion) has been influenced by the current results. First, our ICAI system avoids examples since our GRAPES model suggests that although problem-solving by analogy can facilitate initial performance it may not necessarily facilitate learning. Second, our model suggests that students learn recursion in a piecemeal fashion. The ICAI system thus presents a wide variety of recursion problems to expose students to a large range of coding patterns. Finally, our GRAPES ideal model and experimental results indicate that a formal mental model of recursive programs facilitates learning because it reduces working memory load and more directly maps onto program generation than the flow of control model typically used by novices. This conceptual model and the ideal programming model are employed by the ICAI system in teaching recursion.

References

Anderson, J.R., Farrell, R., & Sauers. (1984). Learning to program in LISP. **Cognitive Science** in press.

Anderson, J.R., Pirolli, P.L., & Farrell, R. (1983, October). **Learning to program recursive functions**. Paper presented at the Conference on Expertise, University of Pittsburgh, Pittsburgh, PA.

Kahney, H. (1982). **An in-depth study of the cognitive behaviour of novice programmers**. (Report No. 5). Milton Keynes, England: The Open University, Human Cognition Research Laboratory.

Neves, D.M. & Anderson, J.R. (1981). Knowledge compilation: Mechanisms for the automization of cognitive skill. In J.R. Anderson (Ed.), **Cognitive skills and their acquisition** (pp. 57-84). Hillsdale, New Jersey: Lawrence Erlbaum Associates.

Acknowledgements: This research was supported by an IBM Fellowship to Peter Pirolli and Office of Naval Research grant No. N00014-81-C-0335 to John Anderson.

A MODEL OF KNOWLEDGE REPRESENTATION BASED ON DEONTIC MODAL LOGIC

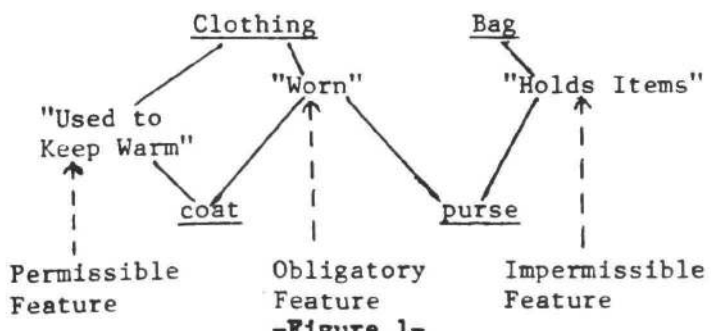
Anthony Rifkin
Developmental Psychology Program
City University of New York Graduate Center
33 West 42nd Street
New York, NY 10036

In this paper a model of knowledge representation is presented along with psychological research supporting this model. This is followed by a general discussion of the model and its possible application to the construction of computer knowledge bases.

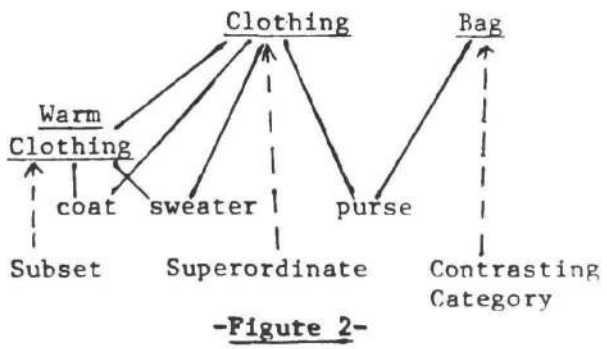
Psychological research has demonstrated that people do not use necessary and sufficient features to determine the membership of instances in natural categories (Hampton, 1979; Rosch and Mervis, 1975). In place of necessary features, or a set of features, that are common to all members of a category, Rosch and Mervis (1975) proposed that category instances bear a "family resemblance" to each other. In support of this proposal they found that the degree of membership (or "typicality") of instances is related to the number of features instances share with other members of that category, and conversely, that instances that are more typical of a category share few features with (bear less "family resemblance" to) members of other categories.

An alternative model was constructed for the present research, using a deontic modal logic (Wright, 1963). While this model can deal with the phenomena of typicality gradients of membership and the family resemblance structuring of category features, it at the same time shows how features can be taken to be common to all members of a category and defining of these memberships. This is done within the framework of extensive cross-classifications and greater taxonomic depth than has commonly been examined in category research. It is also proposed that this hierarchical and cross-classified organization of categories and featural information is accessed in terms of least-upper-bound shared memberships, when retrieving information relevant to the comparison of instances.

In the proposed model, the deontic features are taken as "Obligatory", "Permissible", or "Impermissible" in defining the membership of instances (see Figure 1 for examples for the category Clothing). The relations of these features to the categories can be expressed in terms of quantification and the "ideal" worlds they represent. Obligatory features may be taken as defining the membership of "all" instances, Permissible features as defining the membership of "some" instances, and Impermissible features as defining the membership of "none" of the instances. In contrast to necessary features, Obligatory features are generic norms that can be taken as common to all members, but are not invalidated by individual cases. For example, the feature "Worn" can be applied to all instances of Clothing, including a least typical instance like purse (see Figure 1). Or, this feature may not be applied to the instance purse, when purse is being used as an instance of the category Bag in contrast to the category Clothing and the feature "Holds Items" is being applied to it. In this way the sense of deontic features can be shifted (delimited or extended) to handle the "fuzzy" boundaries of natural categories and to adjust to the comparisons being made of categories and/or instances.



-Figure 1-



-Figure 2-

In relation to the "family resemblance" organization of features (Rosch and Mervis, 1975), more typical instances (e.g., coat in Clothing) would have Permissible as well as Obligatory features defining their membership in the category (e.g., "Used to Keep Warm" as well as "Worn"), and less typical instances (e.g., purse in Clothing) should have only Obligatory features defining their membership in the category and Impermissible features that define their membership in contrasting categories. As well, Obligatory features should vary in how they are applied to least typical instances (as in the "Worn" and purse example above), as these instances don't have Permissible feature definitions, and do have strong (Impermissible feature) connections to contrasting categories. Greater family resemblance would therefore be based on the sharing of Permissible as well as Obligatory features, and less family resemblance would be based on sharing only the Obligatory features and sharing Impermissible features with members of contrasting categories.

The organization of Obligatory, Permissible and Impermissible features should also be reflected in the taxonomic depth of categorization, as well as in cross-classifications. Where the least typical instances should have more memberships in contrasting categories (as Rosch and Mervis, 1975, found), most typical instances should belong to more subsets of the categories (see Figure 2). For example, coat should belong to the subset Warm Clothing under Clothing, because of the Permissible feature "Used to Keep Warm". This would be so because the membership of most typical instances can be defined by Permissible features, and these Permissible features in turn delimit salient subsets within the (superordinate) categories. As well, this subset level of taxonomization should be accessed before the higher, superordinate categories (in terms of least-upper-bound shared memberships), when comparing two instances that share membership in one of these subsets (e.g., coat and sweater in the subset Warm Clothing; see Figure 2). These subsets constitute part of the greater taxonomic depth that has not been traditionally studied in category research.

Experimental Procedures. All of the subjects were undergraduates in lower-division psychology classes, and all of the experimental tasks were paper and pencil tests. Each of the experiments had two tasks. In the first task of Experiment 1, subjects were asked for three categories for "most" and "least" typical instances from 4 superordinate categories, Tools, Clothing, Furniture, and Vehicles. The "most" typical instances were those instances that had received an average typicality rating of 2 or less in these categories (on a 7-point scale) in Rosch's (1975) norms, and the "least" typical instances received an average rating of 4.5 or greater. In the first task of Experiment 2, subjects were asked why most and least typical instances, and most-least typical instance pairs are members of the superordinate categories they had initially been taken from, and why the single instances are members of other categories frequently listed for them in Experiment 1. In the first task of Experiment 3, subjects were asked what is the same and different about most-most, most-least, and least-least typical instance pairs. An equal number of these different types of instance pairs were taken from three levels of categorization (as determined in Experiment 1). They either 1) shared membership in the same subset or contrasting category and the same superordinate (e.g., coat and sweater), 2) shared membership in a superordinate, but did not share membership in a subset or contrasting category (e.g., coat and pants), or 3) came from different superordinates and different contrasting categories (e.g., coat and car).

In the second task for each of the experiments, subjects were asked to compare the sets of the elicited categories and features to those of the superordinate categories (e.g., "Are all things that are worn types of clothing?" and "Are all types of clothing things that are worn?"). If they gave a "no" response, they were asked to list the exceptions (e.g., things that are worn that are not clothing).

Results. In Experiment 1, the most typical instances elicited the superordinate categories more often than the least typical instances did, both as dominant (most frequently listed) categories, $t(6) = 4.39$, p .01, and in terms of any listing of these

categories for the instances, $t(6) = 3.25$, $p < .02$. Of the other categories listed for the instances, 2 types were identified in Task 2, 1) "Contrasting" categories that have overlapping memberships with the superordinates, and 2) "Subset" categories that are subsumed within the superordinates. Comparing the most frequently listed categories in Task 1 (excluding the superordinates), most typical instances elicited a proposrtionally greater number of Subset categories, and least typical instances elicited a greater number of Contrasting categories, $X^2(1) = 6.19$, $p < .02$.

In Experiment 2, the features given as defining the membership of most and least typical instances in Task 1 were identified in Task 2 as either Obligatory (e.g., "Are all types of clothing worn?" receiving "yes" responses), Permissible (e.g., "Are all types of clothing used to keep warm?" receiving "no" responses), or Impermissible (features given as defining membership in contrasting categories that were not given as defining in the superordinate). For each of the superordinates in Task 1, Obligatory features were the most frequently received features defining membership across most and least typical instances, and for the most-least typical instance pairs. No difference was found between most and least typical instances in the frequency with which they elicited the Obligatory features, $t(6) = .22$, n.s.. On the other hand, most typical instances elicited Permissible features as defining their membership more often than least typical instances did, $t(6) = 3.99$, $p < .01$. As well, most typical instances elicited more Permissible features (of the superordinates) as defining their membership in the Contrasting categories they belong to, while least typical instances elicited proportionally more Impermissible features (of the superordinates) as defining their membership in the Contrasting categories, $X^2(1) = 5.21$, $p < .05$.

In Experiment 3, the categories and features most frequently received as responses for what is the same and different about instances corresponded to the least-upper-bound level of categorization instances shared membership in (i.e., subsets, superordinates, or higher features encompassing 2 or more superordinates), $X^2(1) = 20.04$, $p < .001$. For "what is the same" about the instance pairs, the least-upper-bound categories the instances shared membership in (or the features corresponding to these categories) were elicited (e.g., gloves and coat elicited "you wear to keep warm", and jackets and pants elicited "Clothing" and "they are worn"). For "what is different" about the instance pairs, features and categories of the level immediately below the least-upper-bound category were received (e.g., two subset distinctions for instances that share membership in a superordinate, but not in a subset, such as jackets and pants eliciting "one is worn on the upper body and the other is worn lower"). Two more taxonomic levels were identified in this experiment as well. One was a level above the superordinates generally corresponding to "Functional Artifacts" (e.g., in responses such as "Man-made" and "Used by People" received for what is the same about instances belonging to different superordinates, such as shirt and bus). The other was a level below the subsets, and was found in responses to what is different about two instances of the same subset (e.g., for the subset Seats under Furniture, the instances sofa and chair received "many people sit on one and only one person sits on the other").

In Experiment 3, evidence was also found for shifts in (delimiting or extending) the sense of a feature according to the comparison being made. A significant proportion of instance pairs were found to elicit the defining features both for what is the same about them and for what is different about them (e.g., receiving "they are worn" for what is the same about coat and purse, and receiving "one is worn and the other holds items" for what is different about this pair), $X^2(1) = 4.0$, $p < .05$. In each of these cases, the less typical instance was excluded when the contrast of what is different was made.

Conclusions. This research shows that people use a system of cross-classification and multiple taxonomic levels of categorization in their representation and retrieval of object information. Evidence was also found for this organization of natural category information being based on the use of deontic features. Obligatory features (of the superordinates) were most frequently received as defining the membership of instances.

They were also taken to be common to all members of these categories and were the most frequently elicited features when comparing superordinate members that did not share a common subset. The Permissible features were found to correspond to subsets within the superordinates, and as well were used to define the membership of instances within the superordinates. The Impermissible features on the other hand were not used to define membership in the superordinates, but were used to define membership in contrasting categories and distinguish why least typical instances are different from most typical instances.

This system of categorization and the use of deontic features may well explain "family resemblance" structuring (Rosch and Mervis, 1975). The greater "family resemblance" of most typical instances to other members of a category may come from the greater number of Permissible features which define their membership (these features being used to define their membership in contrasting categories as well), and the greater number of Subset categories they belong to. Least typical instances may bear less resemblance to other members because 1) their membership is defined primarily by Obligatory features, 2) they are members of more contrasting categories, and 3) their membership in these contrasting categories is defined by Impermissible features.

In contrast to Rosch and Mervis's (1975) "family resemblance" model, Obligatory features are taken to be common to all members, and to define the membership of instances. These features are not necessary features however, as they can be said to be applicable and not applicable to an instance, depending on the sense they are taken in. As found here, this use of Obligatory features is most evident in their application to least typical instances. As McCloskey and Glucksberg (1978) found, such boundary cases are far more subject to changes in opinion and differences in opinion in the determination of their membership. Obligatory features can therefore be used to define/determine membership of a borderline instance, when extended to a more general sense. They can also be used to exclude an instance when category contrasts are being made, and a more delimited sense of the feature is used.

As to how this extending and delimiting of senses is done can be seen in the following examples. Where an Obligatory feature (such as "Used to Build" for Tools) can be taken to be applicable to all instances of a category in its more general sense, it may also be used in a "delimited" sense (e.g., as synonymous to "Putting (Joining) Things Together", which denotes a subset with sibling relations to "Taking Things Apart", "Making Holes", etc.). The delimited senses of Obligatory features may therefore be constructed through references to Permissible features and subsets (e.g., "Putting Things Together"). The more general senses would come from combining the Permissible feature, subset definitions (e.g., "Putting Things Together" and "Taking Things Apart", etc.). Or, an Obligatory feature may be used in an "extended" sense, the outer limits of extension being metaphor (e.g., with an Obligatory feature for Vehicles, "Gets you from one place to another" being extended to books).

Applications for Computer Knowledge Bases. This model has strong advantages for the programming of natural category information in computer knowledge bases. The present model indicates means for constructing a clearly defined structure of knowledge representation within which each piece of information has a specific location and can be accessed according to it's location. Well-structured, taxonomic inheritance is possible given the specification of cross-classifications and taxonomic levels of information (four general levels having been initially determined in the present research for simple object names). At the same time, typicality gradients of membership and "fuzzy" category boundaries can be handled without having to refer to prototype-related default values.

This power is dependent upon the deontic nature of the featural definitions, rather than categories being "well defined". That is, the deontic features are generic norms which are not invalidated by individual cases, and for which the senses can be shifted. This shifting of senses though is based upon the systematic extending and delimiting of

instance sets. The means for this extension and delimitation was seen in the "Used to Build" example given earlier.

What does appear to be necessary to make this system of taxonomic structuring and inheritance work, is the ability to establish the different category sets and senses through the use of immediate "contexts". This use of "context" can be seen in the findings reported here (Experiment 3) on the accessing of specific taxonomic levels and the delimiting of senses, based on the objects being compared and the type of comparison (same or different) being made. This is similar to Artificial Intelligence work being done on interactive systems (e.g. Grosz's, 1981, "global focus" and Sidner's, 1983, "immediate local focus"). These processes are, of course, dependent on discourse. This appears to be the strength of deontic features however, that is, that they can be interpretted (and re-interpretted) during the negotiation of meanings. At the same time, the "interpretibility" of these features need not undermine the well-structured characteristics of this representation system. As with the "Used to Build" example given earlier, delimiting the sense of this feature to "Putting Things Together" is done by referencing an already-specified subset. An important characteristic of this model is, therefore, that meanings and shifts in senses can be established through reference to quantified instance sets.

The details of programming this system of representation do still need to be worked out. For example, in using the taxonomic inheritance structure of the knowledge representation language KL-ONE, what is true of a concept must be true of all its decendents. Therefore, how a shift in the sense of an Obligatory feature such as "Worn" would effect the sense of a Permissible feature decendent such as "Worn by Women" still needs to be attended to. It is felt though that the present model does hold promise for the programming of computer knowledge bases.

Finally, it should be noted that the "interpretable" nature of deontic features may be based upon a "functional" character of these features. Approximately three quarters of the features elicited in the research reported here were either "functional" features (e.g., "Used", "Used to Build" and "Holds Things") or had functional components (e.g., the feature "Used in Houses" having a functional component and a locational component, "In Houses"). As taken here, "functional" is broadly defined as the functions or purpose of an object, an action performed with it, or an action of an object independent of an agent. This broad definition of "functional features" could also be applied to natural kinds (e.g., "plants grow") and to abstract concepts (as seen in the extensions to metaphorical senses mentioned earlier), though further research is necessary to determine how these features are used with these types of concepts. As well, a number of other types of features had functional correlates. For example, the (structural) feature "Blade" and the (physical) feature "Sharp" have the (functional) correlate "Used to Cut". This use of functional information is in line with Miller and Johnson-Laird's (1976) proposal that functional "schemata" can be used to translate perceptual features into functional conditions. This character of functional features is relevant to the interpretiblity of features in context (Miller, 1976) and extensions to less typical instances. For example, a tree stump may be included as a Table (or a "Thing to Put Things On") in the context of a "picnic".

Miller, G. Practical and lexical knowledge. In E. Rosch & B.B. Lloyd (eds.), Cognition and categorization. Hillsdale, NJ: Erlbaum Associates, 1978.

Miller, G. & Johnson-Laird, P.N. Language and perception. Cambridge, Ma.: Harvard University Press, 1976.

Rosch, E. & Mervis, C.B. Family resemblance studies in the internal structuring of categories. Cognitive Psychology, 1975, 7, 573-605.

Wright, G.H. Von Norm and action:a logical enquiry. London: Routledge and Kegan Paul,1963

A PARALLEL MODEL OF (SEQUENTIAL) PROBLEM SOLVING

Mary S. Riley and Paul Smolensky

Institute for Cognitive Science
University of California, San Diego
La Jolla, California 92093

Nature of Rules and Their Interaction

This paper is concerned with the nature of the rules involved in solving problems and the interaction between those rules. We describe a parallel model designed to solve a class of relatively simple problems from elementary physics and discuss its implications for models of problem solving in general. We show how one of the most salient features of problem solving, sequentiality, can *emerge naturally* within a parallel model that has no explicit knowledge of how to sequence analysis.

Consider the problem shown in Figure 1. The task is to determine the qualitative effects of increasing the resistance of R_2 on other circuit values, assuming the applied voltage and resistance of R_1 remain unchanged.

A common approach to modelling the process of solving problems like these is to assume that knowledge is organized as a production system, similar to that shown in Table 1 (see Riley, 1984, for a review). Here the model's rules for making inferences are in the form of condition-action pairs, or *productions*. The condition specifies the particular elements and relations that must be present in the data base in order for the condition to be true. When the production system is solving a problem, the conditions of the various productions are tested in order until one of them is true; the action of that production is then performed. The action generally makes some change in the data base which in turn means the condition of a different production will be true, causing another action to be performed.

Since production systems are universal computers, they can be programmed to display any behavior (Newell, 1981). However certain kinds of behavior can be achieved with other styles of computation in more economical, elegant, extendible and natural ways. Features that are intrinsic to, or naturally incorported within, a pure production system approach are:

1) *Sequentiality:* each action taken utilizes the knowledge contained in precisely one rule.
2) *Directionality:* the knowledge encoded in each rule has a distinct directionality from input (condition) to output (action).
3) *Exact matching:* each rule acts only when a perfect match to its condition occurs.
4) *Determinism:* performance will be identical on all solutions of a given problem.

Within the production system approach it is difficult to naturally avoid certain difficulties:

1) *Lack of robustness under degradation of rules* (either removal of correct rules or addition of incorrect ones).

This research was supported by a grant from the System Development Foundation and by contract N00014-79-C-0323, NR667-437 with the Personnel and Training Research Programs of the Office of Naval Research.

2) *Lack of robustness under ill-formed problems* that contain inconsistent or insufficient given information.
3) *Lack of variability* in routes to correct answers or in correctness of answers; a problem for modelling human behavior.
4) *Need for explicit conflict resolution rules* that determine which rule will apply when several have true conditions.

The parallel distributed processing approach represented by our model naturally avoids these difficulties, but has its own problems, as we shall see.

The Model

Our model has been constructed within the framework of harmony theory (Smolensky, 1983, 1984). Rules are represented as a collection of nodes in a network, as shown in Figure 2. A typical rule is $<I\ down, V_1\ down, R_1\ same>$; this rule states that the combination of changes "voltage across R_1 goes down, current goes down, R_1 stays the same" is a consistent set (Ohm's Law). In fact, the rules consist precisely of all allowed combinations of qualitative changes in circuit variables that are consistent with each circuit law. There are 65 such instances. [1]

Unlike condition-action rules, there is no directionality associated with the variables in the harmonium rules.

In a particular problem, only some of the instances represented by harmonium rules are relevant. To represent this, each rule node has an *activation value* that can be either 1 (active) or 0 (inactive).

In addition to rule nodes, the harmonium model contains nodes for representing the problem in terms of qualitative changes in circuit variables. Some nodes have values given by the problem ($R_2\ up, R_1\ same, V_{total}\ same$). The model's answer is represented by values assigned to the remaining nodes.

As shown in Figure 2, there is a connection between an individual circuit variable node and each rule involving it; this connection is labelled by the appropriate value for that variable according to that rule.

The goal of processing is to find a set of rule nodes to activate and a set of values for circuit variables that are consistent with those rules. Search toward this goal is guided by a measure of the consistency between a set of activated rules and a set of circuit variables: this measure is called the *harmony function*. The state of highest harmony should be the correct answer to the problem.

Processing is probabilistic and constructed so that at any moment, *the higher the harmony of a state, the more probable it is*. The spread in this probability distribution is determined by a system parameter called the *computational temperature* T. Initially, all rules are inactive, the circuit variables given by the problem are assigned their values, and the remaining circuit variables are assigned random values. The temperature is set to a high value, and the stochastic search begins. Rules are activated and deactivated, circuit variable values are changed (except the given ones), and states are visited in accordance with their harmony. As the search continues, this temperature is lowered, and the system displays less and less randomness, focussing in on the states of highest harmony. After a while, the temperature becomes very low, and the search is effectively stopped: an answer has been selected.

---

1. Thirteen each for: Kirchoff's Law, the equation $R_{total} = R_1 + R_2$, and three versions of Ohm's Law (one each for R_1, R_2, and R_{total}).

Sequentiality of deduction seems to be completely lacking from the harmonium model, although it is a very salient feature of human problem solving. Just the same, in creating this model we expected it to display an emergent seriality. If a single circuit variable is monitored during the search, it will fluctuate randomly at first, and eventually "lock in" to a value that is very resistant to change. The temperature at which this occurs is the "freezing temperature" for that variable. We expected that different variables would have different freezing temperatures, depending on the problem situation; the one with highest freezing temperature would settle first, which would in turn determine the value selected for the variable with the next lower freezing temperature, and so on.

In addition to T, harmonium models have a second global parameter, κ; it is the sole parameter in the definition of the harmony function. When κ is near one, only rules that match the current guesses for circuit values *exactly* can become active without lowering the harmony of the state; for low values of κ, approximate matches are sufficient. Initially, κ is small, approximate matching is encouraged, and many rules become activated; as the computation proceeds, κ approaches one and the set of active rules shrinks toward the five that exactly match the answer.

As the node for each circuit variable freezes into a value, it does so under the influence of all the active rule nodes connected to it. Unlike a production system, matching for rules need not be exact, and several rules can act at the same time.

The harmony function we used, as well as the schedule for lowering T and raising κ, are shown in Figure 3. A trial consisted of 400 iterations of 100 node updates each; since there are 79 nodes in the model, this corresponds to slightly over 500 updates of each node.

The stochasticity of the model produces variability in the behavior. In a run of 30 trials, the correct answer was produced 28 times. When the 30 values the system assigned to the circuit variables for each of the 400 iterations are averaged, Figure 4 results. In this graph, *up* is represented by 1 and *down* by -1. Initially, the values for all variables fluctuate around zero; eventually, each goes towards the correct value. The time at which the four decisions are made are indicated in the last portion of this figure, in which the region between .5 and $-.5$ has been removed. The sequence of assignments is R_{total}, I_{total}, V_1, V_2; *the sequence of "inferences" that emerges naturally from the parallel processing is exactly the same as the sequence produced in a production system model.*

The harmonium model displays both types of robustness that is difficult to achieve naturally with production systems. Since individual inferences are made under the simultaneous influence of several rules, they are less vulnerable to degradation of a single rule. When inconsistent information is given in a problem, the harmonium model finds the most consistent (highest harmony) answer possible. When insufficient information is given, the system finds one of the correct answers, and finds different answers on different trials. Such a robust tendency to form coherent interpretations of inputs is important both for modelling human cognition and for building intelligent machines.

Extensions

While the parallel distributed processing approach has certain advantages over the production system approach, it also has grave disadvantages. The most serious is the difficulty of performing symbol manipulation. Without variable binding mechanisms, types and tokens, it is difficult to imagine how to develop a general model capable of analyzing a variety of circuits; our model is specialized to a single circuit, and even so we must replicate the rules encoding valid instances of Ohm's Law three times (once for each relevant piece of the circuit).

It may be psychologically plausible to postulate a small collection of networks like our harmonium model (or perhaps one integrated, larger network) incorporating knowledge about similarly simple circuits (e.g. a circuit with two resistors wired in parallel). These could conceivably serve as prototypes that would be invoked to deal with pieces of or schematic versions of larger circuits. However some powerful mechanism would still have to coordinate the parallel analyses of circuit fragments.

It is tempting to use a production system for this coordination, combining the strengths of the two approaches. Such a hybrid model might well be able to analyse complex circuits, but would display the production system weaknesses (lack of robustness, and so forth) in those aspects of the analysis that were relegated to the production system.

One interpretation of such a hybrid model is that the production system component is actually just a complex parallel processing network *viewed at a higher level of description*; the hybrid is of descriptive levels, there are not two independent processes. It is a major goal of ours to see if parallel models are capable of exhibiting emergent production-like behavior; the emergent seriality of the present harmonium model is an example of just such behavior.

Discussion

The harmonium model has *implicit* knowledge of circuit laws that enable it to model naturally the nonsymbolic, intuitive component of problem solving that is difficult to model naturally with production systems and is particularly salient in expertise. At the same time it lacks the *explicit* knowledge of symbolic laws that most experts possess. Thus to model expert problem solving in general, it seems necessary to imbed the harmonium model within a hybrid parallel/production system model. We are however investigating whether the symbolic component of expert's processing can be preempted with conditions of very short response times, making such experimental conditions appropriate for testing the pure harmonium model. We are also considering unschooled electronics experts to see to what extent they are free of conscious rule application in their solution of simple circuit problems.

Much work remains to be done in analyzing the variation in the model's performance, and assessing the dependence of performance on the schedules for T and κ and the representation of the circuit. Indeed it is the development of more powerful representations within the parallel distributed processing paradigm that is the primary goal of harmony theory; by trying to enrich the knowledge of our harmonium model to incorporate more "symbol-like" explicit knowledge of circuit laws, we hope to gain more insight into how symbol manipulation might emerge from parallel distributed processing.

References

A. Newell (1981). Physical symbol systems. *Perspectives on Cognitive Science*, D. A. Norman (ed.), 37-85. Norwood, NJ: Ablex.

M. S. Riley (1984, expected date). Structural understanding in problem solving. Doctoral dissertation. Pittsburgh, PA: LRDC, University of Pittsburgh.

P. Smolensky (1983). Schema selection and stochastic inference in modular environments. *Proceedings of the National Conference on Artificial Intelligence*. Washington, DC.

P. Smolensky (1984). The mathematical role of self-consistency in parallel computation. Submitted to the Sixth Annual Conference of the Cognitive Science Society. Boulder, CO.

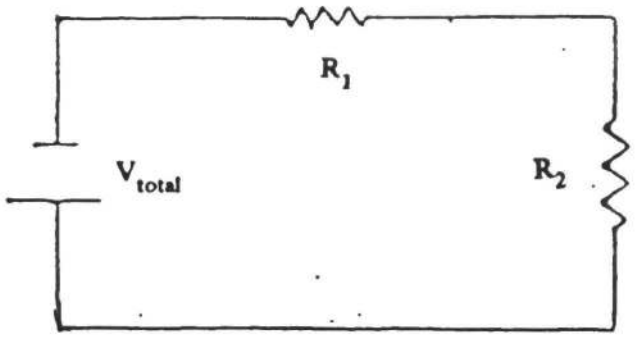


Figure 1. A series circuit with two resistors, R_1 and R_2. What are the effects of an increase in the resistance of R_2, assuming that E_{total} and the resistance of R_1 have remain the same?

Table 1

A Simple Production System for Solving the Problem in Figure 1.

Productions

	Condition	Action
P1.	$<V_x$ *same*, R_x *up*$>$	$<I$ *down*$>$
P2.	$<R_x$ *up*, R_y *same*$>$	$<R_{total}$ *up*$>$
P3.	$<V_x$ *down*, V_{total} *same*$>$	$<V_y$ *up*$>$
P4.	$<R_x$ *same*, I *down*$>$	$<V_x$ *down*$>$

Problem Solution

Problem Representation

Cycle	
1.	R_2 *up*, R_1 *same*, V_{total} *same*
2.	R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*
3.	R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*, I_{total} *down*
4.	R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*, I_{total} *down*, V_1 *down*
5.	R_2 *up*, R_1 *same*, V_{total} *same*, R_{total} *up*, I_{total} *down*, V_1 *down*, V_2 *up*

Matched Production

	Condition	Action
P2.	$<R_2$ *up*, R_1 *same*$>$	$<R_{total}$ *up*$>$
P1.	$<V_{total}$ *same*, R_{total} *up*$>$	$<I$ *down*$>$
P4.	$<R_1$ *same*, I *down*$>$	$<V_1$ *down*$>$
P3.	$<V_1$ *down*, V_{total} *same*$>$	$<V_2$ *up*$>$

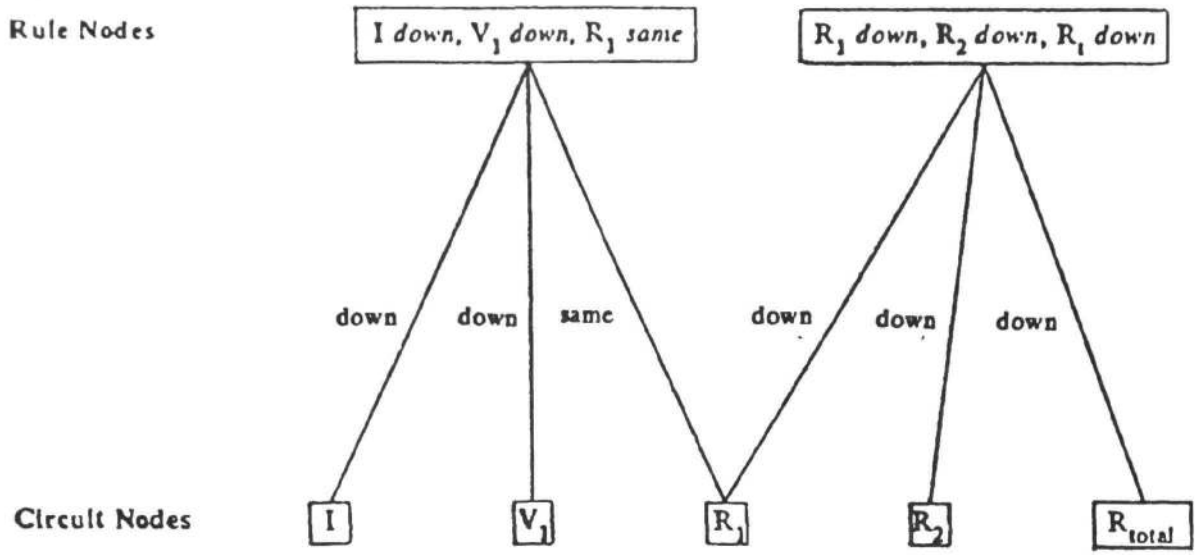
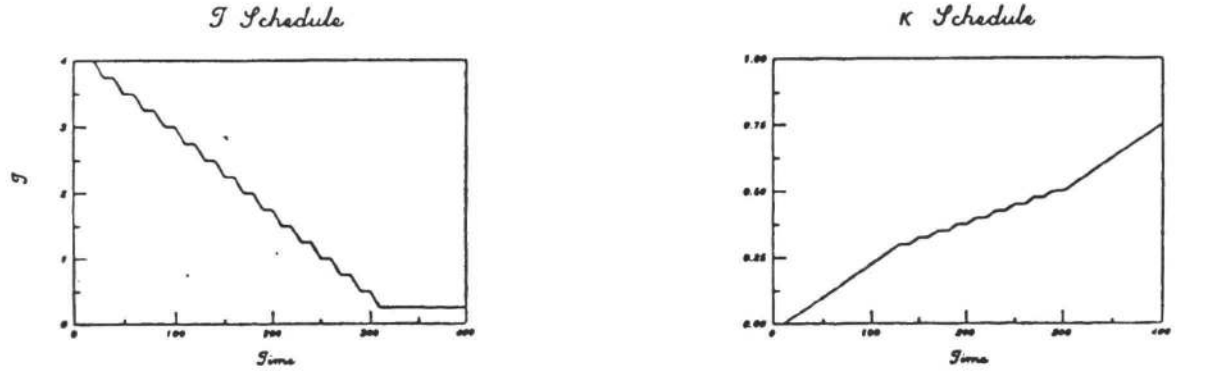


Figure 2, A portion of the harmonium model's network.

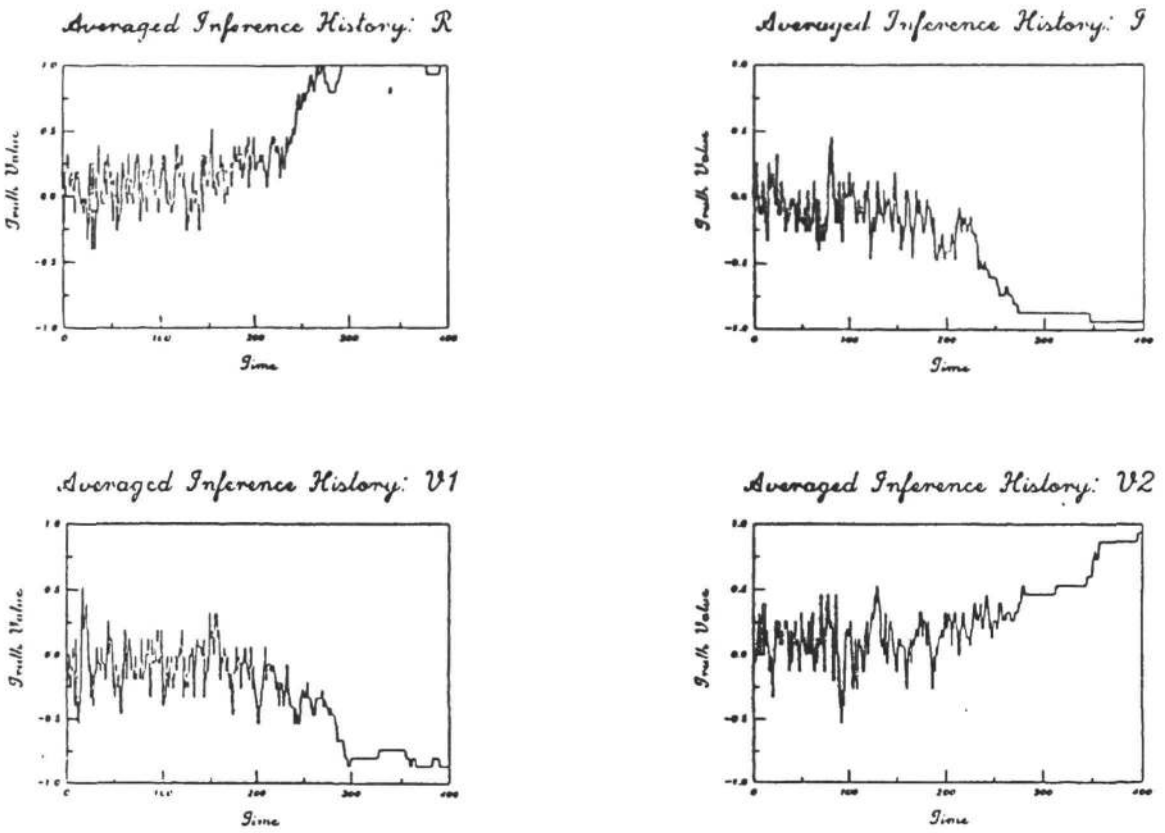


I:	up	down	same	
I.c:	+1	+1	−1	["I changed"]
I.u:	+1	−1	±1	["I went up"]

Harmony function:

$$H = \sum_{\alpha} A_{\alpha} \sum_{i} \left(R_{\alpha i} B_i - \kappa |R_{\alpha i}| \right)$$

Figure 3. Schedules for T and κ, representation of *up*, *down*, *same*, and harmony function used in the harmonium model.



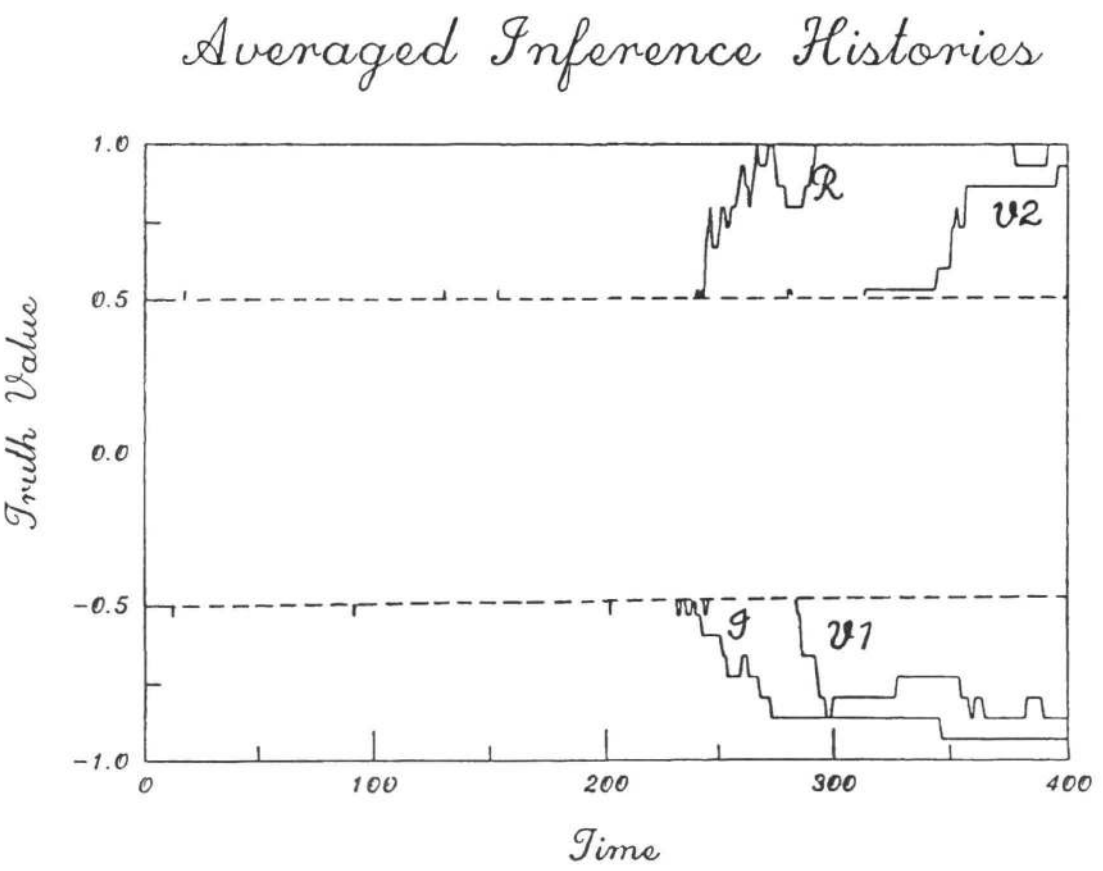


Figure 4. Emergent sequentiality: the decisions about the direction of change of the circuit variables "freeze in" in the order $R = R_{total}$, $I = I_{total}$, V_1, V_2 (R and I are quite close).

Steps Along the On-Line Assistance Spectrum

EDWINA L. RISSLAND

Department of Computer and Information Science
University of Massachusetts
Amherst, MA 01003

Abstract

In this paper, we discuss the spectrum of on-line assistance ranging from passive, canned to active, user-customized. We discuss various aspects of on-line assistance: interactive introductory tutorials, on-line help, and on-line manuals. We then describe two steps to make on-line assistance more intelligent: (1) inclusion and customization of examples in the information provided the user; and (2) integration of various aspects of on-line assistance like tutorials and help.

1. Introduction

As any neophyte would probably attest, it is hard to get started on a new computer system. One thing contributing to this difficulty is the lack of intelligent on-line assistance in the interface. Often there is a rudimentary on-line help facility, but it is clumsy to use. It is also "dumb" in that it lacks many key ingredients of expert knowledge, like examples and heuristics; it consists of canned responses that are always the same regardless of the user's background, task, goals, context, etc.

Then too, the interaction is neither gracious, graceful nor friendly (see, for example, the help facility on VAX/VMS):

1. access is tedious (e.g., menus too long and unorganized);
2. presentation of material is often insensitive (e.g., screenfuls of text whizzing by)
3. it requires the user to speak its language to get anything useful out of the help invocation (e.g., the user might not get any useful information because he asks about "quit" when he should've asked about "logout").

Such difficulties subvert the user's expression of his intentions -- he knows what he wants to do but not how to say it -- and ignore a key source of knowledge in intelligent user interfaces [Norman 1984].

---

[1] This work supported in part by Grant IST-8212238 of the National Science Foundation.

At the other end of this spectrum are intelligent assistance providers -- interactive tutors and coaches that use A.I. techniques like user modelling (e.g., [Woolf 1984]). More intelligent forms of on-line assistance need both more knowledge and power: that is, many types of knowledge (e.g., of the user's task, domain, personal experience, etc. [Rissland 1984]), modes of interaction (e.g., natural language [Walker 1976]), and processes (e.g., inference engines that isolate the user's misconceptions [Lewis and Soloway 1984]).

Thus we see a spectrum ordered by responsiveness and intelligence. At the low or negative end of the spectrum are the dumb systems with canned responses, no interactive capability, and only minimal domain knowledge (e.g., VAX/VMS HELP). A little better are interactive "dumb" facilities, like tutorials, without even explicit models of the user or the knowledge to be explained.[2] Enriching the knowledge base and enhancing the interaction can move assistance further in the "positive" direction. Adding user-modelling and the ability to provide responses custom-tailored in content and form place the assistance facility well towards the positive end of the spectrum.

So far that end of the spectrum has not been much explored, although interesting starts have been made. Wilensky, in his UC system, allows the user to ask for assistance in natural language [Wilensky 1982a, 1982b]; his work has concentrated on request understanding. Finin, in his WIZARD system, focuses on the problem of recognizing when the user needs help; for example, when he is using inefficient means to do something, like using repeated DELETEs instead of PURGE, the system then volunteers advice [Finin 1983; Shrager and Finin 1982].

Unhappily, however, most assistance facilities available today still lie clearly towards the dumb end of the spectrum.

In the rest of this paper, we will describe some steps to move on-line assistance further along towards the intelligent end of the assistance spectrum through the use of richer domain knowledge and the integration of various aspects of assistance. We shall assume that the on-line assistance facility has already been invoked (by the user or the system) and that the facility has already 'parsed' the user's request (i.e., knows what the user requires assistance on). Our emphasis is on the generation of the response, and in particular, on the embedding of examples. Clearly, this work should eventually be tied in with work on invocation and parsing, like that of Wilensky or Finin, and an obvious extension would be the use of a program like McDonald's MUMBLE [McDonald 1982] to dynamically generate text as well as examples.

---

[2] For example, the much touted tutorial for the LOTUS 1-2-3 spreadsheet program is a step-by-step on-line tutorial; it is better than most, but its inflexibilities can be daunting to a user.

2. Aspects of Assistance: Help, Tutorials and Manuals

A recent review article [Houghton 1984], discussed several types of on-line assistance, including some pertinent here, namely, command assistance, introductory tutorials and manuals. Other types are error and prompting assistance. We will not discuss these here but many of our points are relevant to them as well. Of course, the ideal assistant is very often a human on-line consultant.

By a *tutorial*, we mean an interactive, structured program that introduces a user to a system (e.g., the EMACS tutorial [Stallman 1983]). A tutorial presents information but does not necessarily allow free rein in the interaction; some don't even allow one to jump around. What one can do next, like read further or supply parameters to a demonstration, is largely pre-determined.

By on-line *help*, we mean command assistance. The user asks for information about a particular command, like "HELP PRINT", and is then presented with information on PRINT, including relevant parameter options, but almost never including examples of standard, potentially dangerous, or clever uses.

By an on-line *manual*, we mean a version of the hard copy manual (whatever its content or organization) which is available to the user on-line together, hopefully with some sort of access interface, the case of a standard text-editor (in read-only mode) being the bare minimum.

Help and tutorials are clearly more interactive than manuals, although one can easily imagine interactive manuals as well. Tutorials can provide a "guided tour" of a system and an opportunity for the user to try things out in a sheltered, or hypothetical, environment.

3. Embedding Examples in On-line Assistance

One important component of knowledge that is missing in most on-line (and off-line) assistance is examples. Examples, by which we mean specific cases and instantiations, are one of the most important ingredients of expert knowledge. They offer concrete illustrations of what is being explained and memorable hooks into more general information. They are especially important for the beginner.

Examples can provide easily understood and remembered usages. For instance,

PRINT VITA.MEM

is clearly more perspicuous than

"**PRINT** [[*d:*][*filename*[.*ext*]][/T][/C][/P]...]" (from [IBM 1983])

A novice can use simple cases: to figure out how to instantiate the general syntactic description, to use as "recipes" for standard tasks, as a basis for generalizing, and as a basis for a "retrieval+modification" approach to generate another instance. For the more expert, examples can serve as a reminder of syntax and things previously done, much like an icon; this is especially useful with commands used only infrequently.

[Rissland 1978] presented a taxonomy of examples: "start-up" (easy, perspicuous cases); "reference" (standard, textbook cases); "model" (paradigmatic, template-like cases); "counter-examples" (limiting, illegal cases); "anomalous" (ill-understood, strange cases).

Here we use such a taxonomy to select and order the presentation of examples. For instance, we provide the neophyte user with "start-up" examples and the more experienced user with "references". Where a sequence of examples is called for, references are presented before models which are presented before counter-examples and anomalies. This taxonomy can enable the user to ask specifically for examples in a certain class (e.g., "easy" or "dangerous").

Another aspect of examples which we have previously studied is their generation [Rissland 1981]. Two modes of generation are "retrieval+modification" and instantiation. We use these techniques in on-line assistance by linking the assistance program with an example generator, which has an "examples-space" of already existing examples, and procedures for modification and instantiation. The examples, which have been harvested and organized by an expert, are represented as frames; they contain slots for information such as a graphics demonstration, difficulty rating, and pointers to more and less complicated examples. Modification operators include procedures to personalize examples (e.g., if an example needs a file-name, use one of the user's). Instantiation procedures include ways to generate a range of cases, including those that satisfy and violate legal parameter values.

This ability to dynamically generate examples allows the assistance facility to provide examples tailored to the user, his tasks, goals, context, domain, etc.; it depends, of course, on having some sort of user-modelling capabilities. The idea is to work examples into the assistance given the user, and better still to make the examples meaningful in the sense of relating to the user.

4. Integrating Aspects of On-line Assistance

One difficulty in learning or checking out a feature of a system is that the various aspects of on-line assistance do not share a common language or set of examples, and thus it is hard to integrate one's knowledge or to apply information from one source to another. This violates the pedagogical strategy of using information seen before by the learner, like examples which have become "old friends".

Our approach to this consistency/integration problem is to have all the aspects of on-line assistance share common source material -- examples and text -- which is represented in a way usable by each individual aspect. Each aspect then puts its presentations together by retrieving the text and examples it needs from the common source.

In our work, we use a script-like control structure of a text and examples template, a "TEXPLATE". A TEXPLATE typically contains pointers to chunks of text, calls for examples, and control information. It can also contain "literal" material (like text used nowhere else) which is presented "as is". Calls to examples are either requests for explicitly named examples in the Examples Knowledge Base (EKB) or constraints by which the the example generator can generate a new example. For instance, an example call could be for a named counter-example or for an example generated to fulfill prescribed constraints (like one using the name of the user's most recently created file). Control information includes options to present to the user and the appropriate assistance-module response actions: for instance, MORE to cause the tutorial to go on, EXAMPLE for an example, QUIT, etc. Control information also contains directions for which sequence of examples the system should present if the user repeatedly selects the EXAMPLE option.

5. Two On-going Assistance Studies: IA-LADYBUG & VMS

In our on-going work, we are working within two systems. The first is IA-LADYBUG, a system designed specifically for novice programming students. It introduces them to notions useful in the Pascal programming language (like subprocedures) by having them work with a graphics icon, the LADYBUG, which can be commanded by LOGO-like commands like CRAWL, RIGHT-TURN, etc. [Levine and Woolf 1984]. The second is a subset of VAX/VMS command language [DEC 1978] dealing with directory commands like PURGE, DELETE, and SET PROTECTION.

For IA-Ladybug (over whose environment we have total control), the student manual, on-line introductory tutorial and interactive on-line HELP share material. The TEXPLATES are indexed by command and topic and reference the manual's

text file for textual material and a separate EKB for examples. Often, the tutorial and interactive HELP present dynamic examples that are merely summarized in the manual, for instance drawing a ball bouncing or a seven color sunburst. The tutorial and HELP present examples that are too complicated or whose effect (like color) would be lost in the manual.

The simpler "start-up" and "reference" examples presented in the manual are the first examples presented in the tutorial and HELP. HELP, especially, goes on to present more complex or difficult examples, like counter-examples to show the limits of commands (e.g., RIGHT 362 exceeds the parameter range for degrees of turning). At this time, HELP also does some very simple tuning of its examples to the user, for instance by using information about the user's directory and the user's own answer to whether or not he is an expert.

6. Summary

In this paper we have discussed two steps to making on-line assistance more intelligent: (1) inclusion and customization of examples in the information provided the user; and (2) integration of various aspects of on-line assistance like tutorials and command help. We have used knowledge about the structure, types, and generation of examples to implement (1) and a control structure of text and examples, called a "TEXPLATE", to achieve (2).

Currently we are experimenting with our prototype on-line assistance modules. One thing we have learned is that subjects do not read very well and that examples are a quick way to impart a lot of information. We have also found that using texplates to separate the control from the substance makes it easy to re-write the assistance scripts. Another observation is that for examples that are graphics demos, it would be nice for the user to be able to do "instant replays in slow motion" and to be able to take a deeper look at the code behind the example.

7. Acknowledgements

The author acknowledges the work of the members of the "help team", specifically E. Valcarce who implemented the on-line command assistance, L. Gordon who developed the IA-Ladybug tutorial, and R. Filoramo who wrote the IA-Ladybug Manual. Also thanks to B. Woolf and L. Levine for their critical discussions and to O. G. Selfridge for sharing his ideas.

8. References

DEC, *VAX/VMS Command Language User's Guide*. Digital Equipment Corporation. Order No. AA-D023B-TE, 1978.

Finin, T. W., "Providing Help and Advice in Task Oriented Systems". In *Proceedings IJCAI-83*. Karlsruhe, W. Germnay, 1983.

Houghton, R. C., "Online Help Systems: A Conspectus". *CACM*, Vol. 27, No. 2, February 1984.

IBM, *Disk Operating System by Microsoft, Inc.*. IBM Personal Computer Language Series, IBM Corp, 1983.

Johnson, L., and Soloway, E. M., "PROUST: Knowledge-Based Program Debugging". In *Proceedings Eighth Int'l Software Engineering Conference*, Orlando, FLA, March 1984.

Levine, L., and Woolf, B., "Do I Press Return?" In *Proceedings ACM-SIGCSE Symposium on Computer Science and Education*, Philadelphia, February 1984.

McDonald, D. D., "Natural Language Generation as a Computational Problem: An Introduction". In Brady (Ed.) *Computational Theories of Discourse*, MIT Press, 1982.

Norman, D. L., "Stages and Levels in Human-Computer Interaction". To appear in *International Journal of Man-Machine Studies*, summer 1984.

Rissland, E. L., *Constrained Example Generation*. COINS TR 81-24, Department of Computer and Information Science, University of Massachusetts, Amherst, 1981.

Rissland, E. L., "Ingredients of Intelligent User Interfaces". To appear in *International Journal of Man-Machine Studies*, summer 1984.

Rissland, E. L., "Understanding Understanding Mathematics" *Cognitive Science*, Vol. 2, No. 4, 1978.

Shrager, J., and Finin, T. W., "An Expert System that Volunteers Advice". In *Proceedings AAAI-82*, Pittsburgh, PA, August 1982.

Stallman, R. M., "EMACS: The Extensible, Customizable, Self-Documenting Display Editor". In Barstow, Shrobe and Sandewall (Eds.) *Interactive Programming Environments*, McGraw-Hill 1984. Also available as MIT AI Lab Memo 519a, 1981.

Walker, D. E. (Ed.), *Speech Understanding Research: Final Report*. Stanford Research Institute, Menlo Park, CA, 1976.

Wilensky, R., "Talking to UNIX in English: An Overview of UC". In *Proceedings*

AAAI-82, Pittsburgh, PA, August 1982a.

Wilensky, R., *Talking to UNIX in English: An Overview of an On-line Consultant*. Report No. UCB/CSD82/104, Computer Science Division, University of California, Berkeley, September 1982b.

Woolf, B. P., *Context Sensitive Text Planning in Tutorial Discourse Generation*. Ph. D. Dissertation, Dept. of Computer and Information Science, University of Massachusetts, Amherst, May 1984.

Distinct Characteristics of Verbatim, Propositional and Situational Representations in Text Comprehension

Franz Schmalhofer

Universitaet Heidelberg, West Germany

While cognitive scientists have investigated in some detail how subjects remember texts (Kintsch, 1974; Kintsch & Van Dijk, 1978; Schank & Abelson, 1977), in real life texts are often studied with a completely different intention. For example, a student studying a computer science textbook or a car mechanic studying a repair manual is more interested in acquiring knowledge about the respective subject domain as opposed to merely remembering the wording or meaning of the text. In order to become a successful computer programmer, a person must form a general representation of the respective computer language and how to use it, including many possible situations which arise when programming a computer.

It may therefore be expected that in addition to verbatim and propositional text representations, a reader also forms a cognitive representation of real or hypothetical situations addressed by a text. Van Dijk and Kintsch (1983) have presented several arguments that the representation of a text and the representation of real or possible situations mentioned by a text do not always coincide and may therefore have their own distinct cognitive existence, so that three different cognitive structures should be distinguished: Whereas verbatim memory and the propositional textbase reflect the wording and the meaning-

structure of a particular text, respectively, a situational or mental (Johnson-Laird, 1980) model is assumed to represent situations of the real or some possible world about which a given text presents some new information.

The cognitive architecture of verbatim and propositional text representations on the one hand, and representations of the situations referred to by a text on the other, were examinined in four experiments. By instructing subjects to either read a text for text summarization or for knowledge acquisition, the first experiment investigated differences in the encoding processes of the textbase and the situational model, while the second experiment examined differences in the resulting cognitive structures. In order to test the different information retrieval speeds from the three cognitive structures, a speed-accuracy trade-off analysis was employed in a third experiment. A fourth experiment investigated how the construction of propositional and situational representations depends upon a reader's prior domain knowldege.

Experiment 1

Because the textbase and the situational model may be constructed by possibly interacting, but nevertheless separate, mental processes from different cognitive elements (Anderson, 1983), it is expected that subjects who read a text in order to write a summary (text summarization or TS readers) thereafter, would show different reading time patterns than readers who study the same text in order to acquire knowledge about the respective subject domain (knowledge acquisition or KA readers). In order to investigate the construction of a textbase and a situational model in a realistic but controlled setting, 64 subjects who did not

know anything about LISP were given part of a LISP programmer's manual to study. The experimental text had a clearly identifiable hierarchical structure. Whereas the paragraphs at the highest level (level 1) in the text hierarchy expressed the text's macrostructure, substantive LISP information, which is needed for the construction of a situational model, was presented at the lower levels of the text hierarchy. Since the most important information for constructing a textbase and a situational model were contained in different paragraphs, differences in the cognitive processing of TS and KA readers could be assessed by comparing the reading times of different text segments.

The average reading times per word for the different text levels are shown in Figure 1, for each of the two subject groups. Whereas TS readers show a clear levels effect with the longest word reading times for the highest level in the text hierarchy, KA subjects showed the longest word reading times for the second text level, which presented substantial information about the programming language LISP.

These results suggest that, by emphasizing macroprocessing TS readers were more thoroughly engaged in constructing a textbase, whereas KA readers focussed on developing a situational model by processing the more substantive information about LISP. The cognitive products of these differential encoding processes were examined in a second experiment.

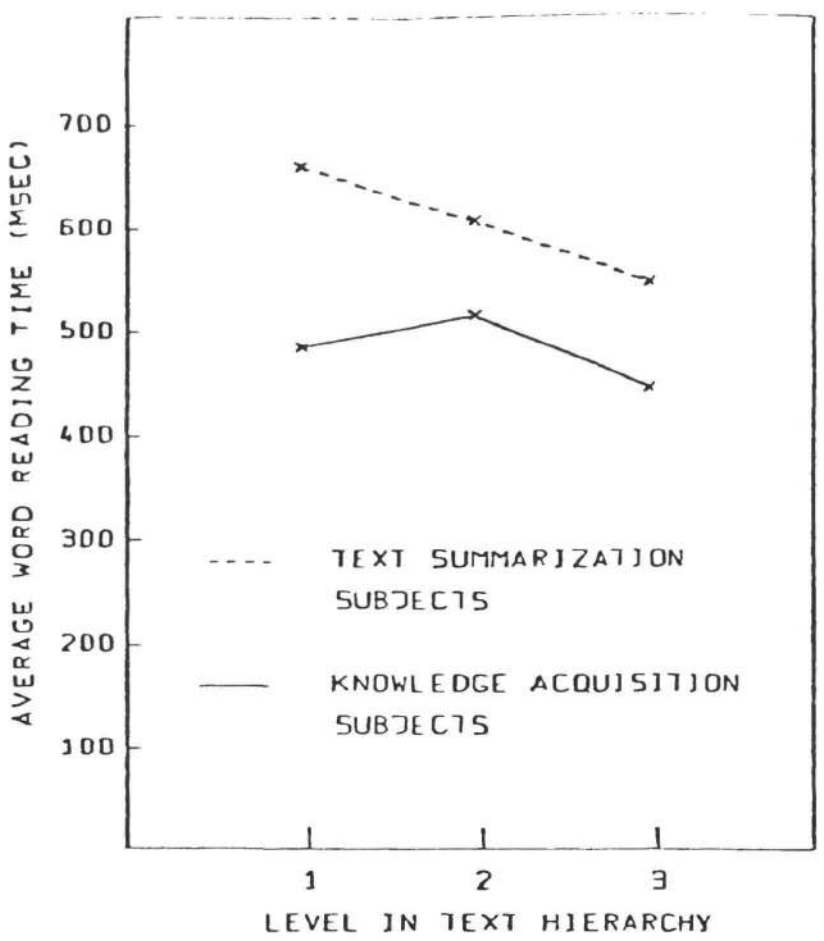


Figure 1. Average reading times per word (msec) as a function of the level in the text hierarchy for each of the two study instructions (studying in order to write a text summary or studying in order to acquire knowledge about the subject domain).

Experiment 2

In order to examine the relative strength of verbatim, propositional and situational representations, a retrieval model was specified for the three cognitive structures. It was assumed that during the recognition processing of a sentence, the retrieval results of the three structures are continuously combined (e.g. added) to yield the currently accumulated recognition strength at any point in time. In

addition, it was assumed that the accumulated recognition strength determines a subject's recognition decision. By presenting subjects with test sentences which differ only by the contribution of one of the three cognitive structures, the strength of the respective structure may be examined. Four different types of test sentences can be constructed: A sentence may be presented in the original form it occurred in the text (O-sentences); it may be paraphrased (P-sentences); its meaning may be changed, while preserving its situational correctness (M-sentences); and its situational correctness could be changed in addition (C-sentences). As shown in Table 1, the O-P, P-M, and M-C sentence pairs differ only by the contribution of the verbatim, the propositional and the situational representations, respectively.

TABLE 1

Contribution of verbatim memory, the textbase, and the situational model to each of the four sentence forms

	test sentence			
	correctness changed	meaning changed	paraphrased	original
verbatim memory	-	-	-	+
textbase	-	-	+	+
situational model	-	+	+	+

Note. The "+" and "-" indicate whether a cognitive structure supplies evidence for a yes (old) or no (new) recognition decision, respectively.

The strength of verbatim, propositional and situational representations may thus be assessed in a signal detection analysis by respective d' values. The mean d' scores of verbatim, propositional and situational representations obtained for TS and KA readers, whose overall text study time was controlled, are shown in Table 2.

TABLE 2

d' accuracy-scores of each processing goal

for the three cognitive structures

processing goal	representation		
	verbatim	propositional	situational
test summarization (TS)	-0.10	0.84	1.15
knowledge acquisition (KA)	0.38	0.25	1.42

These results show that TS and KA readers emphasized different components of text processing. Whereas TS readers developed a better propositional text representation, KA readers emphasized the construction of a situational model. By demonstrating how the development of cognitive structures depends upon a reader's processing goals, these results provide additional evidence for the distinction of a propositional text representation and a situational model.

Experiment 3

In order to eliminate the influence of short term memory and to further examine the speed with which information is retrieved from the three cognitive structures, an experiment with an interfering task between the study and the test phase was performed. Figure 2 shows the

average d' retrieval scores of verbatim, propositional, and situational information which were obtained for the different processing times.

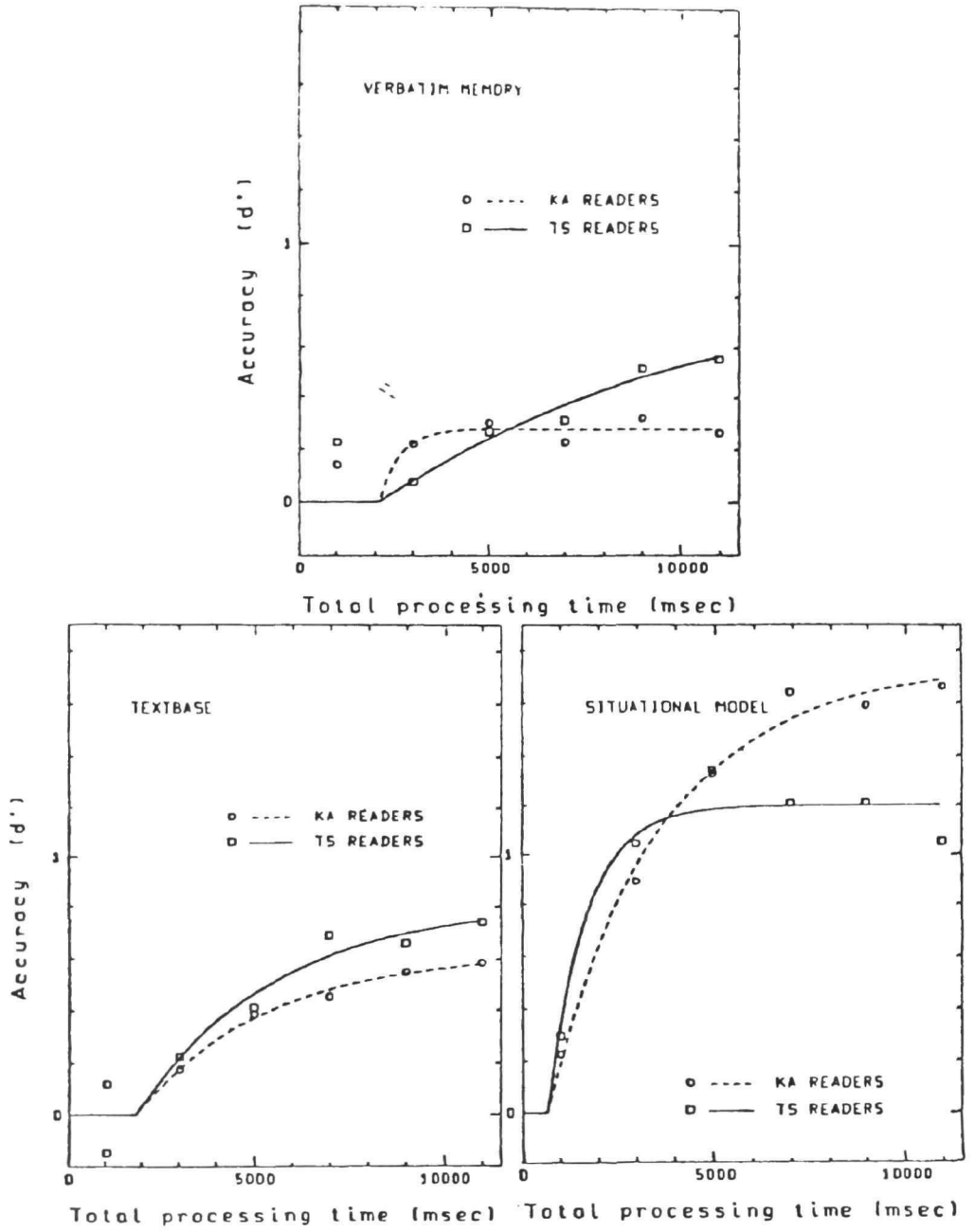


Figure 2. Accuracy scores (d') at different processing times for each of the three retrieval components (verbatim memory, textbase, and situational model) and the two study instructions (studying in order to write a text summary, TS, or studying in order to acquire knowledge, KA. The smooth curves represent best fitting speed-accuracy trade-off functions.

Instead of verbatim information, subjects based their recognition decisions mostly upon propositional and situational information. Also, KA readers retrieved more situational information than TS readers, although situational information was retrieved faster than propositional information for both subject groups. The results thus indicate that accessing a situational model is faster and proceeds at a higher speed than accessing a textbase. Even for recognition decisions, situational information is more important than verbatim or propositional information. It thus appears that in addition to verbatim and propositional text representations, the construction of a situational model is an important component of representing knowledge about the subject domain of a text. Subjects seem to utilize situational information for judging a sentence by its plausibility (Reder, 1982) which proceeds faster than searching memory for a propositional match with the textbase.

Experiment 4

Subjects with and without prior knowledge about computer programming studied a programmer's manual (LISP). For all subject groups, sentence reading times increased with the number of propositions in a sentence, indicating the construction of a textbase. All subjects successfully remembered the text by its meaning rather than by its wording. While subjects without prior domain-specific knowledge only remembered the text itself, subjects with prior domain-specific knowledge in addition acquired general knowledge about LISP. The construction of a situational model is thus more dependent upon a reader's prior knowledge than the construction of a situational model. It may thus be concluded that text memory is a by-product of general

comprehension heuristics, such as micro- and macroprocesses. However, the updating of world knowledge critically depends upon a reader's prior knowledge.

In the four experiments, distinct characteristics of verbatim, propositional, and situational representations were thus determined in the domain of technical texts by examining encoding and retrieval processes, cognitive structures, different processing goals, and expert-novice differences.

References

Anderson, J.R. The architecture of cognition. Cambridge, MA: Harvard University Press, 1983.

Johnson-Laird, P.N. Mental models in cognitive science. Cognitive Science, 1980, 4, 71-115.

Kintsch, W. The representation of meaning in memory. New York: Erlbaum, 1974.

Kintsch, W., & van Dijk, T.A. Toward a model of text comprehension and production. Psychological Review, 1978, 85, 363-394.

Reder, L.M. Plausibility Judgments versus fact retrieval: Alternative strategies for sentence verification. Psychological Review, 1982, 89, 250-280.

Schank, R.C., & Abelson, P.R. Scripts, plans, goals, and understanding. Hillsdale, New Jersey: Erlbaum, 1977.

van Dijk, T.A., & Kintsch, W. Strategies of Discourse Comprehension. New York: Academic Press, 1983.

Being Reminded of Thematically Similar Episodes

Colleen M. Seifert
Robert Abelson
Yale University

Gail McKoon
Northwestern University

Some of our knowledge of the world appears not to be derivable from the circumstances of an episode; rather, the point of the episode lies deeper, in the more abstract relations between concepts. For example, the thematic information involved in the notion of "retaliation" is independent of any particular situation; one can imagine retaliation occurring in a wide variety of settings. A terrorist group retaliating against a government crackdown with a bombing incident is quite different from a child, feeling wronged, tattling on a sibling. However, every episode that embodies the theme of "retaliation" is, at some (more abstract) level, equivalent.

The thematic level of knowledge.

Schank (1982) proposes thematic knowledge structures to account for first, the thematic pattern within an episode, and second, how generalizations are made across episodes that vary greatly in some respects while sharing more abstract similarities. Thematic Organization Points or *TOP*s, are defined as interacting patterns of goals and plans, with certain conditions attached to the pattern. In "retaliation," each side has goals and plans to achieve those goals under the condition of mutual antagonism. *TOP*s are related to earlier versions of "themes" (Abelson, 1973; Schank & Abelson, 1977), and differ from other structures proposed to capture thematic information (e.g. Lehnert, 1981; Wilensky, 1982) in the emphasis on the overall pattern of goal and plan interaction, the importance of the attached conditions, and their functionality as structures in memory.

Table 1 : Sample Episodes Based on a *TAU* Structure.

Story 1: Academia

> Dr. Popoff knew that his graduate student Mike was unhappy with the research facilities available in his department. Mike had requested new equipment on several occasions, but Dr. Popoff always denied Mike's requests. One day, Dr. Popoff found out that Mike had been accepted to study at a rival university. Not wanting to lose a good student, Dr. Popoff hurriedly offered Mike lots of new research equipment. But by then, Mike had already decided to transfer.

Story 2: Wedding Bells

> Phil was in love with his secretary and was well aware that she wanted to marry him. However, Phil was afraid of responsibility, so he kept dating others and made up excuses to postpone the wedding. Finally, his secretary got fed up, began dating, and fell in love with an accountant. When Phil found out, he went to her and proposed marriage, showing her the ring he had bought. But by that time, his secretary was already planning her honeymoon with the accountant.

Test sentences:

Conclusion: by then, Mike had already decided to transfer
Conclusion: his secretary fell in love with an accountant

In the experiments presented in this paper, we examined one kind of *TOP*, namely the Thematic Abstraction Units (*TAU*s) proposed by Dyer (1982). *TAU*s are the patterns of goals and plans reflected in common adages. For example, the adage "Closing the barn door after the horse is gone" expresses the point of the stories in Table 1. The similarity in the two stories involves some general planning information about a common error: waiting too long to execute a plan, causing its failure.

We chose *TAU*s as structures to be used in these experiments for two reasons. First, while *TAU* structures are abstract, they are still reasonably well-defined; Table 2 shows the goal-plan structure, with attached conditions, for the *TAU* of the two stories in Table 1 and the barn door adage (Dyer, 1982).

Table 2: Goal and Plan Structure of a Thematic Affect Unit

TAU-BARN-DOOR

Adage: Closing the barn door after the horse is gone.

(1) x has preservation goal G active since enablement condition C is unsatisfied.
(2) x knows a plan P that will keep G from failing by satisfying C.
(3) x does not execute P and G fails.
x attempts to recover from the failure of G by executing P.
P fails since P is effective for C, but not in recovering from G's failure.
(4) In the future, x must execute P when G is active and C is not satisfied.

Second, the patterns of goal-plan interactions represented in *TAU*s have been shown to be easily recognized by Seifert and Black (1982). Thus, the thematic level of knowledge can be captured by structures that contain relatively abstract information about goal and plan relationships. Further, these structures may organize episodes that contain similar thematic information. The premise that thematic structures are useful in encoding and organizing related episodes suggests that these structures serve as the connection between related episodes.

Connecting thematically similar episodes.

Let's examine a thematic structure and a proposal for how episodes may be stored under it. Given the two episodes in Table 1, and a general structure to encode them, the resulting organization in memory is suggested to be the pattern shown in Figure 1 (Dyer, 1982).

Figure 1.

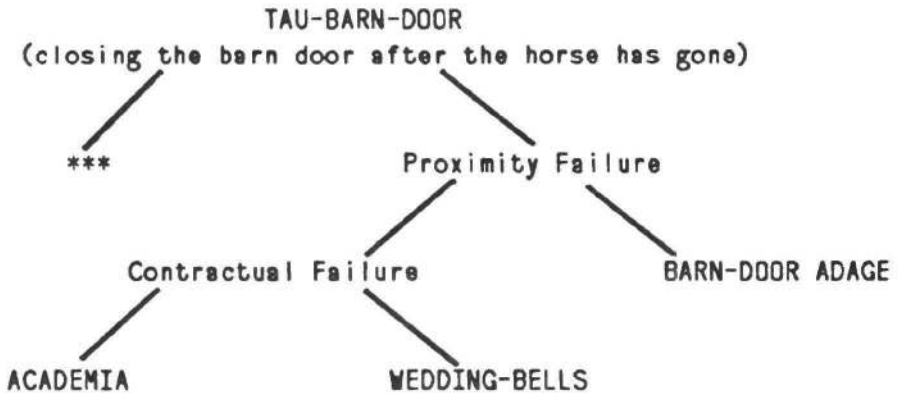


Figure 1. The memory organization of a *TAU* structure and related episodes.

Recalling an episode will activate the associated structure; in addition, this model suggests similarities may result in the activation of not only a structure but a related episode. The role of accessing episodes in

memory has been analyzed by Schank (1982) using the phenomenon of reminding. Reminding occurs when a particular situation causes you to remember another experience that is similar in some way. The relationship between the new input and the old memory retrieved can be at any level of abstraction or similarity. For example, seeing a bearded man in a red suit may "remind" you of Santa Claus, going into Burger King for the first time could "remind" you of McDonald's, and seeing "West Side Story" may "remind" you of "Romeo and Juliet". Schank proposes that in understanding the new situation, you are led to structures in memory that categorize the input; then, under some circumstances, you may find an episode from the past stored in the same way.

Consider the similarities between "West Side Story" and "Romeo and Juliet" (Schank, 1982). The thematic structure is based upon relationships between the goals in the episodes, and interesting deviations in the situation. In these two stories, the two characters are pursuing the same goal while outsiders oppose them. When an episode involves such a complex goal pattern, similar episodes that had been understood using that goal pattern may be brought to mind. Then, the old experience can be further used to aid in processing the current episode. The reminding can provide expectations about the problems the lovers will encounter, ways to prevent making the same mistakes in a similar situation, actions from the old situation that might be possible in the new, warnings of problems to watch for, and predictions about what will happen next. The reminding may also add to understanding by pointing out similarities in the two experiences that hadn't been noticed. In addition, relevant episodes are often poignant illustrations of the generalizations contained in the memory structures. In this way, matching a new situation to a previous experience provides understanding and possibly a way of solving a problem. Without the preservation and access to the old episodes in memory, a system will not be able to change dynamically to learn from the new episodes it encounters (Schank, 1982).

When is reminding based on thematic similarity likely to occur? What kinds of thematic structures are built from episodes? Progress on a theory of memory organization of episodes has relied upon the analysis of protocols of reminding experiences. Examining reminding empirically depends upon developing a methodology that will provide both a naturalistic task which is analogous to reminding and a successful measure of the activation of episodes. In the experimental paradigm reported here, a task analogous to reminding is produced by having subjects study a set of stories based upon thematic structures (old experiences), and then read a series of new stories. By manipulating whether these stories are related in theme, the question of whether the stories are connected based upon their thematic similarity can be addressed. In order to capture the activation of episodes in memory, priming in item recognition is used as a measure of the relatedness of two target episodes in memory (McKoon and Ratcliff, 1980), allowing the empirical examination of episodic reminding.

Experimental Reminding

In two experiments, prestudied stories were paired with two test stories. One of these stories was based on the same thematic structure as the prestudied story; the other was based on a different thematic structure. For both stories, the test sentence was the conclusion of the prestudied story, as shown in the examples in Table 1. We hypothesized that, in the Same-Theme condition, reading a test story with the same thematic structure as the prestudied story may remind subjects of the old story, leading to faster response times for the test sentence.

Method. Detailed discussions of these two experiments are reported in Seifert, McKoon, Abelson, and Ratcliff (1984). There were three phases to Experiments 1 and 2, a pre-study phase, a study-test phase, and a final free recall phase. In the prestudy phase, eight target stories, each of a different *TAU* pattern such as in Table 2, and three practice stories, each from a pool of stories based on similar thematic patterns, were given to subjects to read, answer questions, and then write a one or two sentence summary of each story.

In the study-test phase, stories were presented one word at a time on a microcomputer at a natural reading rate and were followed by a test sentence. All the stories presented were new to the subject, but all the test sentences referred to the target stories presented during the pre-study phase. Eight of the new

stories were paired with the eight prestudied stories so as to have the same thematic pattern (Same-Theme condition), and another eight were paired with the prestudied stories so as to have a different thematic pattern (Different-Theme condition). Thus, each conclusion sentence from a prestudied story was presented for testing twice, once in the Same-Theme condition and once in the Different-Theme condition. Order of presentation was counterbalanced with relatedness using two groups of subjects.

Two different tasks were used as reaction time measures. In Experiment 1, subjects were asked to verify whether test sentences were true according to the story they were from. Negative test sentences were selected from the prestudied story set. In Experiment 2, an identification task was used; subjects had only to press a response key as soon as they could remember which story the test sentence referred to. After responding, they wrote a one-sentence description of the story referred to by the test sentence.

In the final free recall phase of the experiments, subjects were instructed to recall, in any order, the stories from the prestudy phase, and to write an identifying phrase for each story recalled. In addition, subjects in Experiment 2 were instructed to recall the stories from the study-test phase. Eighteen subjects participated in Experiment 1 and eight in Experiment 2.

Results. Data obtained in the pre-study phase (answers to questions about the stories and summaries of the stories) showed that each subject had responded adequately. For the study-test phase, all analyses and statistics for the data for the test sentences were based upon mean response times for each subject and each test sentence in each condition.

In both experiments, responses in the Same-Theme condition were faster than responses in the Different-Theme condition. In Experiment 1 (verification), the mean response time in the Same-Theme condition was 2376 msec (3 % errors), and in the Different-Theme condition, 2554 msec (1 % errors). This difference was significant with subjects as a random variable, $F(1,17) = 11.5$, $p < .01$, and with test sentences as a random variable, $F(1,7) = 5.6$, $p < .05$, though $min\ F'(1,15) = 3.8$, $p < .08$, was marginally significant. The difference in error rates was not significant, F's < 1.

In Experiment 2 (identification) mean response time in the Same-Theme condition was 1253 msec, and in the Different-Theme condition, 1474 msec. These means were significantly different, $min\ F'(1,14) = 5.4$, $p < .05$. All subjects had completed the accuracy check of writing an identifying phrase from the story after hitting the response key.

In the final free recall phase, subjects were able to generate 75% of the prestudied stories in both experiments. In Experiment 2, subjects recalled 27% of the study-test phase stories. Further, the probability of recall for study-test phase stories that matched the prestudied stories in thematic structure was higher than the probability of recall for study-test phase stores that did not match, .35 versus .19. This difference is significant with subjects as a random variable, $F(1,17) = 10.07$, $p < .01$, but not with test sentences as a random variable, $F(1,7) = 2.45$.

Conclusion

These experiments provide strong evidence for the effect of thematic similarity in activating previous episodes. In both the verification task and the simpler identification task, response times for a test sentence from a prestudied story were faster when the story preceding the test sentence matched the test sentence's story in thematic structure. New stories appeared to activate stories already encoded in memory on the basis of their thematic similarity.

Other experiments on the role of thematic similarities between episodes (Seifert, McKoon, Abelson, and Ratcliff, 1984) have indicated the importance of a functional purpose for reminding in the experimental task. This parallels the processing goals in everyday tasks such as problem solving and analogical understanding. The design reported here appears to motivate similar processing within the experimental situation. This technique of using similar episodes to remind the reader of previously learned episodes allows the investigation of the kinds of structures organizing the episodes in memory. Reminding in the experimental context can be used to determine when connections are made between episodes and to

compare types of organizing structures.

Acknowledgements: This research was supported by grants from the Sloan Foundation and the System Development Foundation. We would like to thank Michael G. Dyer and Roger Ratcliff for valuable assistance during the course of this research.

References

Abelson, R. P. The structure of belief systems. In R. C. Schank amd K. M. Colby (Eds.), *Computer models of thought and language*. San Francisco, CA: W. H. Freeman, 1973.

Dyer, M. G. In-depth understanding: A computer model of integrated processing for narrative comprehension. Research Report #219, Department of Computer Science, Yale University, 1982.

Lehnert, W. G. Plot Units and Narrative Summarization. *Cognitive Science*, 1981, *5*, 293-331.

McKoon, G. & Ratcliff, R. Priming in item recognition: The organization of propositions in memory for text. *Journal of Verbal Learning and Verbal Behavior*, 1980, *19*, 369-386.

Schank, R. C. *Dynamic memory: A theory of reminding and learning in computers and people*. New York: Cambridge University Press, 1982.

Schank, R. C. & Abelson, R. P. *Scripts, plans, goals, and understanding*. Hillsdale, NJ: Erlbaum, 1977.

Seifert, C. M. and Black, J. B. Thematic connections between episodes. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*, Rochester, NY, 1983.

Seifert, C. M., McKoon, G., Abelson, R. P., and Ratcliff, R. Memory connections between thematically similar episodes. Technical Report #25, Cognitive Science Program, Yale University, 1984.

Wilensky, R. *Planning and understanding*. Reading, MA: Addison-Wesley, 1983.

The Integration of Goals and Actions in Text Understanding

Noel E. Sharkey & Gordon H. Bower
Department of Psychology
Stanford University

An important part of story understanding is the reader's ability to relate the actions of the characters to their goals. Often the reader is required to keep track of several of a character's goals at the same time. In this paper we investigate some of the processes involved in such tasks. We propose a model which assumes that the relationship between a goal and the various means of fulfilling that goal (e.g. through plans and actions) is represented as an associative network in memory. For our purposes, a goal such as seeking a girlfriend will be represented as a single node in the network (see Figure 1). This node will have links to associated general plan nodes (e.g. CONSULT PROFESSIONAL) and these in turn, to more specific action nodes (e.g. use dating service).

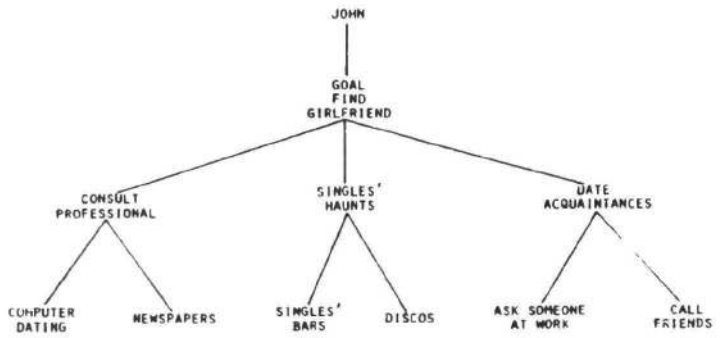


FIG. 1 PARTIAL NETWORK REPRESENTATION FOR A SINGLE GOAL

In these terms, we suppose that a reader comprehends an action by connecting it to an active goal for the actor. In the model, activation spreads out from the goal node to associated general plan nodes and thence throughout the network. At the same time, activation spreads out from the concepts in the stated action. If the goal and the action are related, their paths of activation will eventually intersect. When this occurs, a check is made to see if the stated action is an instantiation of one of the action nodes associated to the goals. This hypothesis predicts that the more goals currently active in memory, the longer it will take an action to be integrated with some one of them (except in special cases to be discussed later). This prediction follows from two assumptions. First, activation will be divided approximately equally among the K active goals for a given character; thus a character node with activation A will send activation A/K down each goal link. Second, we assume that the time required to check whether a stated action instantiates a candidate action node is shorter the greater the activation on that node.

This "goal-fan" effect was reported in a preliminary study by Bower (1982). He found that the more independent goals readers had to keep in mind, the longer it took them to decide whether an action fulfilled one of those goals. We used a similar experimental method which is

described briefly below.

Subjects read a large number of brief vignettes in each of which a series of goals were ascribed to an actor. The goals were presented on a CRT screen, always in the form "<Character-name> wanted: X" (e.g. John wanted: to eat a hamburger.). Each goal was studied for three seconds and, after a one second pause, was replaced by another goal. During the pause the frame "Character-name wanted:" remained on the screen. At the end of each discrete trial (4 goal maximum) a prompt of the form "And so <character-name>" appeared for one second and was then followed by an action statement (e.g. And so John went to MacDonalds). During each vignette the same character preceded each goal. The subject's task was to press a 'yes' button if the action fulfilled some one of the goals which had just been presented and a 'no' button if it did not. This time to respond was the dependent measure in the studies.

We varied the number of goals that subjects had to monitor and the relationship between these goals. In Experiment 1, subjects were presented with either one or three goals on each trial. In the three goal conditions the goals were either independent of one another or they each could be satisfied by the same action. We call the latter condition Goal-Overlap (c.f. Wilensky, 1983). For example, the goals of wanting to live an outdoor life, to work in a forest, and to develop his physical strength can all be fulfilled by the action of becoming a lumberjack. As before, we expect the Three-Independent-Goals condition to take a longer time to verify due to greater dispersion of activation. In the Goal-Overlap condition the total activation divides equally among the three goals. However, as shown in Figure 2, the activation from the three goals re-converges on the overlapping action node. This node will then have approximately the same amount of activation as it would in the presence of a single goal. From this reasoning we expect no response time difference between the One-Goal and the Goal-Overlap conditions but both conditions should produce faster responses than the Three-Independent-Goal condition.

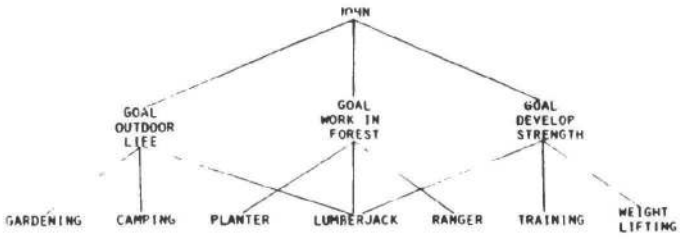


FIG. 2 PARTIAL NETWORK REPRESENTATION FOR THREE OVERLAPPING GOALS

The results accorded with these predictions. For both the 'yes' and 'no' responses verification times were as fast for an action satisfying three overlapping goals as for an action satisfying a single goal. Both were significantly faster than the time to verify an action satisfying a one of three independent goals. Effects were significant beyond the .001 level by MinF.

In a second experiment subjects were presented with either two or four goals. The goals were either independent or they conflicted with each other in pairs e.g. "John wanted hamburgers for dinner this evening. John wanted to eat chinese food this evening." So the four conditions studied were: 1 or 2 pairs of conflicting goals and 1 or 2 pairs of independent goals. In the network model the activation pattern for a goal conflict pair differs somewhat from that for

independent goals. Figure 3 shows a pair of conflicting goals that share many thematically related concepts and high level plans (e.g. EAT FOOD). In Figure 3 we can see that activation initially gets divided between a pair of conflicting goals and then re-converges on the thematically related plan nodes. If the activation level on a goal node is a_g, then the activation on each of the plan nodes underneath it will be $a_p = (a_g/k_g).K$, where k_g = the total number of plan nodes, and K = the number of conflicting goals. Thus these plan nodes will receive approximately twice as much activation as equivalent nodes in the case of two independent goals. However, the specific instantiations of these plans are mutually exclusive and so activation will divide again at these nodes. In this way activation on the action nodes may be at an equivalent level in the case of conflicting and independent goals.

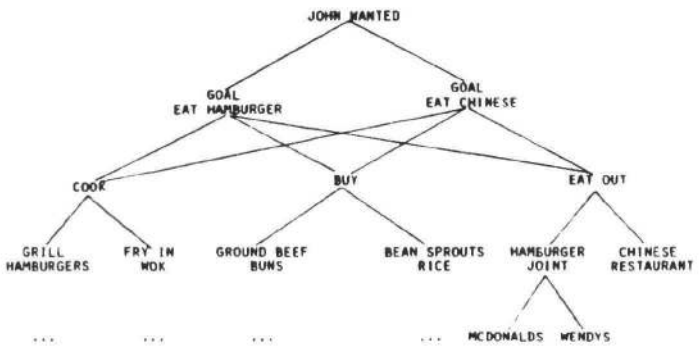


FIG. 3 PARTIAL NETWORK REPRESENTATION FOR TWO CONFLICTING GOALS

As in Experiment 1, we hypothesize that the time a person waits before rejecting a foil in the Goal-Conflict conditions should be governed by the activation levels at the thematically related nodes. Put simply, if an action doesn't fit the theme of a conflict, then it must be an unrelated foil. On this basis, foils should be rejected sooner when they are preceded by conflicting goals than by independent goals. Following similar reasoning, the intersection of activation for 'yes' decisions should be faster when the action probe is preceded by conflicting goals. However, given that actions associated with conflicting goals are mutually exclusive, we have good reason to believe that the evaluation of the intersection may be slowed.

The results came out as predicted. Subjects were faster to reject foils which were preceded by conflicting goals rather than independent goals. Decisions were always slower with 4 goals than with 2 goals, replicating earlier results. Goal type and goal number did not interact significantly. For 'yes' responses 4 goals caused slower times than 2 goals; conflicting goal pairs were slightly but not significantly faster than independent goal pairs. Although these results fit our predictions, further experimentation is needed to bolster our claims.

In summary, we proposed a spreading activation network model of how people relate actions to goals. The results from our Overlapping and Conflicting goal experiments provide some initial support for the model. We are currently pursuing follow-up experiments which use thematically related foils. Such foils should prevent subjects from simply using the activation level at the overlapping nodes to reject the foils. Thus we would expect conflict goal pairs to take as long to reject as independent goal pairs.

Thanks to Amanda Sharkey for making a significant contribution to writing materials and collecting and analyzing the data.

References

Bower, G.H. Plans and Goals in Understanding Episodes, In August Flammer and Walter Kintsch (Eds.) Discourse Processing, New York: North-Holland Publishing Company, 1982.

Wilensky, R. Planning and Understanding: A computational approach to human reasoning. Massachusetts: Addison-Wesley, 1983.

THE MATHEMATICAL ROLE OF SELF-CONSISTENCY IN PARALLEL COMPUTATION

Paul Smolensky

Institute for Cognitive Science C-015
University of California, San Diego
La Jolla, CA 92093

Analysis of Emergent Properties of Neural Systems

One approach to the mind/body problem is to view the description of mind as a higher level description of brain; to view psychological principles as emergent properties of neural systems. Certainly before such a view can be scientifically tested, a better understanding of both brain and mind must be established. However enough is already known about each to make feasibility studies possible.

What methodology is capable of analyzing the emergent properties of large complex systems of interacting elements? One discipline where this job needs to be done is statistical physics, where large-scale properties of matter are derived mathematically from the principles believed to govern the interactions of molecular and sub-molecular constituents.

Is it possible to apply similar kinds of mathematical analysis to deduce emergent properties of neural systems? Although the principles governing neuronal interaction are by no means as well understood as those governing particles, models that abstract some of the characteristics of neural networks have been studied for some time. Hopfield (1982) has shown that with certain modifications, standard neural models can be analyzed with mathematics much like that of statistical physics, and emergent properties can be analyzed.

One of the central concepts in statistical physics is *temperature*. The utility of this concept in performing difficult computations has been shown by Kirkpatrick et. al. (1983). However the most important concept in statistical physics, as in all branches of physics, is that of *energy*. The meaning of "energy" in the computational context is not obvious; rather than a computational interpretation, Hopfield offered a general formula for the "energy" of a neural net while Kirkpatrick et. al. hand crafted "energy" formulae for their particular computations.

The application of statistical physics concepts to computation is now a rather active field of study (Hinton and Sejnowski, 1983; Hofstadter, 1983; Geman and Geman, 1983). To provide a solid foundation for this analysis, what is required in my opinion is *an interpretation of "energy" that establishes a deep connection between the formalism of statistical physics and the central problems of cognition.*

The help of David Rumelhart, Francis Crick, and other members of the UCSD Parallel Distributed Processing research group is gratefully acknowledged.

This research was supported by a grant from the System Development Foundation and by contract N00014-79-C-0323, NR 667-437 with the Personnel and Training Research Programs of the Office of Naval Research.

In this paper I will present the interpretation of "energy" that lies at the heart of a general computational approach I have been developing independently of the work of those interested in neural nets or in particular difficult computations. In this interpretation, *"energy" is a measure of the self-consistency of a computational state*. In place of the term "energy", which emphasizes the physical analogy, or the more technical term "Hamiltonian", which serves only to recall history and account for the physicist's notation H, I choose to foreground the measurement of self-consistency by using the term *harmony function*, denoted H. The general framework, *harmony theory*, is described in Smolensky (1984); an analysis of learning using this theory is begun in Smolensky (1983), and an application of the theory to modelling qualitative analysis of a simple electric circuit (with a discussion of the model's emergent properties) is described in Riley and Smolensky (1984). In this paper I will focus on the computational meaning of harmony, passing quickly over other aspects of the theory. The treatment will be very informal; for more formal presentations the reader is referred to the previously cited papers.

The Role of Harmony in Computation

Before considering how the harmony function is *defined*, we start with a discussion of how the harmony function is *used* during computation. The basic idea can be framed at a very general level. During computation, search for an answer is guided by a measure of "goodness" of possible answers: the harmony function H is that measure. The search is stochastic; the computation is a Monte Carlo random walk through the solution space under the guidance of H. The random walk is designed so that eventually, the probability at any moment of visiting a point p in the solution space is given by the *canonical distribution*:

$$\text{prob}(p) = Ne^{H(p)/T}$$

N is the constant needed to normalize the probabilities so that they sum to one. T is a global parameter that determines the spread in the probability distribution.

The canonical distribution is the only continuous relationship between H and probability that correctly treats the independence of components of a computation. The canonical distribution also happens to be the distribution on which most of statistical physics is based. (This is no coincidence, as the notion of independent subsystem in physics maps onto that of independent subcomputations.) There is an isomorphism that maps the harmony function into minus the Hamiltonian (energy) function, and T into temperature. This suggests calling T the *computational temperature* of the system.

In physics, the Hamiltonian determines what states are most probable: the states with lowest energy are most probable at all temperatures, and states of high energy have negligible probability except at high temperatures. In harmony theory, the harmony function determines what states are most probable: the states with highest harmony are most probable at all computational temperatures, and states of low harmony have negligible probability except at high temperatures. T can be thought of as setting the *scale* for what constitutes significant differences in harmony values. In fact, the ratio of probabilities of two states is $e^{\Delta H/T}$, where ΔH is the difference in harmony between the states. If this difference is small compared to T, the ratio of probabilities will be close to one; if ΔH is large compared to T, the state with higher harmony will be many times more probable.

The goal of the computation is to find the state of highest harmony. This means, in particular, that the state of next highest harmony should be much less likely. This requires that T be small compared to the harmony difference between the two highest levels of harmony.

We could simply set T to be such a low value and be done with it. However, this is not a practical search procedure. The Monte Carlo procedure will, if let run long enough, visit points with the probabilities given by the canonical distribution. However, the time required to reach this "thermal equilibrium" grows extremely rapidly as T is lowered. A more practical way of zeroing in on the state of highest harmony is to start with a high temperature and gradually lower it. Early in the search, only large harmony differences are significant, and the system quickly makes a crude cut at the problem, avoiding states of extremely low harmony. As the system cools down, smaller harmony differences become significant, and more and more states are avoided as the search focusses on states with harmonies close to the maximal value. If the cooling is done gently, the state of maximal harmony should be found in *much* less time than by giving T a constant low value.

The Relation of Harmony to the Environment

We have discussed a stochastic search technique that will find states of high harmony. But how do we design the function H so that the states with high H values give the correct solutions to problems? Now we must discuss the sense in which H measures self-consistency.

The "correct" answer to problems are often those that satisfy a set of rules. In the circuit analysis problem considered by Riley and Smolensky, for example, the rules are the physical laws of simple circuits. Any system that can correctly solve problems such as this must in some sense have a representation of the rules. In harmony theory, the rules are encoded in the harmony function. The question is, how are these rules encoded, and how can a system develop an appropriate harmony function through experience?

Of course most cognitive tasks are not as strictly governed by rules as is formal problem solving. Yet all cognition hinges on the *exploitation of regularities in the environment*, even if those regularities are less formal than Ohm's Law. Cognition enables organisms to do the *completion task*: take some limited information about the current state of their environment and make reasonable guesses about what else is likely to occur in the environment. That is, given *some* of the features that specify the environmental state, the organism can make reasonable guesses about missing features.

In harmony theory, the "rules" applied during the completion task are simply *statements that certain features can co-occur in the environment*. In the circuit application, for example, in place of a symbolic version of Ohm's Law, $V = IR$, there are many "rules" that each record a single combination of qualitative changes in V, I, and R that are consistent with the law. These "rules" can in fact be thought of as *memory traces* that might be left behind by individual experiences in the environment in which the regularities hold.

Here is the general idea of how to set up a harmony function for performing the completion task in a given environment. Imagine the system experiencing many encounters with the environment; each leaves many traces that each record some of the features that co-occurred. When partial information about the current state of the environment is given in a completion problem, the harmony of a possible completion of that information is *the overall consistency between that completion and the set of all traces*. To spell this out, we consider first how the traces are determined and then how the "overall consistency" is computed.

The traces can be produced automatically by simulating exposure to an environment, or they can be produced manually by the modeller. The latter technique was used in the circuit problem: each trace was chosen to be an allowed combination of qualitative changes in the circuit quantities appearing in a single circuit law. The automatic generation of traces is yet to be explored; the idea is

that traces would be produced in a random fashion (guided by the degree to which potential traces would enhance system harmony); the *statistical properties* of the resulting set of traces would then govern the emergent behavior of the system.

How is "the overall consistency between a completion and the set of all traces" computed? The idea here is that for each trace, a decision needs to be made whether the instance it recorded is relevant to the current situation or not. Borrowing the usage of schema theory, a match between part of a trace and a completed set of environmental features can cause the trace to become *active*. The "overall consistency" – the harmony – of a completion is the sum over all active traces of a measure h of the degree of match between the trace and the completion. A simple definition of h is the number of features in the completion that match the trace, minus the number that do not match. (A slightly more complicated definition of h was used in the circuit analysis model.)

There are now two kinds of variables used in the computation: features of the environmental state, and activation values for traces. The processing has two components: computing the harmony values of possible completions, and making corresponding random decisions about which completions to visit. Computation of the harmony value requires deciding which traces to activate, which requires computing the quality of match h between traces and the completion. Just as the Monte Carlo search is used to decide what completions to visit, it can be used to decide what traces to activate. So using the traces to define the harmony of completions leads naturally to extending the search space to include both environmental feature variables *and* trace activation values.

The Network Interpretation: A Computer Implementation

It is useful to represent the computation by a network like that shown in Figure 1, which shows a portion of the network for the circuit model. The activation variables are represented by nodes in the upper layer; each corresponds to a trace. The environmental feature variables are represented by nodes in the lower layer. There are connections between a trace variable and all the environmental features it incorporates. For simplicity all variables (nodes) are taken to have binary values: trace activation nodes have values *active* and *inactive*; environmental feature nodes have values *present* and *absent*.

The Monte Carlo search in this network representation proceeds as follows. Initially a high temperature T is chosen, all the traces are set inactive, the environmental features are permanently assigned their given values, and the remaining environmental feature variables are assigned random initial values. Then processing begins. A node is selected at random (but not one of the given features). Next the difference ΔH between the overall network harmonies that would result from the two possible values for the node is computed. This computation, it turns out, can in principle be performed in the node itself, for the only quantities needed are those to which the node is connected. Finally, the node randomly selects a new value, using as the ratio of probabilities for the two values $e^{\Delta H/T}$. The process of selecting a node and selecting a value for that node is iterated while the temperature T is gradually lowered according to some schedule.

The repeated selection of nodes and assignment of new values can be viewed (following Hopfield) as the asychronous processing of processors located at the nodes and running in parallel. The relation between this parallel processing network and those considered by Hopfield and Hinton and Sejnowski is that the harmony model has a special architecture: there are two classes of nodes, and connections between but not within the two classes. The formula for harmony turns out to be minus that for Hopfield's network "energy", taking into account the special architecture and the numerical assignments *active* $= 1$, *inactive* $= 0$; *present* $= 1$, *absent* $= -1$.

Comments on Neural Implementation

Since harmony theory is computationally- rather than neurally-inspired, the relation between the harmony network and neural networks has not been developed. However the close resemblance of the harmony network to Hopfield's neural network might suggest that harmony nodes correspond to neurons, so a brief comment is appropriate. While it does not seem unreasonable in principle to identify environmental feature nodes with neurons, it is *not* reasonable to to identify trace nodes with neurons. Indeed, I imagine that each trace is distributed over the synapses of the neurons corresponding to the environmental features involved in that trace. "Activation" of the trace might correspond to a feedback-mediated rapid enhancement of the strengths of these synapses, as in von der Malsberg (1981). In this sense, even the activation of traces, a primitive operation in the theory as presently formulated, may be an emergent property of synaptic dynamics.

Even without a precise specification of the relation between harmony networks and neurons, harmony theory offers a mathematical framework within which to explore the emergence of mind from brain-like processing. The isomorphism between computation and statistical physics which it represents rests on the identification of self-consistency – harmony – as playing a central role isomorphic to that played by energy in physics.

References

S. Geman and D. Geman (1983). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. Manuscript.

G. E. Hinton and T. J. Sejnowski (1983). Analyzing cooperative computation. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*. Rochester, NY.

D. R. Hofstadter (1983). The architecture of Jumbo. *Proceedings of the International Machine Learning Workshop*. Monticello, IL.

J. J. Hopfield (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences USA*, 79, 2554-8.

S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi (1983). Optimization by simulated annealing. *Science*, 220, 671-80.

P. Smolensky (1983). Schema selection and stochastic inference in modular environments. *Proceedings of the National Conference on Artificial Intelligence*. Washington, DC.

P. Smolensky (1984). Harmony theory: thermal parallel models in a computational context. Manuscript.

M. S. Riley and P. Smolensky (1984). A parallel model of (sequential) problem solving. Submitted to the Sixth Annual Conference of the Cognitive Science Society. Boulder, CO.

C. von der Malsburg (1981). The correlation theory of brain function. Internal report 81-2. Department of Neurobiology, Max Planck Institute for Biophysical Chemistry, Gottingen, W. Germany.

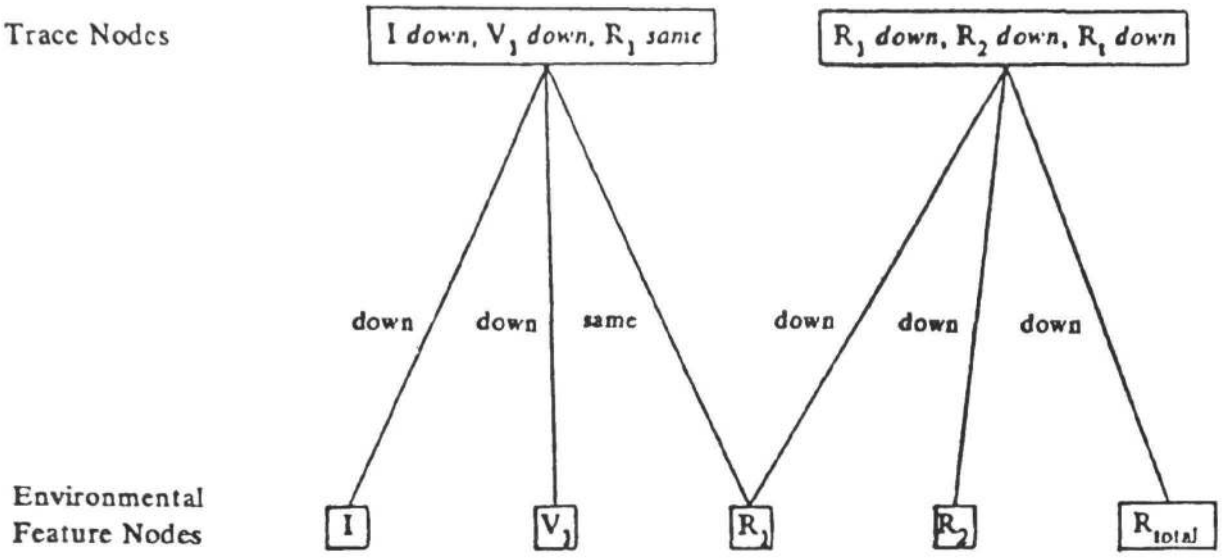


Figure 1. A portion of the network representation of the circuit analysis model (from Riley an Smolensky). [The values *up*, *down*, *same* for environmental features (circuit variable changes) are actually represented by using two binary nodes for each variable.]

Intent to Deceive:
On Creating Deceptions

Gregory B. Taylor

Artificial Intelligence Project
Department of Information and Computer Science
University of California
Irvine, Ca. 92717

ABSTRACT

Counterplanning can be successfully used against most methods of resolving goal conflicts. However, if one's intentions are disguised by deception then an opposing actor will use incorrect counterplanning or possibly none at all. This paper describes two components in the creation of a deception, the deception type and the enablement type, the range of their possible values, and how the selection of each can be used to create different deceptions for the same situation.

1. INTRODUCTION

Much work has been directed at understanding how people interact in a planned and intentional manner to resolve their goal conflicts (Schank [1977], Wilensky [1978]). Consider the following example,

> [1] John and Mary, brother and sister, wanted to watch different tv programs at the same time. There was only one tv set and both knew the other wasn't going to give in. John threatened to hit Mary if she did not let him watch his program.

The possible plans for resolving John's conflict can be ordered based on their likelihood of success and their difficulty in execution. Mary will counterplan against John based on her knowledge of his plan to resolve the conflict between them (Carbonell [1979]). For example, the story might of ended,

> [1a] Mary told their mother that John had threatened her. The mother sent John to his room.

John's ignorance of Mary's possible counterplans resulted in his goal failure.

Deceptions are a class of plans where the intentions of the deceiver are purposefully not communicated thereby preventing successful counterplanning. For example, John could have deceived Mary by,

> [1b] John mentioned to Mary that he had seen several girls going to the theatre to see Robert Redford making a public appearance. Mary immediately left, leaving John to watch his program.

John led Mary to believe that his intentions were to give her a chance to meet Robert Redford, when in fact he simply wanted her out of the house.

This work was supported in part by the Naval Ocean Systems Center under contract N00123-81-C-1078.

2. HOW DID JOHN LIE?

Here we briefly raise the question, "HOW DID JOHN KNOW TO LIE?" As mentioned earlier, John's selection of a plan to resolve the conflict (to lie) is based on how likely that play is to succeed. Deceptions are most successful when the relationship between two people is seen as a benevolent from the perspective of the person to be deceived (Mary) and malice from the perspective of the deceiver (John). For example, does she trust him, does he dislike her, etc. Opportunistic aspects of deceptions also exist and can be recognized by characters.

In this section we introduce the main topic of this paper - the two components of a deception, the deception type and the enablement type, and show how they combine to create a deception such as the one in 1b.

2.1 Deception types

There are four major classes of deception types or d-types. Each class of d-types is used to either achieve the deceiver's goal or cause the deceived person plans to fail. The deception occurs when the deceived person is not aware of what is happening.

D-types in the first class select a precondition of the deceived person's plan to be negated. By "undoing" a precondition that is difficult to re-establish the deceived person's plan will fail. The original deception goal is reduced to negating this precondition. The negation of this precondition becomes the new deception goal. If the actual goal of the deception is to prevent some action on the part of the deceived person then this d-type is very useful. The d-types are described in the first person (I deceive you).

1. Undo preconditions for object within a plan. The new deception goal is to make you believe that the particular desired attributes for some required object in your plan no longer exist. For example, if I want you to leave the apple pie so that I can eat it, I might tell you that the pie is rotten; if John wants Mary not to watch television, he convinces her that the set is broken.

2. Undo delta goal preconditions (Schank [1977]). The new deception goal suggested here is to undo any one of the simple preconditions commonly found within the deceived person's plan such as control of an object, being at a location or knowledge of some simple fact.

Because delta goals appear in most plans, the conflict with the actual goal is difficult to notice. In 1b, John uses the undo delta goal d-type (PROX is selected). The new goal to be achieved is to undo Mary near the television, i.e. to make Mary leave the television area.

D-types in the second class of the four classes are useful when the deception goal is a simple action and incorporate it into some larger action (backwards from the previous class of d-types). The simple action is usually a delta goal, although not necessarily. It is often the result of reducing an original deception goal using a d-type from the first class of d-types. The d-types of the second class are described below (again in the first person).

3. Challenge. I identify a plan that contains you acting out the present deception goal as a small part or precondition. The plan should contain use of some boastful attribute, e.g. strength, quickness, singing, etc. I use reverse psychology in communicating the plan. For example if a mother wants her son to empty the trash, she says, "I bet you're not strong enough to empty the trash with one arm."

4. Demonstration. Similar to challenge but I communicate the plan in a straight forward manner - no reverse psychology.

5. Instantiate common context. We call a commonly done activity a context. I locate one of your contexts that has the present deception goal as one of its preconditions or steps in execution. I communicate or instantiate the goal or intention of the context to you and play out the context UNTIL the deception goal is reached.

6. Posit better goal. Very similar to instantiate common context, except that the activity is not commonly done. Here, I must propose an explicit goal for you to pursue. It must be of higher importance to you than your present goal. Often the new goal is just an instantiation of the present goal with some parameter changed to effect the greater value.

The common feature of these d-types is that the plan selected should contain some boastful attribute that can be used to "emotionally push" the deceived person into enacting the plan.

Recall in 1b, John's new deception goal is to make Mary leave the television area. Here John uses the posit better goal d-type. The goal selected is the same as Mary's present goal - that of entertainment. However the restrictions on the exact construction of the goal are that the location of the entertainment be away from the television and that it be more "interesting" than the program that Mary was going to watch. From these descriptions, John creates the goal that Mary should go to the theatre to see Robert Redford. The deception is far from complete, he still he still must make her believe that Robert Redford is at the theatre. However, it is unlikely that she will see the connection between Robert Redford at the theatre and watching TV.

2.1.1 Structure of a deception

John's deception has used two d-types. We can represent the present structure of what we have analyzed.

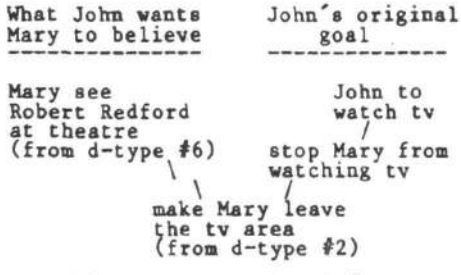


Figure 2-1: Structure of John's deception

The number of d-types used in a deception depends on how many are necessary before a "believable" deception is reached. That is, after John's first d-type, it is unlikely that Mary will leave the television. Making a "believable reason" for her to leave is accomplished by "stretching" the distance from the original deception goal to the final deception goal.

The length of the deception (number of d-types used) obscures John's original intentions/goals. Also, as Mary views his goals as "less conflicting" to hers, his d-types become more believable to her. She believes that John is no longer out to compete with her for the television, but instead he is her benefactor.

2.1.2 Other d-types

D-types in the third class of d-types distract the deceived person by communicating the existence of some very high priority goal. These d-types are similar to the posit better goal d-type except that the new goal presented is one of such high priority that the decision of which goal to pursue to not made, rather achieving the previous goal is postponed. These d-types are listed below, again in the first person.

7. Attract attention. I communicate the existence of a very high opportunistic goal that you can easily achieve such as money on the ground or seeing a beautiful girl walking down the street.

8. Crisis goal violation. I communicate the possible violation of one of your crisis goals such as maintain-health.

D-types in the final class deal with more complicated goal modification and interaction. Deceptions are introduced by convincing the deceived person of false goal relationships. For example,

[2] John was spraying the garden with pesticides. His wife who didn't like chemicals told him that she knew from biology that pesticides kill cats and that she was thinking about getting a cat. He stopped spraying.

Mary made John believe that spraying prevented a higher goal of her happiness (having a cat) from being achieved. Because of length limitations we are unable to individually describe these last d-types.

Another example of a deception in this class might be if we want to convince someone that a goal is undesirable or can't be achieved we might first show them "the entire" set of plans that can achieve their goal and then show how each plan is undesirable or can't be achieved. The deception in such a strategy could be in an incomplete breakdown of the possible plans, showing a good plan to be bad, or showing how an achievable plan will fail.

2.2 Enablement types

In example 1b, John's second d-type still leaves him with the deception goal of how to make Mary believe that Robert Redford is at the theatre. The d-types have been methods of altering the deception goal. Introducing the false fact occurs in the enablement type or e-type.

In general, e-types appeal to the emotions. Enablement types or e-types have two functions. First, appealing to the emotions of peer pressure or the feeling of "not wanting to miss anything" as in John's deception. Secondly, e-types cause the deceiver to believe different attributes about the deceiver. Either general attributes like feelings of trust and friendship which result in a decreased level of suspicion, or specific attributes regarding a particular piece of information. The e-types are listed below, again in the first person.

1. Others say or do. I make you believe some fact because I tell you that others believe it. For example, John said to Mary that "he had seen several other girls ...".

2. Complement you. I can say to you, "You are so nice ..." or complement you by comparing you favorably to your enemy. Also included are attributing your resent goal failures to your enemies.

3. Do favor. I can help you achieve a goal. I chose a goal where my assistance is necessary to achieve a goal that you are currently pursuing or have abandoned because of a plan failure. For example, "Would you like some help on your calculus problems?"

4. Knowledge and experience. I can convince you that I have some specific knowledge. For example, "I know how a tv works 'cause I took a class ... and this one is busted." When in fact it is simply unplugged.

5. Experience yielding specific attributes. I can "prove" to you that some past experience has happened to me by relating specific details of the experience. The inference that relates the desired attributes to the experience must be known to the person being deceived. For example, if I am a woman and know men only seriously date women who have dated before, I might carry a locket showing a picture of a man to whom I "would claim" I was engaged.

In John's deception of Mary, he has selected e-type #1. John tells Mary that "others are doing" the same goal (constructed using the previous d-types) he suggests for her. The selection of e-types and their "execution" is also aided by the previously used d-types. For example, the complement e-type is suggested by the demonstrate d-type as in the story below from Firman and Maltby [1918].

[3] A sparrow sitting on a log noticed a robin on a branch directly above him holding a worm in his mouth. The sparrow said to the robin, "Robin, you sing the most beautiful songs in all the forest - won't you sing for me now?". The robin always willing to show off opened his mouth to begin singing. The worm immediately fell out of the robin's mouth and dropped to the ground next to the sparrow. The sparrow quickly ate the worm and left.

The d-type demo of singing suggested the complement e-type along with what specifically to complement.

3. FUTURE WORK

A program is being written to use the first three classes of d-types with all the e-types to construct different deceptions. Each d-type and e-type will be a procedure that builds up the representation for the deception. The program will eventually be integrated into a simulation environment such as Tale-spin (Meehan [1976]).

3.1 A Deception Matrix

The current set of d-types and e-types can be used to create a deception matrix that results in every possible deception for a given situation. The information we hope to obtain from such an analysis includes how complete the d-types and e-types are in creating believable deceptions and which combinations tend to produce deceptions most nearly to those that humans produce.

3.2 Better d-types

Examples of deceptions that need to be understood in greater detail include third party deceptions. For example,

[4] John's love was not returned, she loved Bill instead. John wrote her a letter saying it was all over and signed it Bill.

Complex deceptions not using the d-types discussed exist and must be studied. For example, from our original television example, John's deception might have been,

[1c] John hid the tv guide from Mary and told her that her program had been cancelled. She left to go play.

How did he know to hide the television guide to make his deception work?

The forth class of d-types that we mentioned will probably always need more work. We have tried to categorize this class, however it still remains full of the most complex deceptions involving goal relationships.

3.3 Better e-types

Carbonell [1979] has used some basic personality traits to describe an individuals method of goal pursuit. But how do these same traits influence other goal interactions and believability? Also of interest are psychological theories of how people can be made to feel friendly towards others based only on common experiences and/or friends.

4. CONCLUSIONS

The role of identifying a set of d-types and e-types is not to necessarily produce every possible deception, only a large set of varying types of deceptions. To this end we have already succeeded; with 8 d-types and 5 e-types the number of possible deceptions for any situation is forty. However, in most cases only a few of these are "acceptable". Our future work will focus on increasing the possible deception plans in order to select the best deception plan.

REFERENCES

[1] Bruce, B. "Analysis of Interacting Plans as a Guide to the Understanding of Story Structure". Poetics 9 (1980) pp. 295-311.

[2] Carbonell, J. Subjective Understanding: Computer Models of Belief Systems. Ph.D. thesis. Yale Computer Science Department Research Report 150, 1979.

[3] Firman, S.G. and E.H. Maltby. 1918. The Winston readers: first reader. Philadelphia: Winston.

[4] Meehan, J. The Metanovel: Writing Stories by Computer. Yale University, Computer Science Department, Ph.D. thesis 1976.

[5] Schank, R. and Abelson R. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, N.J, 1977.

[6] Wilensky, Robert Understanding Goal Based Stories. Ph.D. thesis. Research Report 140, Yale University, Department of Computer Science, Yale, 1978.

RULES FOR CONCEPTUAL COMBINATION

Paul Thagard
Philosophy, University of Michigan, Dearborn
Psychology, University of Michigan, Ann Arbor
February, 1984

Fodor (1981) and Osherson and Smith (1981) have claimed that interpretations of concepts as prototypes encounter problems in dealing with combined concepts such as "striped apple" and "brown cow". This paper offers a theory of how concepts, construed as prototypes, can be combined. The theory takes the form of three kinds of rules for selecting what elements of the component concepts will be carried over into the new one. Pure rules take into account only the prior elements of the components. Data-driven rules are contextual in that they employ features of prospective instances of new concepts. Finally, goal-directed rules are contextual in a larger sense, in that they take into account the problems and goals of the inductive system.

A theory of conceptual combination requires that concepts have components which can be used to form new concepts. This assumption is rejected by some who want to treat concepts as unitary nodes, atomic in the original sense of indivisible (e.g. Fodor 1981). Such writers are reduced to silence about how new concepts might arise. The justification for considering concepts as componential is empirical: the assumption enables us to account for a variety of empirical phenomena.

But what are those components? I shall adopt the terminology of Minsky (1975) and treat concepts as frames which are data structures consisting of slots. Such structures can be easily implemented in computer programs (Winston and Horn 1981).[1] A frame contains information about the typical characteristics of a kind of thing; for example, the frame for dog will contain a slot with the information that dogs typically have four legs. It is crucial that the slots need not contain definitional information. Having four legs is neither a necessary nor sufficient condition for dogness, but is nevertheless typical and should generate an expectation. We therefore say that the default value for the number of legs of a dog is four. The slots in the frame for dog do not constitute a definition of dog, but contain lots of information about what is typical of dogs or what it is useful to expect about dogs. Slots in the concept of dog will generally contain default values, not actual values which must hold of all dogs universally. But some actual values may be included, for example that dogs are warm-blooded. There is thus no problem in seeing a concept as containing some slots which involve features which are in fact definitional, but it would be a major mistake to suppose that such slots, if available, would exhaust the meaning of the concept. Looser connections of the sort established by additional default values also matter.

2 Pure, Concept-driven Rules

Definition is the epitome of pure conceptual combination, independent of context. Suppose you have necessary and sufficient conditions for existing concepts C_1 and C_2. Then it is simple to define

the new concept C_3 whose set of necessary and sufficient conditions is just the union of the set of necessary and sufficient conditions for the donor concepts. For example, if we have definitions of "square" and "table", the concept of "square table" is formed merely by amalgamating the existing definitions. However, such definitions can be hard to come by, and all we often have to work with in amalgamating concepts are default expectations rather than defining conditions. Because the expectations generated by combined concepts may conflict, conceptual combination requires complex processes of reconciliation. To repeat an example from Osherson and Smith (1981), our concept of a striped apple is no simple sum of "striped" and "apple", since we expect apples to be green or red. As a result, any instance of a striped apple is more typical of the concept "striped apple" than it is of either "striped" or "apple". How, then, do we combine "striped" and "apple" into "striped apple"?

The following very simple rule suffices:

R1. Actual values drive out defaults.

The concept of an apple contains a slot which sets up the expectation that an apple will be red or green or some combination of those colors, but this expectation is not definitional: a golden delicious is still an apple. The adjective "striped" however incorporates an expectation about coloring which is more than a default, since, to put it tritely. something has to be striped to be striped. Hence this definitional expectation overrides the merely default expectation found in the apple concept. In most adjective-noun combinations of this sort, the actual value found in the adjectival concept will drive out the merely default value in the noun. Green cows are green.

Most conceptual combination will not be so simple. Consider an example of Tversky and Kahneman (1983). They show that subjects will often violate the conjunction law of probability, which says that the probability of the conjunction of two propositions is always less than the probability of either conjunct. They gave subjects a description of a woman Linda who had been a philosophy major, was outspoken, bright, and concerned with issues of discrimination and social justice. Then they asked how probable subjects would estimate her to be 1) a feminist 2) a bank teller and 3) a feminist and a bank teller. Unsurprisingly subjects thought it more probable that she was a feminist than a bank teller, but the startling result, violating the conjunction law for probabilities, is that subjects think it more probable that she is a feminist bank teller than that she is a bank teller simpliciter.

According to Tversky and Kahneman, subjects think that Linda is more probably a feminist bank teller than a bank teller because the former category is more <u>representative</u> of Linda. I shall describe a rule for conceptual combination based on representativeness below: such a rule will be data-driven since the description of Linda appears to play a role in how people construct the new concept of feminist bank teller. In this example, however, conceptual combination should not be data-driven, since subjects are not told that Linda is a feminist bank

teller, only asked whether she might be. A normatively correct rule of conceptual combination should ignore Linda.

An appropriately pure rule can be formed on the basis of considerations of variability similar to those which play a role in assessing the degree of confirmation of a generalization (Thagard and Nisbett 1982; Nisbett, Krantz, Jepson, and Kunda 1983). Suppose the slot in the new concept of feminist bank teller under dispute concerns political activity. Here we have a case of real conflict, since our default expectations are that feminists will be politically active but that bank tellers will not be. R2 resolves the conflict by saying:

R2 On a given dimension, carry over the value from the donor concept which is less variable on that dimension.

In the case of feminist bank teller, we expect that feminists are more consistently politically active than bank tellers are politically inactive. Hence the slot in the concept "feminist bank teller" for political activity should contain the expectation that feminist bank tellers will be politically active. The description of Linda fits this expectation better than it does the expectations established by the bank teller concept alone.

A third rule of pure conceptual combination is necessarily more vague. We can expect that in some concepts slots are rules are linked to each other, developing connected expectations. For example, a concept concerning a kind of physical object which has a value for size is also likely to have a value for weight. Conceptual combination will want to preserve such linkages:

R3 If the new concept C_3 will contain the slot $C_{1,j}$ and $C_{1,k}$ is linked to that slot, then include $C_{1,k}$ in C_3.

The operation of this rule assumes that the representation of concepts will include some expression of linkages between slots.

3 Data-driven Rules

Conceptual combination requires the reconciliation of conflicting expectations, but there is no reason that the reconciliation should have to be a function of the donor concepts alone. Conceptual combination is selective: for most concepts, occasions of combination will simply never arise. You probably will never have occasion to think of Mongolian watermelon eaters. When occasions of combination do arise, they will do so in a particular context, and the context can help to govern default reconciliation.

The simplest sort of contextual factor consists of instances of the prospective concept. Suppose C_1 and C_2 are being combined to form C_3, and some slot is incompatible between the two donor concepts. For example, upon meeting a Canadian violinist, you are pressed to combine your two concepts of Canadian and violinist, which is difficult because you might expect Canadians to be rugged and outdoorsy while violinists are expected to be more delicate. Failing the kind of variability

calculation suggested by R2, a natural solution is to reconcile the defaults in the direction of the one example of a Canadian violinist you have met, adding whichever value on the rugged/delicate dimension the person possesses. This process is different from bottom-up concept formation in its general form, since you are not generalizing all of your friends characteristics to be those of the typical Canadian violinist. The datum enters into the new combined concept only to the extent it enables you to reconcile conflicting defaults. The relevant rule is:

R4 If C_3 is being formed from C_1 and C_2 which conflict on some dimension, and you have examples of C_3 which have a value on that dimension, then choose for C_3 the value of the examples.

A looser variant of R4 is based on the notion of representativeness (Tversky and Kahneman 1974). Whereas R4 deals with the case where contextual examples have the properties which are needed to choose between the conflicting values in the donor concepts, R5 is designed to deal with cases where the combined concept is only similar to the examples. For example, in the feminist bank teller case, if Linda were taken to be an example of a feminist bank teller, then the default values of feminist would tend to win out over those of bank teller, since feminist is more representative of Linda than bank teller. The appropriate rule is:

R5 Choose for C_3 values taken from that concept, C_1 or C_2, which is more representative of the given instances of C_3.

4 Goal-directed Rules

A concept need not be completed all at once: default reconciliation may be an extended process. In some cases, none of R1-5 will be appropriate for reconciling conflicts between the expectations generated by donor concepts. The appropriate response then might be to wait and see which of the default values of the donor concepts will prove to be most suitable. Suitability here can mean just representation of the yet to be discovered properties of instances of the new concepts, but it can also mean usefulness in solving problems with which the new concept was intended to help. For example, the concept of a virus was formed from a kind of combination of concepts of macromolecule and living cell, and it was some time before biologists were able to reconcile conflicting properties of those entities. Induction and concept formation must be understood within the context of a scientist's general problem solving behavior.

This suggests the following rule:

R6 Reconcile slots in favor of ones which contribute to desired problem solutions.

The rules which result from R6 are likely to be tentative and subject to further testing, but can still play an important role in problem solving and explanation. Suppose, for example, that the situation which triggered the conceptual combination of feminist and

bank teller concerned the need to explain some feature of Linda's political behavior, where it was given that she is a feminist bank teller. Then adding the slot that feminist bank tellers are politically active provides an explanation of why Linda is politically active, since she is a feminist bank teller. Of course we already had the slot that feminists are politically active, but this alone may not be a good explanation of Linda's political activity since our knowledge that she is also a bank teller suggests the existence of a potentially relevant alternative reference class. Adding the slot about the expected political behavior to the combined concept of feminist bank teller resolves the problem. Similarly, suppose that in forming the combined concept of a Canadian violinist you notice that your friend the Canadian violinist prefers hamburgers to classical French cuisine. In order to explain this preference, you may add the default expectation about Canadians to your frame for Canadian violinist, overruling the expectation derived from the frame for violinists.[2]

We have seen how Minsky's frame notion can provide the basis for plausible mechanisms of conceptual combination. Prototype theories are not contradicted by phenomena of conceptual combination, and in fact increase our understanding of them.

NOTES

[1]Psychologists usually prefer the term "schema". For a discussion of the epistemology of such structures, see Thagard (forthcoming-FKI).

[2]Goal-directed conceptual combination is particularly important for scientific discovery (Thagard forthcoming-CCSD). New scientific concepts referring to non-observed entities such as light waves can be formed by combination of existing concepts.

REFERENCES

Fodor, J. (1981), Representations, Cambridge, Mass.: MIT Press.

Minsky, M. (1975), "A Framework for Representing Knowledge," in P.H. Winston (ed.), The Psychology of Computer Vision, New York: McGraw Hill, 211-277.

Nisbett, R., Krantz, D., Jepson, C., and Kunda, Z. (1983), "The Use of Statistical Heuristics in Everyday Inductive Reasoning," Psychological Review 90: 339-363.

Osherson, D. and Smith, E. (1981), "On the Adequacy of Prototype Theory as a Theory of Concepts," Cognition 9: 35-58.

Osherson, D. and Smith, E. (ms.), "Gradedness and Conceptual Combination," unpublished.

Smith, E. and Medin, D. (1981), Categories and Concepts, Cambridge, Mass.: Harvard University Press.

Smith, E. and Osherson, D. (1982), "Conceptual Combination and Fuzzy Set Theory," Proceedings of the Fourth Annual Conference of the Cognitive Science Society, Ann Arbor, MI, 47-49.

Thagard, P. (forthcoming-CCSD), "Conceptual Combination and Scientific Discovery," unpublished.

Thagard, P. (forthcoming-FKI), "Frames, Knowledge and Inference," Synthese.

Thagard, P. and Nisbett, R.E. (1982), "Variability and Confirmation," Philosophical Studies 42: 379-394.

Tversky, A. and Kahneman, D. (1974), "Judgment Under Uncertainty: Heuristics and Biases," Science 185: 1124-1131.

Tversky, A. and Kahneman, D. (1983), "Probability, Representativeness, and the Conjunction Fallacy," Psychological Review.

Winston, P. and Horn, B. (1981), LISP, Reading, Mass.: Addison-Wesley.

On Semantic Decomposition of Verbs

Karl F. Wender
and
Uwe Konerding
Technische Universität Braunschweig

Theories of semantic memory propose that the core meaning of a verb can be broken down into semantic components (e.g. Gentner, 1975; Miller, 1978; Miller & Johnson-Laird, 1976; Schank & Abelson, 1977). However , there exists some controversy about the psychological reality of these semantic components. The question is to what degree decomposition has to take place whenever a person uses or understands a verb.

Two differing points of view can be distinguished on the problem of decomposition. These views correspond to the processes of analysis and synthesis. Analysis is a process of abstraction which separates out parts of the meaning of one particular word. In related words similar parts are found. As an example, the notion of possession is a component of all HAVE verbs. The hypothesis of decomposition in the sense of an analytic process does not necessarily imply something about the format in which the meaning of a verb is stored. It merely says that it is possible to isolate components from the meaning of a verb that are also found in other words. It is like asserting that oak trees and apple trees have leaves which are similar although not identical. Research by Fillenbaum & Rapoport (1971), for example, documents that people can consistently carry out this process of analysis.

Synthesis, on the other hand, is a process for combining elementary components into word meanings. The system starts out with a set of fundamental components which are combined in different ways to create the different meanings of verbs. The same component is always identical even if used in different words. It is like placing identical parts into different cars, as e.g. batteries. This view of the decomposition hypothesis is found in psychological theories of semantic memory (e.g. Gentner, 1975; Grimm & Engelkamp, 1981; Sanford & Garrod, 1981). The view is even more prevalent in text processing systems of artificial intelligence (e.g. Schank & Abelson, 1977).

Several authors have argued against the hypothesis that the meaning of a verb is represented by a bundle of independent components, among them Fodor, Garrett, Walker, & Parkes (1980), Hörmann (1976), and Kintsch (1974, 1980). The present paper summarizes experiments that support the arguments of these authors.

In previous experiments we have already found some evidence against the synthetic version of decomposition theory (Wender, in press). In these experiments subjects had to decide whether the meaning of a given verb contained a specified component. Reaction times were measured. Pairs of components were used in which one component was included in the other. For example, INTENSION is embedded in INTENSIONAL ACTIVITY, when properly defined. We assume that search processes should, in principle, stay the same when different verbs are searched for these components. It was predicted that the difference in verification times between embedded and embedding components should remain constant across different verbs. Three successive experiments revealed significant interactions between components and verbs. That is, the difference in verification times depended on the particular verb being judged. Assuming that the search process in memory does not change its nature from verb to verb it was concluded that the structure of the embedding component cannot be the same in different verbs as claimed by synthetic decomposition theories. As an example, INTENSIONAL ACTIVITY is not the same in the verb "steal" as it is in the verb "buy". The present study investigates the same problem using a different

methodology.

Method

The method of paired comparisons was used to scale the relative importance of one component with respect to the total meaning of a verb. Six components were defined: GIVE, HAVE, TRAVEL, SPEED, INTENSION, INTENSIONAL ACTIVITY. Subjects were first given an explicit definition of one of the components. Then they were presented with two verbs and had to decide for which of the two the relative importance of the component was greater. Three sets of seven verbs each were used. Each set was judged with respect to two components. This resulted in six 7x7 paired comparison matrices.

Results

First the matrices were searched for violations of stochastic transitivity. Results did not accord with the predictions. Stochastic transitivity was never violated in the weak case, seldom in the moderate case, and rarely in the strong case.

In the next step we fitted a paired comparison model to the data. In this model the probability of choosing verb a over verb b when judging component c is given by

$$p(a>b/c) = 1 - (A-C) / ((A-C)+(A-B))$$

where A, B, and C are parameters corresponding to the "total amount of meaning" of a verb or a component, respectively. When applied separately to the six paired comparison matrices the model did a fairly good job. It had to be rejected in only one case on the 5% level of significance as measured by a Mosteller goodness-of-fit test. This result is taken as evidence that by and large the verbs may be represented on a one dimensional scale corresponding to the judged component. This is in accordance with decomposition theory.

Each set of verbs had been judged with respect to two different components. Within each pair of components there was one component embedded in the other. For example, HAVE is embedded in GIVE. Hence the two scales for each pair of components should be related in a simple fashion. To test this, the paired comparison model was applied simultaneously to the two corresponding paired comparison matrices. One additional parameter was included in the model which captured the difference in relative importance of the two judged components. The results of the Mosteller goodness-of-fit test were clear cut. The Chi-square values were 393.7 for the ACTIVITY verbs, 398.4 for the TRAVEL verbs, and 109.1 for the HAVE verbs with 29 degrees of freedom in each case. That is, the one dimensional model had to be rejected for all pairs of components.

Discussion

We found that the separate paired comparison matrices were scalable in accordance with decomposition theory but that the combined matrices were not. This is regarded as evidence against the theory. If one of two components is embedded within the other then the difference in relative importance should remain constant across different verbs in which both components are included. This was not the case. We suggest that this occured because the components do not have the structure that is assumed by synthetic decomposition theory. This is based, of course, on the assumption that judgemental processes do not change from verb to verb.

In conclusion we subscribe to the view of Bierwisch (1981) who distinguishes between a semantic and a conceptual structure in memory. The decomposition into semantic components is perhaps an essential part of the semantic structure which contains the rules of language. In the conceptual structure the meanings of words are represented in a Gestalt like manner. This notion resembles mental models as discussed by Johnson-Laird (1980). It must be admitted, of course, that this notion of a conceptual structure

has yet not been worked out to any satisfactory degree. The data of our experiments suggest that the decompositional theory, at least in the synthetical version, is not rich enough for representing the meaning of verbs.

References

Bierwisch, M. Basic issues in the development of word meaning. In W. Deutsch (Ed.), *The child's construction of language*. New York: Academic Press, 1981.

Fillenbaum, S., & Rapoport, A. *Structures in the subjective lexicon*. New York: Academic Press, 1971.

Fodor, J.A., Garrett, M.F., Walker, E.C., & Parkes, C.H. Against definitions. *Cognition*, 1980, *8*, 263-367.

Gentner, D. Evidence for the psychological reality of semantic components: The verbs of possession. In D.A. Norman, & D.E. Rumelhart (Eds.), *Explorations in cognition*. San Francisco: Freeman, 1979.

Grimm, H., & Engelkamp, J. *Sprachpsychologie*. Berlin: Erich Schmidt, 1981.

Hörmann, H. *Meinen und Verstehen*. Frankfuhrt: Suhrkamp, 1976.

Johnson-Laird, P.N. Mental models in cognitive science. *Cognitive Science*, 1980, *4*, 71-115.

Kintsch, W. *The representation of meaning in memory*. Hillsdale, N.J.: Lawrence Erlbaum, 1974.

Kintsch, W. Semantic memory: A tutorial. In R.S. Nickerson (Ed.), *Attention and performance* (Vol. VIII). Hillsdale, N.J.: Lawrence Erlbaum, 1980.

Miller, G.A. Semantic relations among words. In M. Halle, J. Breshan, & G.A. Miller (Eds.), *Linguistic theory and psychological reality*. Cambridge, Mass.: MIT-Press, 1978.

Miller, G.A., & Johnson-Laird, P.N. *Language and perception*. Cambridge, Mass.: The Belknap Press, 1976.

Sanford, A.J., & Garrod, S.C. *Understanding written language*. Cichester: John Wiley & Sons, 1981.

Schank, R.C., & Abelson, R. *Scripts, plans, goals, and understanding*. Hillsdale, N.J.: Lawrence Erlbaum, 1977.

Wender, K.F. Zur Komponententheorie des semantischen Gedächtnisses für Verben. *Sprache und Kognition*, 1984, in press.

Modeling Expertise in Troubleshooting and Reasoning About Simple Electric Circuits

by Barbara Y. White and John R. Frederiksen

Working within the framework of designing a computer-based system for teaching automotive electrical troubleshooting, we are developing a model of expert troubleshooting and a qualitative causal model of circuit behavior. The purpose of these models is to demonstrate to students the troubleshooting process and to explain the operation of circuits in faulted and unfaulted conditions. Our instructional interest is in determining how models of circuit behavior influence the learning of troubleshooting and how training in troubleshooting influences learning to reason about circuits. In this paper we will focus on the psychological criteria for constructing models of troubleshooting and reasoning about circuits.

In modeling the troubleshooting process, we interviewed and observed expert mechanics, studied automotive manuals, and reviewed computer based troubleshooting systems (Forbus & Stevens, 1981; Rouse & Ruston, 1982; Sleeman & Brown, 1982). We have observed three broad categories of troubleshooting behavior: (1) symptom fault associations (Rasmussen & Jensen, 1974), (2) decision trees, and (3) knowledge based inferencing strategies. Training based upon symptom fault associations requires higher fidelity than is possible using computer simulations and requires many years to gain experience with low probability faults in the real world. Decision trees have the drawback of being difficult to remember, incomplete, and do not develop skills that would enable the learner to troubleshoot systems other than the one explicitly trained. Our goal, therefore, was to find a knowledge based inferencing strategy that would be flexible, transferable, and not too difficult to learn.

We have worked with an expert troubleshooter who utilizes and teaches a knowledge based strategy (the Feed-Device-Ground or FDG strategy) to students in a technical high school. The strategy

appears to have general applicability and has several properties that make it easy to execute.

Firstly, the FDG strategy minimizes the number of entities about which one has to reason at any moment by focusing on one device at a time and dividing the circuit into three parts: feed, device, and ground (see Figure 1). The feed, for example, consists of all circuit elements between the positive voltage source and the device of focus. The ground is analogously defined. All inferences are made with reference to these three entities.

Secondly, the strategy seeks to eliminate ambiguities about the location of the fault before shifting the focus to another device. For instance, suppose that a test light (or voltmeter) is connected between the device and the negative terminal of the battery and that it does not light (or indicate a voltage). There are multiple faults that are consistent with this result: There could be an open or a short to ground in the feed to the device, the device itself could be faulty, or the ground circuit could be shorted. Given the ambiguity of such a test result, our expert has several techniques for determining whether the fault is in the feed, device, or ground system. One technique is to detach a possibly faulty ground system from the device. If the ground were shorted this detachment would cause the light to come on. Another technique is to provide a substitute feed to the device. If the light were then to come on, one could infer that there is a fault in the feed system. A further technique is to detach the feed system, while leaving the substitute feed in place. Then if the light were to come on, one could infer that the feed is shorted. If all of these techniques have been employed and the light still does not come on, one could then conclude that the device itself has an open or a short to ground. Using these methods, the fault can be isolated to be within the feed, device, or ground portion of the circuit.

Thirdly, the FDG strategy minimizes the memory demands for keeping track of what parts of the circuit are known to be good. Once the section of the circuit with the fault has been determined, the strategy involves serially searching that section of the circuit known to contain a fault (either the feed or ground) moving in a direction away from the device. This is done by repeatedly shifting the focus to the next device in the feed (or ground) system and utilizing the techniques for resolving

ambiguity outlined above to constrain further the location of the fault. It should be noted that a serial search is not the most efficient procedure for large circuits; however, it does make it easy to remember what parts of the circuit have already been tested and found to be free of faults. This makes the FDG strategy easier to execute than strategies which utilize potentially more efficient search procedures, such as repeatedly shifting the device focus to the middle of the suspect part of the circuit.

The FDG strategy has the advantage of being generally applicable to simple electrical circuits and minimizes memory requirements for executing it. However, it does presuppose knowledge of electrical circuits and appears to require an ability to reason qualitatively about circuits. This may make it a difficult troubleshooting technique for a novice to master. For example, a student needs to understand that for current to flow through a device, there must be a continuous path from a voltage source to the device and back to the opposite terminal of the voltage source. The implication of this principle is that in a series circuit an open will prevent current from flowing. However, in a parallel circuit, an open will not necessarily prevent current from flowing through the device if there is an alternative path for the current to take. Furthermore, students need to understand that in the case of parallel circuits, more current will flow in the path of lower resistance and that if there is one path with no resistance, all of the current will follow that path. The important implication of this principle for troubleshooting is that shorts to ground provide alternative paths with negligible resistance and thereby prevent current from flowing through the remainder of the circuit.

Given the need to teach these electrical principles and their implications as a prerequisite to teaching the FDG strategy for troubleshooting, we are creating an instructional environment that is capable of demonstrating and providing practice in using these principles. The basis of this system is a qualitative causal model that simulates the operation of an electrical circuit in both unfaulted and faulted states. The qualitative causal model incorporates knowledge of the structure of the circuit, the functioning of the devices within the circuit, and the electrical principles presented above.

There are a number of instructional requirements that constrain

the form of this qualitative causal model. Firstly, the model is to be capable of supporting graphical representations of circuit operation. These representations illustrate circuit topology, states of devices (e.g., an open or closed switch, a coil with a field around it), and current flow. They can show, for simple circuits, the effects of faults such as opens and shorts to ground on current flow and on test light behavior. Secondly, the model is to provide a simulation environment within which the FDG "expert" program can demonstrate troubleshooting concepts and procedures and the student can practice execution of the strategy. Faults can be introduced without the student's awareness, and the student has facilities for inserting a test light, setting the positions of switches and points, establishing a device of focus, installing substitute feeds, and detaching feed and ground circuits. The system will faithfully reproduce the effects of these manipulations on the behavior of the test light and the operation of the circuit. Thirdly, the model is to be capable of generating explanations of circuit operation. Moreover, these explanations employ the same qualitative reasoning principles used in the execution of the FDG strategy. When a component is set to be faulty, the system describes the effects of the fault on the operation of other components, on the behavior of a test light inserted into the circuit, and on the functioning of the ignition circuit as a whole.

In order to meet these instructional requirements, the model consists of (1) a representation of circuit topology, (2) a functional model for each device within the circuit, (3) rules for evaluating device states at each point in time, and (4) procedures for tracing the circuit to aid in evaluating conditions for device states. The topological representation describes the connections between devices within the circuit. The functional model for a device specifies its operating states and the conditions for entering those states. For example, the coil has two states: field building and field dissipating. The condition for entering the field building state is that an electrical potential exists across the Primary-Plus and Primary-Minus terminals. The condition for the field dissipating state is that there is no such electrical potential. In order to determine if an electrical potential exists, the model must check for a continuous path from each terminal to a voltage source. To determine if the path is continuous, the model must check the states of each device to see if it provides a conductive path.

If a break in the circuits occurs (if a switch is open or a device is faulted open), alternative parallel paths are searched. If no continuous path is found, the model concludes that there is no operating voltage for the coil. On the other hand, if at least one continuous path is found for both the feed and ground systems, then the model checks for shorts (paths of negligible resistance to the opposite side of the voltage source). In this way, characteristics of the circuit are examined in order to evaluate the operating requirements for a device. An advantage of this model is that the effects of faults on the operation of the circuit are easily determined. A further advantage is that the circuit tracing procedures are similar to those employed by the FDG strategy. Thus, in a sense, the troubleshooting strategy and the qualitative causal model utilize a similar kind of reasoning.

The model runs in discrete steps in time. All devices evaluate their states in parallel and appropriate changes in the graphic display are made. If explanations are desired, each step in the reasoning process can be articulated. The model also supports student initiated tests. When a test light is positioned in the circuit, procedures analogous to those described above deduce the state of the light. Running the model also enables one to determine the set of faults consistent with a given test result.

This model is similar to that of de Kleer and Brown (1983) in that it is based on qualitative causal reasoning. We selected this class of model (see also Kuipers, 1982) because it enables the instructional system to generate causal explanations that may help students to understand circuit behavior. Since our focus is on troubleshooting rather than "envisioning", our model differs from that of de Kleer and Brown in several respects. We sought a model that would be robust in permitting faults to be introduced without requiring a new model for each perturbation in the circuit. By utilizing context free functional models for devices along with a topological search process for evaluating device conditions, we were able to construct a single model that accounts for the effects of faults on the operation of the circuit.

Instructional considerations led us to decompose this domain into a troubleshooting component (the FDG strategy) and a circuit reasoning component (the qualitative causal model). This decomposition allows us to consider the effects of teaching

reasoning about circuits on the learning of troubleshooting skills and the effects of teaching troubleshooting on learning to reason about circuits. This latter case is interesting from the standpoint of helping students understand basic circuit theory since the troubleshooting task is a form of qualitative problem solving that can motivate the learning of circuit principles.

References

de Kleer, J., & Brown, J.S. (1983). Assumptions and ambiguities in mechanistic mental models. In D. Gentner & A. S. Stevens (Eds.), Mental Models. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Forbus, K., & Stevens, A. S. (1981). Using qualitative simulation to generate explanations. BBN Report No. 4490.

Kuipers, B. (1982). Commonsense reasoning about causality: Deriving behavior from structure (Working Paper). Medford, MA: Tufts University, Department of Mathematics.

Rasmussen, J., & Jensen, A. (1974). Mental procedures in real-life tasks: A case study of electronic trouble shooting. Ergonomics, 17, 3, 293-307.

Rouse, W. B., & Ruston, M. H. (1982). Human problem solving in fault diagnosis tasks. IEEE Trans. on Systems, Man and Cybernetics, SMC-8.

Sleeman, D., & Brown, J. S. (1982). Intelligent Tutoring Systems. London: Academic Press.

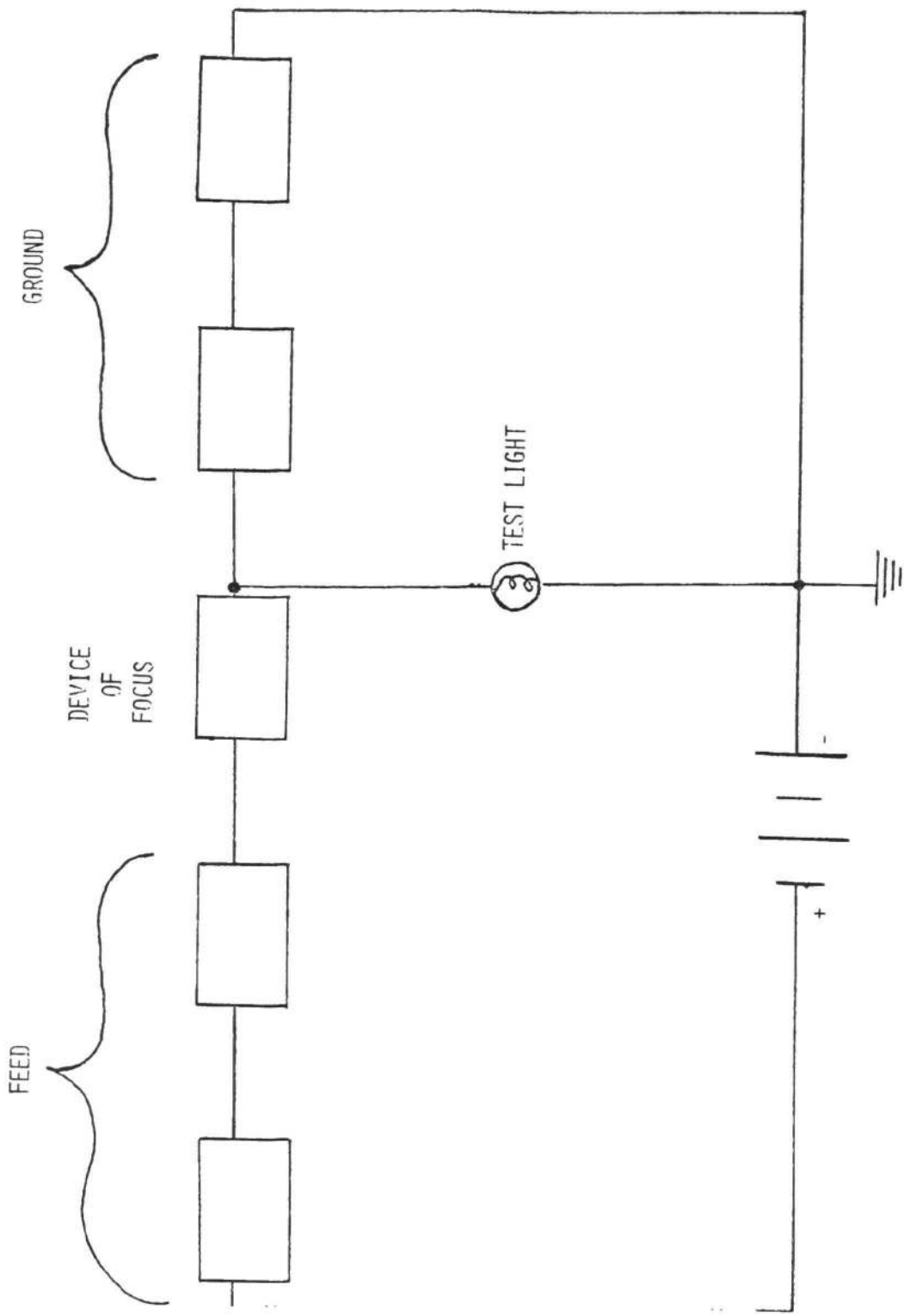


Figure 1.

Division of a circuit into feed, device, and ground circuit elements.

KODIAK - A Knowledge Representation Language

Robert Wilensky

Department of Electrical Engineering
and Computer Science
Computer Science Division
University of California, Berkeley

1. Introduction

A new theory of representation is proposed. The theory attempts to encompass representational ideas that have emerged from different schools of thought, in particular from work in semantic networks, frames, frame semantics, and Conceptual Dependency. The most important characteristic of the theory is the elimination of the frame/slot distinction made in frame-based languages (alternatively, case/slot distinction found in semantic network-based systems). In its place is a new notion called the "Absolute/Aspectual" distinction.

The theory described here provides a means of representation that has the following characteristics: It is broad and uniform, applying to any number of semantic domains; it is object-oriented; it contains a finite set of primitive epistimological relationships; it has the ability to create new relationships; it is cognitively plausible (i. e., it may reflect how things are represented in the mind); it conforms to other desirata for representations, such as canonical form and usefulness as a memory organizer.

2. The Problem with Frames

As has been pointed out by Woods (1975) and Brachman (1979), the interpretation of most semantic network formalisms is at best non-uniform. Attempts to address these inadequacies has led to the development of systems such as KL-ONE (Brachman et al. 1979). The theory proposed here similarly begins with a dissatisfaction with a number of existing formalisms. It ends up with a new formalism that is not unlike KL-ONE and its descendents in spirit. In detail, the formalism described below makes some different distinctions, and in some cases directly opposes the particular decisions made in KL-ONE and other recent attempts at knowledge representation.

We begin with frame-based systems (Minsky 1975) rather than semantic networks as the starting point. Research on frame-based systems have produced a number of interesting products, arguably, Conceptual Dependency* (Schank 1975) and scripts (Schank and Abelson 1977), which were specific to particular types of knowledge, and KRL (Bobrow and Winograd 1977), FRL (Roberts and Goldstein 1977) and FRAIL (Charniak et al. 1983), which were intended as general frameworks for representation.

In all the general frame languages, it is possible to define frames, and include in the definitions

*Conceptual Dependency preceeded frames historically, but was based on "case frames" that were frame-like in the Minsky sense.

assertions about what slots the frame has. It is also possible to write down arbitrary constraints on what may fill these slots, and to specify defaults for subclasses or instances of that frame. For example, one can define a "Person" frame, and specify that it takes slots for "Age", "Name", and "Address". In effect, such a frame system would be quite similar to a semantic network with a node for Person from which emanated links for Age, Name and Address.

Problem 1: The Meaning of a Slot is Completely Unconstrained

Despite the apparent usefulness of frames, what it means to be a slot in a frame is just as ill-defined as what it means to be a link in a network*. In particular, the meaning of a slot appears only procedurally, if at all. For example, if we fill the Address slot for some Person with "393 Foxon Road", this presumably means that that person's place of residence is at the location so designated. However, filling in the Name slot with "John Smith" means that the person is called by this name. Unfortunately, this difference in meaning appears only in the way various routines happen to manipulate those slots, i. e., it is encoded procedurally, and therefore, outside of the formal system of representation.

Problem 2: What May be a Slot in a Frame is Completely Unconstrained

There also appears to be no "in principle" answer to the question of which frames can support which slots. For example, if we allow Age to be a slot in Person, and Father (to be filled by the Person's father) to be a slot in Person, could we allow Father's-Age to be a slot? How about Person's best friends between the ages of 25 and 35? Regardless of our own intuitions, the frame languages do not distinguish the suitability one from another.

In actual practice, frame systems users appear to represent such knowledge outside the frame system. For example, "best friends between the ages of 25 and 35" might be represented as a conjunction in a predicate calculus-like formalism. The problem with this is that now there are two systems of representation. We have no way of decide what would be represented in which, or what it would mean to represent it one way rather than the other.

Problem 3: Many Concepts Do Not Get Defined

Most importantly, what we have been calling "slots" seem to be perfectly good structured concepts in their own right. These concepts are not only undefined - they tend to be completely unrecognized in frame systems. For example, the concept of Age has a perfectly well-defined meaning (in fact, more so that does Person). Namely, the Age "slot" implicitly refers to a concept which is the amount of time since the creation of an object to some other moment in time. Similarly, Address is a "referring object" for a building; Name is a "referring object" for a person.

In sum, frame systems tend to divide up the world into frames and slots, the latter not having true concept status. But the latter do appear to be full-fledged concepts. Frame systems neither recognize this fact nor allow for the expression of the meaning of these items.

3. KODIAK

KODIAK (Keystone to Overall Design for Integration and Application of Knowledge) is a knowledge representation language being created at the Berkeley Artificial Intellgience Research

*Charniak, Riesbeck and McDermott (1980) talk about these languages as "form languages" This nomenclature suggests, I think correctly, that the formalism is more of a form to collect knowledge than a representation of that knowledge.

Project that attempts to redress the above grievences. We view KODIAK as an extention of frames. However, the system is actually no more frame-like than semantic-network-like (which also appears to be the case for the more advanced semantic-network derived languages like KL-ONE).

Like KL-ONE, the primary structure of KODIAK is the *Concept*. However, there is no notion of role, slot or case. Instead, the idea of have a slot or role is replaced by one of a set of primitive epistemological relations. This relation is called **MANIFEST**. A Concept is in a **MANIFEST** relation to another *Concept* when, intuitively, the first *Concept* "has" the second *Concept* as a property. For example, if we want to indicate that physical objects have ages, we could assert that the *Concept* **Physical-Object MANIFESTs** the **Age** *Concept**. Furthermore, once the **MANIFEST** relation has been asserted to exist between two *Concepts*, a new relation comes into existence. This relation lets us assert that particular kinds (or instances) of one *Concept* can **MANIFEST** particular kinds (or instances) of the other. If *Concept1* **MANIFESTs** *Concept2,* say, then we name this relationship *"Concept2-of-Concept1"*. We call such a relationship an **aspectual.** In contrast, we call all other *Concepts*, such as **Age** and **Physical-Object**, **absolutes.**

For example, if we assert that **Physical-Object MANIFESTs Age**, then the aspectual relation **Age-of-Physical-Object** comes into existence. We can use this relation to assert the age of some particular physical object, among other things.

The intuition behind the idea of aspectuals to capture the dual use of terms like "name" and "color". When we talk of the "name of" someone or the "color of" an object, the claim is, we are referring to color as an aspectual (more properly, we are referring to the **Color-of-Physical-Object** aspectual). When we say "red is a color", we are talking about both **Color** and **Red** as *Concept*ual categories. Similarly, **Age** is the *Concept* of age, but **Age-of-Physical-Object** is the "age" implicitly referred to in "John is twelve years old".

In effect, we have split the idea of slot into several parts. One is the idea that a "frame" can have a slot of a certain type (this is expressed by the **MANIFEST** assertion); another is the *Concept* that is the slot (this is represented as another, in principle independent, *Concept*); finally, there is the fact that particulars or subtypes of the "frame" and **MANIFESTed** *Concept* can be in a relation of this sort to one another (this is enabled by the semantics of **MANIFEST**, and expressed by a particular derivative aspectual relation assertion).

It is awkward to talk about the assertion of a relation between two *Concepts*. Therefore, I shall loosely refer to such an assertion as a *link*, and depict it graphically as such.

The advantage of this formulation is that we can provide explicit definitions for and assertions about *Concepts* such as **Age**. In a traditional frame based system, such *Concepts* could not be predicated about explicitly.

For example, we would like to assert that the *Concept* **Age** is the difference between the creation time of an object and some other time (usually **Now**). To do so, we need to introduce some additional epistemological relations.

---

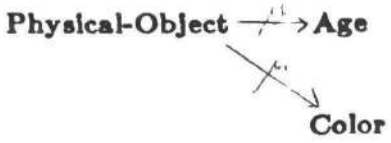
*Of course, we may want to assert this fact about some category more general than **Physical-Object**, so it would be meaningful to talk about the age of an idea, for example. In this paper, I shall not be terribly concerned about the correctness any such assertion. Instead, I will use categories that are familiar rather than those that may be technically necessary to describe properly a conception of the world.

4. Primitive Epistemological Relations

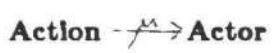
In KODIAK, the following set of epistemological relations is supposed:

MANIFEST

The semantics of **MANIFEST** is described above. We indicate this relation graphically by a directed arrow labelled "μ". Formally (i. e., in non-pictorial language) we indicate this by the form **(MANIFEST** *Concept Property-Concept*). For example, to indicate that a **Physical-Object** has an **Age** and a **Color**, we can draw the following:



Similar, we can indicate that an Action has an Actor:



These examples illustrate several different kinds of **MANIFEST**ation. Maida (1984) has suggested that *Concept*s like **Action MANIFEST Actor** *definitionally* (i. e., the *Concept* **Actor** is defined in terms of the *Concept* **Action**), whereas *Concept*s like **Physical-Object MANIFEST Color** *assertionally* (i. e., this asserts a true but non-definitional fact about the world). In addition, we suggest that **Physical-Object MANIFEST**s **Age** *derivatively* (i. e., the definition of **Age** entails this particular **MANIFEST** relation). See Maida (1984) for a further exploration of these ideas.

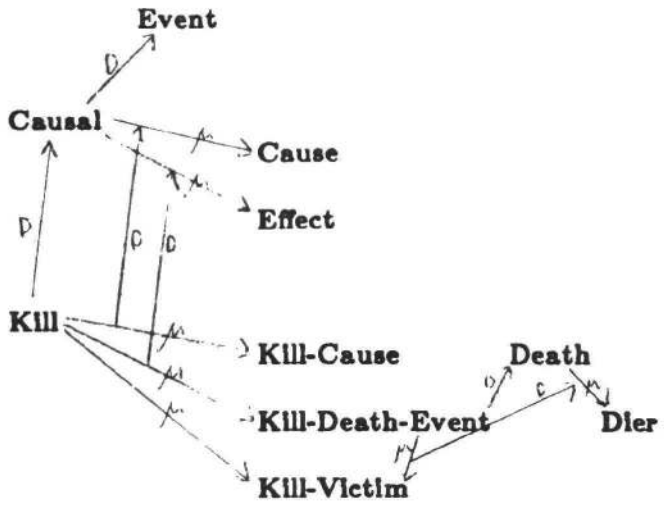
DOMINATE

This is a "structured inheritence" relation between *Concept*s. Its semantics is essentially ISA. We indicate it graphically by a link labelled "D" and formally by an expression of the form **(DOMINATE** *general-concept specific-concept*).

To indicate the relations between the parts of one *Concept* and those of a *Concept* that **DOMINATE**s it, we use an informal relation called "role-play". For technical reasons, this relation is implemented in terms of another, so it is not a true relation of the system. Nevertheless, it is convenient for expositional purposes.

As an example, we propose that there exists a type of **Event** called **Causal**, which **MANIFEST**s a **Cause** and an **Effect**. If we accept the interpretation that **Kill** means "cause to die", this can be represented by specifying a *Concept* **Kill** which **MANIFEST**s, among other things, a **Kill-Victim** and a **Kill-Death-Event**. The latter *Concept* is represented as meaning that the **Kill-Victim** died. We want to establish the meaning of **Kill** now by saying, intuitively, that **Kill-Death-Event** *plays* the *role* of the **Cause**, when **Kill** is viewed as a **Causal** event.

Rather than introduce an explicit role-play relation, however, we take advantage of the fact that the **MANIFEST** relations between **Causal** and **Effect** and between **Kill** and **Kill-Death-Event** give rise to aspectuals. In particular, they create the aspectuals **Effect-of-Causal** and **Kill-Death-event-of-Kill**. Since aspectuals are full-fledged *Concept*s in KODIAK, we can represent the role-play relation simply by asserting that the latter aspecutal is **DOMINATE**d by the former. Thus we have the following graphic depiction:



First, note that these terms refer to the actual *Concept*s. For example, the term **Cause** refers to the idea of "cause", and the term a **Effect** to the idea of "effect". These are not meaningless placeholders in a form.

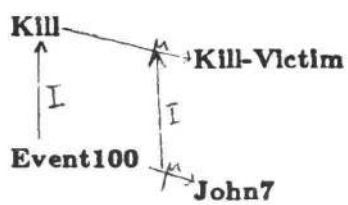
Second, much has been omitted in this diagram, for example, the semantics of **Cause**, **Effect** and **Death**. These are of course a crucial part of the overall system, and are omitted here for simplicity's sake.

Third, note that some *Concept*s, for example, **Kill-Cause**, have no additional semantics associated with them. That is, this is an "empty" *Concept*. **Kill** could have inherited the general **Cause** from **Causal**, so in this case the new name is not strictly necessary. However, it would become necessary if we wanted to make an assertion about the **Cause** of a **Kill** event. In contrast, the *Concept* **Kill-Death-Event** has an explicit definition as a kind of **Death** event.

INSTANTIATE

This relation holds when one *Concept* is to be considered as an instance of another. Its depiction is similar to that for **DOMINATE**s. For example, the fact that some *Concept* represents an individual human being would be represented by an **INSTANTIATE** link between that *Concept* and the *Concept* **Person**. Similarly, a particular killing event would be represented by an **INSTANTIATE** link between the particular event *Concept* and the *Concept* **Kill**.

Like **DOMINATE**, **INSTANTIATE** allows for "role-play" relations between the respective **MANIFEST**ed Concepts. For example, to represent the event in which John was killed, we create a new *Concept*. We call this Concept **Event100**, to suggest mneumonically that it is an event, and to indicate that such *Concept*s are rather numerous. Similarly, **John7** denotes the *Concept* of the person named "John." We then indicate that **Event100** **INSTANTIATE**s **Kill**, and that **John7** plays the role of the **Kill-Victim:**



Again, the representation shown here is abbreviated. For example, the link between **John7** and **Person** is not shown, nor is the information that the first name of **John7** is "John."

Note that in KODIAK, there is no such thing as an individual per se. Rather, the notion of an individual is meaningful only with respect to another concept. For example, all of the rather general category concepts mentioned above may be individuals of other categories. For example, all of them could be individuals of the *Concept* **Category**, should we introduce such a term in the system. The properties of some individuals that usually leads to typing objects "individual" or "generic," as in KL-ONE, are here considered to be peculiar properties of physical objects rather than something intrinsic to individuals.

As a further example, consider the **War and Peace** problem. The book **War and Peace** is an individual of the *Concept* **Book**. However, the particular copy of **War and Peace** sitting on my shelf is appears to be in the same relationship to the *Concept* **War and Peace** as that *Concept* is to the *Concept* **Book**. This situation can be represented in KODIAK by asserting that the *Concept* **War and Peace INSTANTIATE**s the *Concept* **Book**, and that the particular copy of a book **INSTANTIATE**s the *Concept* **War and Peace.**

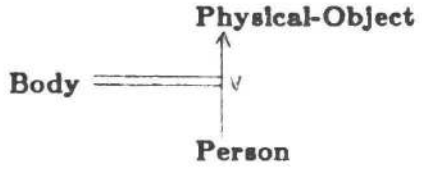
VIEW

An important aspect of the theory underlying KODIAK is that conceptual structure is not monolithic or static. In particular, we want to be able to talking about viewing one *Concept* in terms of another. This idea was first suggested as a representational technique in KRL (Bobrow and Winograd 1977). KRL does not admit to a notion of definition, and treats all perspectives as equally valid. We do not adopt this extreme position, but want to allow the flexibility of viewing a (possibly defined) *Concept* as something other than its "ordinary" interpretation.

For example, it is desirable to realize that a person can have properties, such as weight and color, that are generally considered to be general properties of all physical objects. In most representational schemes, to capitalize on this knowledge about physical objects, it is necessary to assert that persons are a kind of physical object. This is peculiar, because such a view of people is at odds with a normal working distinction between people and physical objects.

In KODIAK, we resolve this problem by introducing the relation **VIEW**. **VIEW** is similar to **DOMINATE**, except that it does not imply a primary or definitional status to the relation. For example, in KODIAK, we can assert that **Person** is **DOMINATE**d by **Living-Thing**, or some such *Concept*, and also assert that we can **VIEW Person** as a **Physical-Object.**

VIEW is more complicated than the other relations we have seen. This is the case because the **VIEW** of one object as another is itself a full-fledge *Concept*. For example, the **VIEW** of a **Person** as a **Physical-Object** is itself the *Concept* **Body**. Thus we represent **VIEW**s as three-part relations. We depict this graphically as follows:

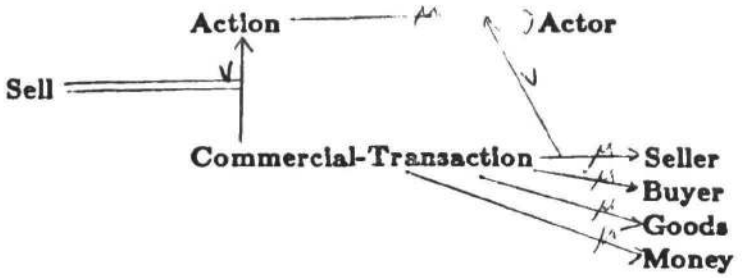


Formally, we can say that **(VIEW** *viewed-concepted viewed-as-concept view-concept*), meaning that *view-concept* is *viewed-concept* viewed as *viewed-as-concept*.

As is the case with **DOMINATE**, we can elaborate on a view by specifying additional **VIEW**s between the derived aspectuals of the *Concepts* participating in the relation.

One application of **VIEW** is to express some of the notions that arise in frame semantics (Fillmore and Kay 1980). In this system, some concepts are defined in reference to a background

frame. For example, "buying" and "selling" are defined in reference to the frame for "commercial transaction". We can represent this with **VIEW** as follows:



Buy is defined similarly.

GENERIC-INDIVIDUAL

This relation is used to define a *Concept* that acts as an exemplar of another *Concept*. Properties that are typically true of a *Concept* but not strictly necessary may be asserted about a *Concept* that is in a **GENERIC-INDIVIDUAL** relation to another *Concept*. Information about "prototypes" can be accommodated in this manner. **GENERIC-INDIVIDUAL** is similar to the ***TYPE** feature of Fahlman's NETL system (Fahlman 1979).

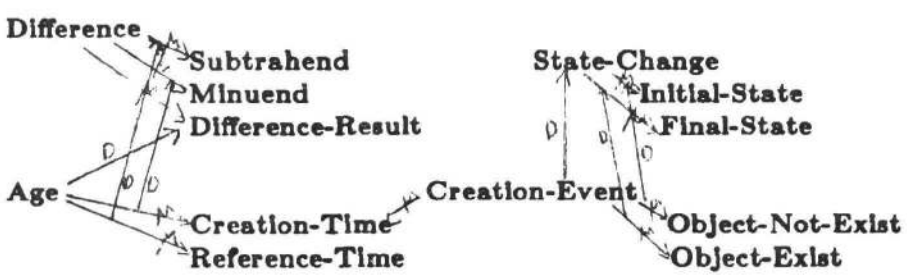
EQUATE

This relation is used to show that two descriptions are co-referential. We shall not elaborate on its use here.

5. Examples

Age

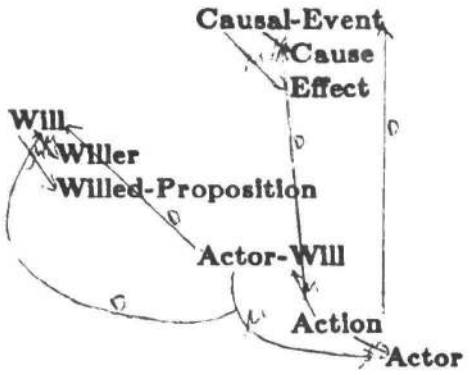
As mentioned about, a strong motivation for KODIAK was to be able to represent the semantics of concepts like "age". Given the above relations, we can define an **Age** *Concept* which is the difference between the creation of a thing and some other time:



In this representation, **Age** is represented as a **Difference-Result** of the **Difference** between **Creation-Time** and a reference point. **Creation-Time** is further defined, although the representation of **Object-Exist**, etc., is abbreviated.

Action

In KODIAK, an **Action** is just another type of **Causal-Event**. In particular, it is the class of such events where the **Cause** is the **Actor** willing some intended state. We can thus represent the general idea of **Action** as follows:



Here we neglect to represent that the *Concept* **Will** is a kind of **Mental-State.**

6. Processing and Representation

One advantage of this representation is that it allows for the full and deep meaning representation, but, at the same time, has the property that simple linguistic forms (i. e., one's that seem to be easily understood) can be easily represented. For example, to represent the fact "Bill was killed", we need only create a new symbol designating the particular event, and a new symbol designating the person and then grow the appropriate links. To represent "John killed Bill", we could add further links indicating that the symbol designating the new event is also an **Action**, with the symbol designating "John" being the **Actor**.

Now, if we wished to represent "John killed Bill intentionally", we would first have to have represented the *Concept* **Intended-Action**. This could be represented as a kind of **Action** in which the **Actor** Willing something is the actual **Cause** of that thing. Then the representation of the sentence just entails an additional link to this *Concept*.

The advantage here is that we capture the full semantics of these sentences, but do not require processing that seems out of line with the ease with which these sentences can be understood.

7. Conclusions

An outstanding feature of KODIAK is the proliferation of concepts. Rather than a small set of semantic notions from which all meaning is derived, there will end up being many more concepts in KODIAK than there are words of a given language. This does not appear to be problematic, because even more reductionistic systems seem to end up with such concepts. For example, the various knowledge structures of proposed by Schank seem to posit the existence of a large class of elements similar to those explicitly acknowledged in KODIAK. What we have attempted to provide is a uniform means to represent these notions, independent of their particular semantic concept.

Of course, there are many representational problems which the current system does not address. However, most of these appear to be problematic for all systems. We are hopeful that the framework established in KODIAK will be able to accommodate solutions to these problems without radical changes, although we have not had enough experience with the system to support such a claim.

8. References

(1) Bobrow D. G. and Winograd, T. An Overview of KRL, a Knowledge Representation Language. In **Cognitive Science** Vol. 1, No. 1, pp. 3-46. 1977.

(2) Brachman, R. J. On the Epistemological Status of Semantic Networks. In **Associative Networks: Representation and Use of Knowledge by Computers.** N. V. Findler (ed.). New York: Academic Press, 1979.

(3) Brachman, R. J. et al. Research in Natural Language Understanding. BBN report No. 4274, Cambridge, Mass. 1979

(4) Charniak, E., Gavin, M. K., and Hendler, J. A. The FRAIL/NASL reference manual. Technical reprot CS-83-06, Department of Computer Science, Brown University. 1983.

(5) Charniak, E., Riesbeck, C. K., McDermott, D. **Artificial Intelligence Programming Techniques.** Lawrence Erlbaum Associates: Hillsdale, New Jersey. 1980.

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

(7) Fillmore, C. and Kay, P. Progress Report: Text Semantic Analysis of reading Comprehension Tests. Manuscript. 1980.

(8) Maida, A. Conceptual Coherence (working paper) 1984.

(9) Minsky, Marvin. A framework for representing knowledge. In P. H. Winston, (ed.) **The Psychology of Computer Vision.** McGraw-Hill, New York. 1975.

(10) Roberts, R. B. and Goldstein, I. P. The FRL Manual. Technical Report AIM-408, MIT Artificial Intelligence Laboratory. 1977.

(11) Schank, R. C. **Conceptual Information Processing.** North Holland, Amsterdam. 1975.

(12) Schank, R. C. and Abelson, R. P. **Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures.** Lawrence Erlbaum Associates: Hillsdale, New Jersey. 1977.

(13) Woods, William A. What's in a Link: Foundations for Semantic Networks. In **Representation and Understanding: Studies in Cognitive Science.** D. G. Bobrow and A. COllins (eds.). New York: Academic Press, 1975.

On Self-Organization in Connectionist Networks

Ronald J. Williams
Institute for Cognitive Science
University of California, San Diego C-015
La Jolla, CA 92093

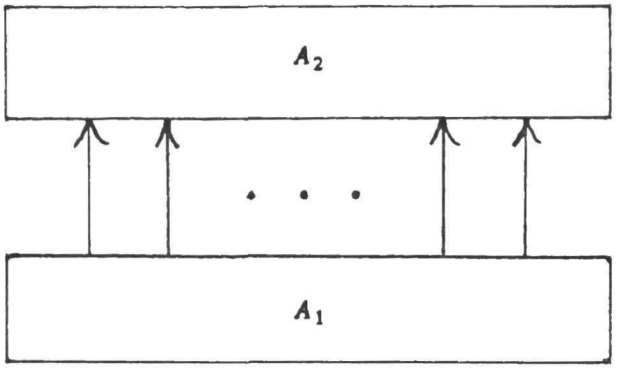
The aim of this paper is to present some observations about certain types of representations, or encodings, in connectionist, or neural-like, networks. In particular, this paper will call attention to two distinct categories of encoding in such networks and examine some results bearing on the issue of self-organizing networks which use one or the other type of encoding. This discussion will be limited to the encoding of data which is fundamentally numerical (or, more precisely, geometric). It is an interesting question whether semantic data can also be imbedded in a geometric framework, but such matters will be ignored here.

A number of interesting attempts have been made to provide an answer to the general problem of how a network might be shaped to a particular environment through self-organization. Among these are the early perceptron studies of Rosenblatt (1962), the investigations into possible neural net dynamics by Grossberg (1980), the recent theoretical approach of Hinton & Sejnowski (1983), and several works with the goal of finding ways in which cells in visual cortex might become tuned to specific features through self-organization (von der Malsburg, 1973; Nass & Cooper, 1975; Bienenstock et al., 1982). Two recent works which share a common perspective with the approach to be taken here are that of Kohonen (1982) and that of Amari (1983).

On the other hand, a number of non-self-organizing connectionist networks have been hand-crafted to perform particular sensory or cognitive processing tasks in ways which are generally intended to account for human performance data and/or be compatible with neurobiological data (Feldman & Ballard, 1982; Ballard, 1981; McClelland & Rumelhart, 1981; Hinton, 1981). Certain classes of network have even been proposed as having a certain universality in sensory processing, at least in the visual system (Ballard, 1981). Such universality might reasonably be taken as making such networks plausible candidates for the actual implementation of these algorithms in the brain. It then becomes reasonable as well to investigate possible mechanisms by which such networks might be able to self-organize to some degree; if such mechanisms can be shown to exist, it could then be argued that these types of network represent a general processing strategy which could find wide applicability in the brain.

This research was supported by a grant to David Zipser from the System Development Foundation.

In what follows, attention will be restricted to the following 2-layer architecture:



A_1 and A_2 are layers of units and each may optionally have fixed lateral connections. As depicted, there are connections from A_1 units to A_2 units, and these will be assumed to be variable and thus subject to self-organization. In addition, there may be connections (not depicted above) from A_2 units to A_1 units, and these may also be variable. The reason that only the $A_1 \rightarrow A_2$ connectivity is depicted above is that it is the $A_1 \rightarrow A_2$ transfer function that is of paramount interest here. Specifically, input to the system will be assumed to consist of a pattern of activation in the A_1 layer, and output will be taken as the resulting pattern of activation in the A_2 layer. While this network is being considered here in isolation, one may view this more generally as simply a sub-network consisting of two adjacent layers in a larger hierarchical network.

I will consider the $A_1 \rightarrow A_2$ transfer function as performing a mapping between two individual encodings, from that in the A_1 layer to that in the A_2 layer. A pattern in either layer can be considered as a Euclidean vector whose coordinates are simply the respective activation values of all of the units in that layer. The distinction I suggest drawing between encodings essentially revolves around how useful such a vector space description is for capturing the essential dimensions along which the lawful patterns may vary.

A full characterization of the two types of encoding will not be given here; it will be sufficient for present purposes to simply give examples of each and to cite a closely related distinction already existing in the literature.

Ballard (1981) calls attention to the distinction between having each unit in a network represent a particular point in a parameter space (with its activation representing confidence in the validity of that point) and having units whose activation represents the value of a (necessarily one-dimensional) parameter. The former is called a *value unit* encoding by Ballard and is used extensively in his generalized Hough transform approach to early visual processing; the latter is called a *variable unit* encoding.

For purposes of this paper, call any variable unit encoding a *Type I* encoding; the class of *Type II* encodings will include any value unit encoding as well as any representation typified by pixel-level descriptions of retinal images.

As a concrete example, consider a 1-dimensional array of 10 units such that the only patterns which appear in this array all consist of two adjacent 1's with the rest 0's. This is a Type II encoding of a pattern space which may be considered essentially one-dimensional; the 10-dimensional vectors

which represent the patterns jump around in the space in such a way that this one-dimensionality is not easily recognized. This one-dimensionality is really a consequence of the manner in which the patterns overlap.

In contrast, this same pattern space may be given a Type I encoding in a single unit whose activation is a monotonic function of, say, the distance of the leftmost 1 in the pattern from the left-hand end of the array.

At this point, the central thesis of this paper can be stated: *Self-organizing mappings from Type I representations is straightforward; self-organizing mappings from Type II representations, if possible at all, will require the use of mechanisms yet to be discovered.* In support of the first half of this thesis, I present the following two examples of self-organizing mappings, the first taken from work of Kohonen (1982) and the second from recent work of my own. Following these examples is a discussion in support of the second half of this thesis.

Example 1. Let the A_2 layer have a certain pattern of lateral feedback connections so that the only patterns of activity which it supports are such that all non-zero activity is confined to a very small number of nearby units. In particular, assume that the units are laid out in 2-dimensional space in such a way that nearby units excite one another but more distant units inhibit one another. Suppose that the A_1 layer consists of 2 units, with patterns drawn uniformly from a convex subset of Euclidean 2-space. Suppose also that there are no $A_2 \rightarrow A_1$ connections. Then Kohonen (1982) has shown that, by using a common variant of what has come to be known as the Hebb learning rule, the $A_1 \rightarrow A_2$ mapping will generally self-organize in such a way that nearby units respond most strongly to nearby patterns. [1] The resulting mapping re-codes the 2-dimensional pattern space implicit in the activations of the A_1 units in such a way that its 2-dimensionality becomes explicit in the A_2 layer. In the language of Ballard (1981), the resulting mapping can be said to turn a variable unit encoding into what is essentially a value unit encoding; in the terminology of this paper, the resulting mapping recasts a Type I representation into a particular Type II representation.

Example 2. Let the system have no lateral connections in either the A_1 layer or the A_2 layer, but let there be reciprocal $A_2 \rightarrow A_1$ connections. Let the A_1 units apply a weight modification rule to their incoming $A_2 \rightarrow A_1$ connections which has the effect of trying to more closely match their current pattern; furthermore, let them apply this same correction to their outgoing $A_1 \rightarrow A_2$ connections. Then, if the bottom-up and top-down connections are symmetrical, [2] the system performs a principal component analysis of the training stimuli during self-organization. More precisely, let n_2 denote the number of units in the A_2 layer. Then, if this system is trained with patterns having mean 0, self-organization causes the output corresponding to any given input vector to consist of a projection onto the subspace spanned by the eigenvectors corresponding to the n_2 largest eigenvalues of the scatter matrix of the training stimuli. This output is expressed in some orthonormal basis which need not be these eigenvectors themselves. In other words, individual units in A_2 will not necessarily be feature detectors for individual principal components; instead the output encoding may be distributed with respect to these components. A fuller account of the details of this system and an analysis of its behavior will appear elsewhere.

---

1. I have slightly simplified the actual details of Kohonen's work in order to avoid discussion of technical matters not germane to this presentation.

2. These weights need not be assumed symmetrical at the outset; the simple trick of allowing all weights to decay slowly will accomplish the necessary symmetry eventually.

The key point to be made about the system of Example 2 in the context of this paper is that it readily self-organizes a useful mapping from one Type I representation to another.

While the work of Kohonen (1982) and Amari (1983) may leave one with the impression that certain Type II → Type II mappings may be self-organized in the same way as described in Example 1, I would claim that, in general, a good mapping is not achieved through the application of such a learning rule. For example, suppose that the A_1 layer is an identical copy of the A_2 layer as described in Example 1 and the system is expected to self-organize what is essentially an identity mapping. This is a simple version of the problem of forming a topographic map between, for example, the retina and visual cortex. While Amari (1983) makes certain claims about such self-organization being possible, he readily concedes that it is difficult to obtain a topographic map from such a system if one starts with totally random initial connections. In fact, my simulations of such a system would suggest that unless one starts with initial connections very close to what one intends as the final outcome, the system is very unlikely to form a true topographic map. The major difficulty, it appears to me, is that the learning rule basically requires that, at statistical equilibrium, the stimulus vector to which each A_2 unit most strongly responds must be equal to a weighted average of the stimulus vectors to which its neighbors (in the topology of the lateral connectivity of A_2) most strongly respond. This is the underlying reason why such a system works well for self-organizing convex Type I input, and why I claim that it cannot be expected to do the same for Type II input. Indeed, this fundamental difference is the main motivation behind my drawing the distinction between these two types of representation. This argument in fact suggests that any learning rule which causes an A_2 unit to learn to respond to an average of the stimulus vectors which have excited it cannot be expected to achieve a good mapping between Type II encodings. Some other learning rule must be used to achieve this.

Thus I argue that it remains an open question whether mappings can, in general, be self-organized from a Type II representation to another Type II representation. Discovery of such a mechanism would be quite interesting, since it is possible to specify lateral connectivity patterns in the A_2 layer which could force any particular topology on the stimulus space. As an example, if the A_2 layer has the connectivity of a Möbius band and the A_1 layer is a patch of 2-dimensional retina upon which patterns of activity are elongated bars of various orientation and position, then application of this mechanism should lead to a mapping in which each A_2 unit is maximally responsive to a particular combination of orientation and position. One can imagine self-organizing just about any Hough-style transform in this manner.

Another intriguing possibility which is suggested by this work is that of self-organizing a mapping from a Type II representation to a Type I representation. As an example of such a mapping, consider a description of a connected pattern on a 2-dimensional retina in terms of Fourier descriptors for the boundary (Zahn & Roskies, 1972; Persoon & Fu, 1977) along with the coordinates of the center of mass, all encoded in variable units. What is appealing about this particular example is that it should be much more economical in both units and connections to compute a mapping between a retina-based description of an object and an object-based description of that object (Hinton, 1981) if these descriptions are encoded in variable units than if they are encoded in value units.

References

Amari, S. (1983). Field theory of self-organizing neural nets. *IEEE Transactions on Systems, Man, and Cybernetics, 13*, 741-748.

Ballard, D. H. (1981). Parameter networks: Towards a theory of low-level vision. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, 1068-1078.

Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience, 2*, 32-48.

Feldman, J. A., & Ballard, D. H. (1982). Connectionist models and their properties. *Cognitive Science, 6*, 205-254.

Grossberg, S. (1980). How does the brain build a cognitive code? *Psychological Review, 87*, 1-51.

Hinton, G. E. (1981). A parallel computation that assigns canonical object-based frames of reference. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, B.C., Canada, 683-685.

Hinton, G. E., & Sejnowski, T. J. (1983). Analyzing Cooperative Computation. *Proceedings of the Fifth Annual Conference of the Cognitive Science Society*, Rochester, NY, 683-685.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics, 43*, 49-69.

McClelland, J. L., & Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception, Part 1: An account of the basic findings. *Psychological Review, 88*, 375-407.

Nass, M. M., & Cooper, L. N. (1975). A theory for the development of feature detecting cells in the visual cortex. *Biological Cybernetics, 19*, 1-18.

Persoon, E., & Fu, K. (1977). Shape discrimination using Fourier descriptors. *IEEE Transactions on Systems, Man, and Cybernetics, 7*, 170-179.

Rosenblatt, F. (1961). *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*. Washington, DC: Spartan.

von der Malsburg, C. (1973). Self-organization of orientation sensitive cells in the striate cortex. *Kybernetik, 14*, 85-100.

Zahn, C., & Roskies, R. (1972). Fourier descriptors for plane closed curves. *IEEE Transactions on Computers, 21*, 269-281.

In Search of Selective Inhibitory Processes[1]

Penny L. Yee
University of Oregon

Abstract

These studies discuss two possible explanations for the selective effects observed in lexical ambiguity studies: one is selective inhibition and the other is attention. The two views make different predictions when a neutral target item is introduced between presentation of a homograph and a subsequent related target. The data show little signs of selective suppression, but they do suggest that attention may increase priming without producing selectivity.

Introduction

The term inhibition is often used to refer to the suppression of some concepts or nodes in memory to allow some conscious mental representation of another. One form of general inhibition is presented by Posner and Snyder (1975). General inhibition in their terms is a product of attention such that all things not currently attended to are less likely to come to consciousness. Other researchers have proposed the existence of selective inhibitory proceses within the memory system in which there is suppression of specific pathways in memory to allow conscious representation of a competing one (e.g. Marcel, 1980; and Neill, 1977). The resolution of lexical amibiguities is one instance in which this seems to occur. In the classic study by Schvaneveldt, Meyer and Becker (1976) a series of three letter strings were presented for word-nonword judgements, and in each of the experimental trials the second word of the series was always ambiguous with the first and third words in some way related to it. Schvaneveldt et al. demonstrated that lexical decisions on the third word were faster when the first and third words were related to common meanings of the ambiguous word (e.g. SAVE-BANK-MONEY), but not when the words were related to different meanings (e.g. RIVER-BANK-MONEY). This is typically what is found from studies presenting isolated single words as stimuli (see Marcel, 1980). When the ambiguous word is embedded in a sentence (e.g. Conrad, 1974 and Swinney 1979; Tanenhaus, Leiman and Seidenberg, 1979) the typical finding is for both meanings to exhibit evidence of activation followed by a rapid decline in the

priming effects for contextually inappropriate meanings. Thus, these data can be viewed as evidence for the selective inhibition view. Although selective inhibition explanations are appealing, it is possible that the observed selective effects are a result of attention focussing on certain parts of semantic memory

This paper presents a technique designed to determine if these selective suppression effects found in lexical ambiguity studies are due to selective inhibitory processes or if they are a result of focussed attention. The experiments described use a sequential lexical decision paradigm similar to the Schvaneveldt et al. study but include an additional factor referred to as the "separated" factor. If a related target appears immediately after the ambiguous word it is called unseparated if it appears one item later in the series it is called separated. To accommodate this additional factor all trials consisted of four items. Completely crossed with the separated factor were relatedness conditions. These were as follows: Congruent - the word preceding and following the homograph were related to common meanings; Incongruent - the word preceding and following the homograph were related to different meanings; Unbiased - the word preceding the homograph was unrelated, but the word following it was related.

Both views predict selective suppression effects in the unseparated cases, providing a replication of the Schvaneveldt et al. work. In the separated trials the semantic suppression view predicts continued suppression of the incongruent meanings, while the attention view predicts comparable facilitation for all conditions. Equivalent facilitation is predicted because the neutral item presented between the homograph and the subsequent related target word induces a shift of attention thereby dislodging its focus from one particular meaning. Hence, subsequent shifts of attention from this neutral point would give equal opportunity for either meaning of the ambiguous word to exhibit priming due to semantic activation.

Experiment 1

In this experiment subjects were presented a sequence of four items on each trial. In the experimental trials the second word presented always had a double meaning. In all trials the first two items were always words and they were only read by the subjects. The first item was presented for 500 msec. before the second word appeared directly below it. The two words remained on the screen for 1000 msec. followed by a blank period of 250 msec. At this point the first lexical decision target was presented which was the third item in the series. After a response there was another blank period of 250 msec. before the last lexical decision target was presented.

The means of subjects' median reaction times for the related target following the homograph are presented in Table 1. From these data there appears to be some priming in the Congruent condition but very little in the other experimental conditions. These differences are not reliable. A further analysis was performed, however, which suggested that some priming of related targets actually had occurred. This analysis involved splitting each subject's reaction times within a condition into two parts, on either side of the median, then averaging these scores together to give a fast and slow score for each subject in each condition. The rationale for performing such an analysis arose from the assumption that only a portion of the trials were affected by preceding context because they were, for some reason, more difficult to process than other trials. This idea reflects the findings by some researchers that context has a greater influence on slow readers (people with slower lexical access) (Perfetti, Goldman and Hogaboam, 1979; Stanovich and West, 1981). Consequently, one might expect more pronounced context effects in the slow trials than in the fast trials, and this is what was found (see Table 1). There are reliable differences between the Congruent and the control conditions in slow trials and virtually no differences between conditions in the fast trials. These data are not appropriate for examining the effects of the separated factor since the unseparated trials do not demonstrate the selective effects found by Schvaneveldt et al.

Experiment 2

In this experiment the target stimuli were the same as in the previous experiment, but the task was to perform a lexical decision on each item in the series. In performing a lexical decision it is assumed that subjects search for a meaning that can be associated with the letter string presented, and in this way attend to the semantic code for the word. Each item was presented in isolation with a 250 msec. interval between a response and the next target.

The means of subjects' median, fast and slow scores are presented in Table 2. Very strong nonselective priming effects are observable in all three breakdowns of the data. Thus, as predicted from the first two studies, having subjects attend to the semantic aspects of the priming words lead to much stronger effects. However, since I was unable to replicate the results of Schvaneveldt, et al., the intended comparison betwcen the selective inhibition view and the attentional view to assess the basis of selective suppression effects cannot be made on these data.

General Discussion

These studies present a method for identifying the processes that underly selective suppression effects often found in lexical ambiguity studies. The method is concerned with whether they arise from selective inhibition or from attention. To perform this test it is assumed that observable differences between the two views arise only after selective suppression is obtained. Although this seems to be a common finding for single word lexical decision studies these experiments were unsuccessful in reproducing the effects and so the test for selective inhibitory processes could not be conducted. This suggests that the finding of selective suppression with this paradigm may not be as common as initially thought.

Even though these experiments were not successful in fully explaining what produces selective suppression effects, other interesting points deserve mention. The first is the role of attention in obtaining larger priming effects. When subjects were forced to attend to the semantic aspects of the first two words (experiment 2) very large general priming effects were observed in all conditions. Notice, however, that priming in general increased for all conditions, but that no advantage for one meaning over the other was found. The order of the priming effects for each condition is compatible with the selective suppression predictions, but statistically the experimental conditions do not differ from each other. This result is troublesome for the attention view since according to it an increased focussing on the semantic code should lead to greater selective suppression effects. Instead, it only leads to increased priming overall, which is observable in all analyses (the median, fast, and slow scores).

When the data from the first experiment were analyzed globally negligible effects of condition were observed. Splitting each subject's data into fast and slow times produced a measure that was more sensitive to priming effects. The global statistics themselves were effective in picking up the effects in the last experiment in which attending to semantic aspects of the primes boosted the context effects. This suggests that the slow reaction time analysis is more sensitive in picking up weak effects in data, whereas the more global descriptive statistics will only pick up very strong effects and will mask weaker trends in the data.

REFERENCES

Conrad, C. Context effects in sentence comprehension: A study of the subjective lexicon. Memory and Cognition, 1974, 2, 130-138.

Henik, A., Friedrich, F. J., & Kellogg, W. A. The

dependence of semantic relatedness effects upon prime processing. Memory and Cognition, 1983, 11, 366-373.

Marcel, T. Conscious and preconscious recognition of polysemous words: Locating the selective effects of prior verbal context. In R. S. Nickerson (Ed.) Attention and Performance VIII. Hillsdale, N.J.: Lawrence Erlbaum Associates, 1980.

Neill, W. T. Inhibitory and facilitatory processes in selective attention. Journal of Experimental Psychology: Human Perception and Performance, 1977, 3, 444-450.

Perfetti, C. A., Goldman, S. R. & Hogaboam, T. W. Reading skill and the identification of words in discourse context. Memory and Cognition, 1979, 7, 273-282.

Posner, M. I. & Snyder, C. R. R. Attention and cognitive control. In R. L. Solso (Ed.). Informational Processing and Cognition: The Loyola Symposium. Hillsdale, N.J.: Lawreance Erlbaum Associates, 1975.

Schvaneveldt, R. W., Meyer, D. E. & Becker, C. A. Lexical Ambiguity, semantic context, and visual word recognition. Journal of Experimental Psychology: Human Perception and Performance, 1976, 2, 243-256.

Stanovich, K. E. & West, R. F. The effect of sentence context on ongoing word recognition: Tests of a two process theory. Journal of Experimental Psychology: Human Perception and Performance, 1981, 7, 658-672.

Swinney, D. A. Lexical access during sentence comprehension: (Re)consideration of context effects. Journal of Verbal Learning and Verbal Behavior, 1979, 18, 643-659.

Tanenhaus, M. K., Leiman, J. M. & Seidenberg, M. S. Evidence for multiple stages in the processing of ambiguous words in syntactic contexts. Journal of Verbal Learning and Verbal Behavior, 1979, 18, 427-440.

[1]This research project was supported in part by BRSG Grant S07RR07080 awarded by Biomedical Research Support Grant Program, Division of Research Resources, National Institute of Health and by ONR Contract No. N0014-83-K-1601.

TABLE 1

Median, fast and slow reaction times to the related targets following a homograph in each condition in experiment 1. Proportion of errors in each condition are shown in parentheses.

UNSEPARATED

	FAST	MED	SLOW
CONGRUENT (.03)	620	713	854
INCONGRUENT (.04)	648	746	948
UNBIASED (.03)	654	759	940
CONTROL (.03)	627	744	1006

SEPARATED

	FAST	MED	SLOW
CONGRUENT (.03)	580	669	834
INCONGRUENT (.02)	602	690	860
UNBIASED (.04)	599	689	848
CONTROL (.03)	580	687	961

TABLE 2

Median, fast and slow reaction times to the related targets following a homograph in each condition in experiment 2. Proportion of errors in each condition are shown in parentheses.

UNSEPARATED

	FAST	MED	SLOW
CONGRUENT (.03)	429	482	594
INCONGRUENT (.05)	457	520	638
UNBIASED (.06)	446	506	612
CONTROL (.13)	506	584	781

SEPARATED

	FAST	MED	SLOW
CONGRUENT (.11)	482	545	672
INCONGRUENT (.11)	499	567	709
UNBIASED (.12)	490	565	701
CONTROL (.19)	539	625	851

The Rôle of Internal Representations in the Acquisition of Motor Skills

Alf C. Zimmer
Department of Psychology
Westfälische Wilhelms-Universität
Fed. Rep. Germany

The typical situation in the learning of complex motor skills can be described as follows: An observer transforms perceived or described motor actions into his or her innervation patterns which initiate and control a motor behavior similar to or identical with the observed or described motor action. The internal or external evaluation of this similarity then serves as feedback for the learning process.

In the tradition of psychology a couple of different approaches to the analysis of this situation can be ditinguished:
(i) the behavioristic approach (Greenwald & Albert, 1968; and especially Skinner, 1968, where the learning of 'high jump' is analyzed) which concentrates on the situational variables and the related reactions.
(ii) the systems approach (Bernstein, 1967; Adams, 1971) in which the regulatory process of motor activity is central, that is, the modification of actions depending on the comparison of observed results with internal or external criteria.
(iii) the internal-representations approach (Bartlett, 1932; Schmidt, 1975), the central assumption in this approach is that perceptual as well as regulatory processes are governed by internal models (e.g. schemata).

An evaluation of these approaches can be oriented at Stelmach & Diggles' (1982) suggestion that theories of motor behavior should be able to explain the following phenomena:
(i) motor equivalence,
(ii) the variability of motor behavior, and at last
(iii) the complexity of the motor system.

The behavioristic approach fails for any of these criteria. Therefore, its seemingly elegant solution of the representation problem, namely by simply skipping it, does not work. Bernstein's (1967) original approach and Schmidt's approach (1975) succeed only partially. Bernstein's systems-theoretic approach fails because the assumption of a rigid motor program is not sufficient even for very simple positioning tasks. Schmidt (1975) who overcomes these difficulties by introducing seperate recall and recognition schemata fails for the criterion of complextiy of the motor system because he does not take into account that the very characteristics of a movement change if this movement is integrated into a movement of higher complexity.

The schema-theoretic approach as suggested by Cassirer (1944) can be used to integrate the systems approach and the internal representation approach.

The concept of the schema is defined according to Cassirer (1944) as consisting of:
(i) a set of primitives which are not further analyzable in the given context,
(ii) a set of organizational rules which can be paralleled to Helmholtz' logic of unconscious inferences,
(iii) a set of admissible transformations, that is, transformations which define the class of invariants of the objects (here: motor patterns) in question.

One important consequence of this definition is that the schema of a certain motor skill cannot be reduced to its primitive components and their relations, that is, (i) and (ii), but that the set of admissible transformations of this skill has be taken into account too. This is in line with the behavioral effects of the ablation of the motor cortex (Pribram, 1971), that is, a break-down of complex motor skills without an impairment of particular muscle functions. Pribram (1971, p.14) concludes: "... behavioral acts, not muscles or movements, were encoded in the motor cortex."
In an experiment how to learn cutting the spin in table tennis I have investigated the influence of different instructional methods on the internal representations of a biomechanically identical motor pattern. The two instructional methods were (group I) 'learning the underlying physical principle' and its consequences for the trajectories of a spinning ball, and (group II) 'learning by observing the correct motor pattern'.

In a first analysis it could be shown that group I changed from the state of non-competence to the state of competence without going through intermediate states. In contrast to this, subjects in group II exhibited the pervasive tendency to repeat rigidly the last reinforced movement pattern without taking into account the changed situational variables (e.g. speed of the ball etc.). However, in the end both groups learned the topspin, that is, they arrived at the same correct motor pattern. The state-transition diagrams in Figure 1 describe the differences in complexity of the learning process in the two experimental groups.

Insert Figure 1 about here

In the second part of the experiment the subjects had to learn the undercut. This task was chosen because from the point of view of mechanics the underlying invariant (the tangential impulse on the ball) remains the same for top-spin and for undercut. However, the required motor activities are completely different. Therefore it was expected that the internal representation of the task by means of a physical model would facilitate transfer. In contrast to this a purely motor or visukinesthetic schema (as it can be assumed for group II) should not be conducive to an immediate mastering of the new task.
In group I 6 out of 10 subjects were immediately able to perform the undercut (i.e. the transfer task) whereas only 1 subject out of 10 in group II was able to do it.
This result can be interpreted in the following way: The 'successful' subjects in group I had learned the schema 'spin' which is characterized by all transformations on actions which cause a rotation of the ball and thereby influence its trajectory. The subjects in group II had only acquired the schema 'top-spin' and had to learn the 'undercut' as a new schema. However, the times necessary for the acquisition of the new schema reveal that these subjects are able to utilize the preceding practice partially: their learning times are significantly shorter than the learning times for those subjects in group I who failed to identify the new task as a transformation of the schema 'spin'. This result indicates that there is one important negative consequence of the reduction of complexity by integrating motor schemata into an interdependent hierarchy, namely, that this integrated structure does not allow for an utilization of partial knowledge. An example for such an interdependent hierarchy is shown in Figure 2.

Insert Figure 2 about here

In this graph schemata are integrated upwards into a schema hierarchy which leads to a reduction of the complexity of the system. However, parallel to this kind of upward integration the higher-order schemata impose constraints upon the lower-order schemata. Such a hierarchy with upward integration and downward constraints is not decomposable in the sense of Simon (1965). The consequences of decomposable vs. non-decomposable representations of motor skills have been investigated by Körndle (1983) and by myself (in press). The underlying hypothesis of our experiment is that the described model of schema integration underlies the acquisition of skills. The practical consequence of this model is that complete transfer from one task to another is only possible if both tasks are admissible transformations of the same schema. Partial transfer (i.e. some but not all sub-skills necessary for one task are necessary for the other) is only possible as long as the sub-skills are not integrated into the superordinate schema.

This imputed mechanism has been investigated in an experiment where children were taught to ride a Pedalo - an instrument resembling partially a bicycle which is used to train the sense of equilibrium in children. The performance of the subjects has been measured by computing the difference between the velocity as prescribed by a metronome and the actual (observed) velocity. In Figure 3 this difference is given by the dotted area between the curve indicating the prescribed speed and the actual velocity of the Pedalo as produced by a subject.

Insert Figure 3 about here

A more detailed analysis of the motor action underlying this performance is possible by measuring the vertical forces (pressure), the horizontal forces (thrust) and the resulting forces. Typical examples for these data are shown in Figure 4 for a low degree of performance, a medium, and a high degree.

Insert Figure 4 about here

The comparison of the effective forces indicates that the acquisition of the skilled action is accompanied by an increasingly smooth flow of effective forces. This is achieved by integrating the actions controlling thrust and pressure into one action of higher order. In a transfer task (riding the Pedalo backwards) it was studied how the different levels of performance in the initial task influence the acquisition of the new skill. As predicted from the described mechanism of schema integration the transfer was best for subjects on an intermediate performance level. The reason for this can be seen in Figure 4 b): Subjects on an intermediate performance level are able to control thrust and pressure separately but not in the perfect coordination necessary for a smooth forward movement of the Pedalo. Since the coordination of thrust and pressure is different for the backward movement, the medium-level subjetcs are able to utilize 'pressure' and 'thrust' as sub-skills (i.e. lower-level schemata) in building up the new pattern of coordination, whereas the high-performance subjects have to start the learning process anew.
The subject's verbal reports on their coping with the task of riding the Pedalo are in line with the interpretation of the performance data. It turned out that on the intermediate stage the reports were highly detailed and consistent for the greater part of descriptions of perceptual and specific motor actions, whereas on the final stage subjects reported only very global strategies (e.g. "I try to thrust").

The result indicates the optimal timing for transfer is before the final stage of competence has been reached because on higher levels of competence the downward constraints impede the utilization of the sub-skills which are to be transferred from the initial task to the new task.

In conclusion, the results of the reported experiments support the suggested model for the internal representation of motor skills according to which the acquisition process is characterized by the progressive integration of lower-level schemata into schema hierarchies. The different levels of performance correspond to levels of integration: starting from a mere collection of lower-level schemata (sub-skills), a first level of integration is approached when independent sub-skills are roughly coordinated. On this stage the sub-skills are still available as building blocks (Rumelhart, 1980) for alternative forms of coordination. However, if on the final level of integration downward constraints restrict the admissible transformations of lower-level schemata, the schema hierarchy is no longer decomposable and therefore its constituents cannot easily be utilized for alternative skills.

References:

Adams , J.A.: A closed-loop theory of motor learning, Journal of Motor Behavior, 1971, 3, 111-149.

Bartlett, F.C.: Remembering, Cambridge: University Press 1932.

Bernstein, A.: The Coordination and Regulation of Movements, Oxford: Pergamon Press, 1967

Cassirer, E.: The concept of group and the theory of perception, Philosophy and Phenomenological Research, 1944, 5, 1-36.

Greenwald, A.G. & Albert, S.M.: Observational learning: A technique for elucidating S-R mediation processes, Journal of Experimental Psychology, 1968, 76, 273-278.

Körndle, H.: Zur kognitiven Steuerung des Bewegungslernens, Doctoral dissertation, Oldenburg: University of Oldenburg, 1983.

Pribaum, K.H.: What Makes Man Human?, New York: Museum of Natural History, 1971.

Rumelhardt, D.E.: Schemata - the building blocks of cognition. In: R.J. Spiro; B.C. Bruce & W.F. Brewer (eds.): Theoretical Issues in Reading comprehension: Perspectives from Cognitive Psychology, Linguistics, Artificial Intelligence, and Education, Hillsdale: Lawrence Erlbaum Ass, 1980.

Schmidt, R.A.: A schema theory of discrete motor skill learning, Psychological Review, 1975, 82, 225-260.

Simon, H.A.: The architecture of complexity, Proceedings of the American Philosophical Society, 1965, 106, 467-482.

Skinner, B.F.: The Technology of Teaching, New York: Appleton Century Crofts, 1968.

Stelmach, G.E. & Diggles, V.A.: Control theories in motor behavior, Acta Psychologica, 1982, 50, 83-105.

Zimmer, A.C.: Representações internas da habilidade dos movimentos nos esportes, Sprint, in press.

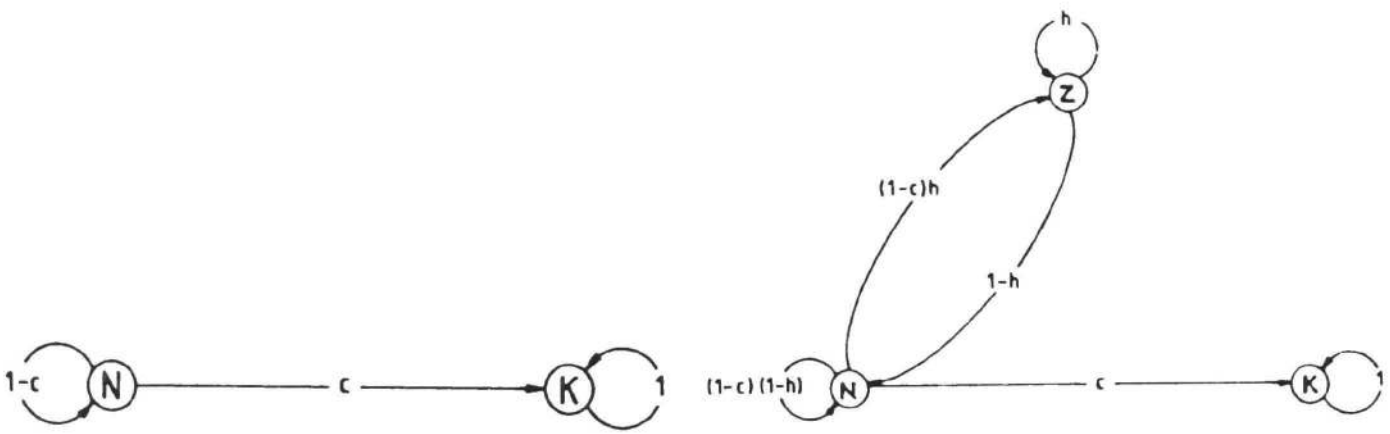
Figure Captions

Figure 1: State-transition diagrams for (a) group I and (b) group II. N indicates the initial state, K the final state and Z the representational state.
c,h,(1-c),1,(1-h),(1-c)h, and (1-c)(1-h) are the transition probabilities.

Figure 2: The model of schema integration (——→) and downward constraints (- - -).

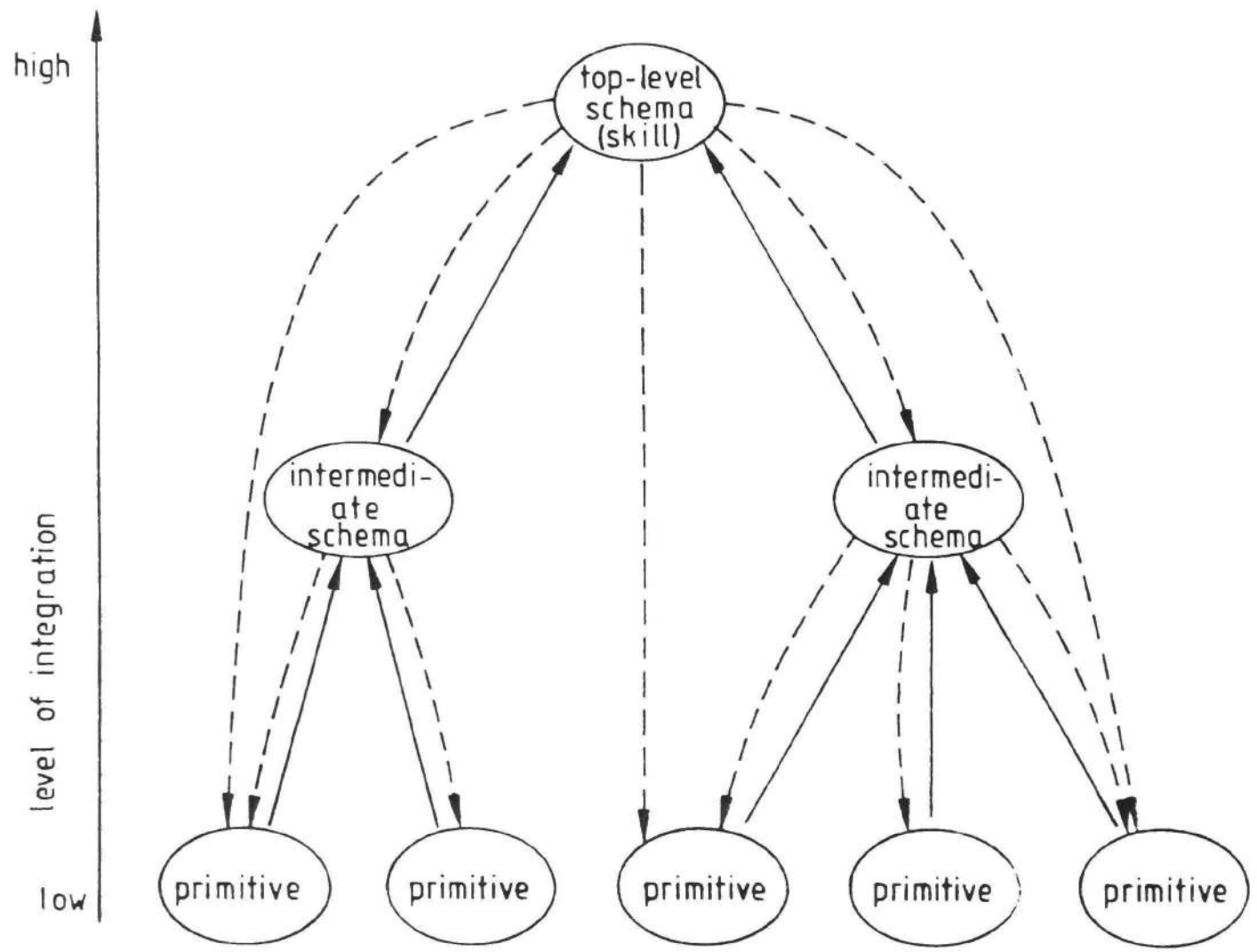
Figure 3: The measurement of performance in riding the Pedalo. The difference between the prescribed velocity (——) and the observed velocity (·····) is dotted. The dotted area is the performance measure.

Figure 4: (a) vertical, horizontal, and resulting forces in riding the Pedalo on a low performance level, for two full turns of the wheels
(b) the same for an intermediate performance level,
(c) the same for a high performance level. The length of the arrows inidicates the amount of force, the angular orientation gives the direction.



(a) (b)

Figure


high
level of integration
low
top-level schema (skill)
intermedi-ate schema
intermedi-ate schema
primitive
primitive
primitive
primitive
primitive

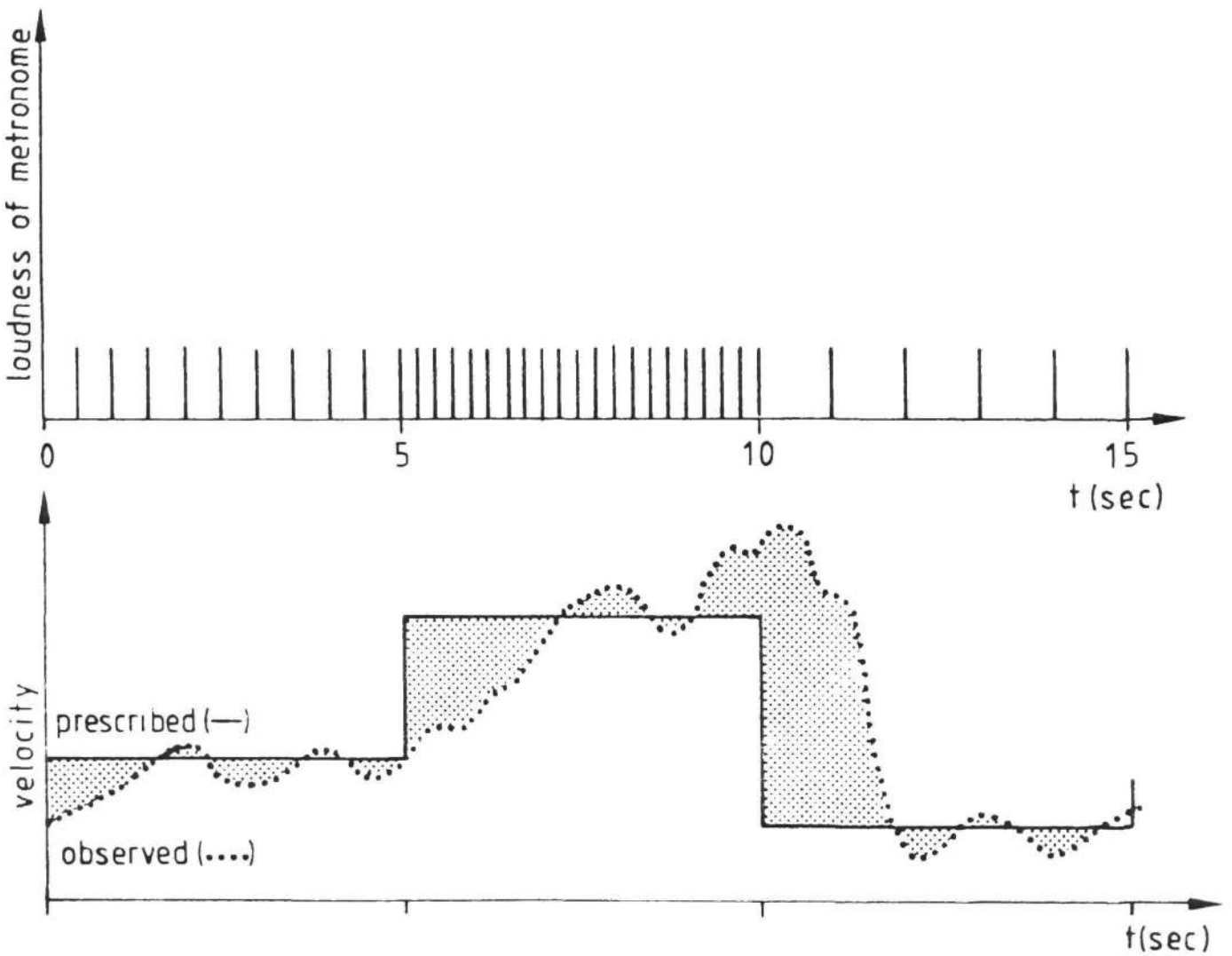

loudness of metronome
0
5
10
15
t (sec)
velocity
prescribed (—)
observed (....)
t(sec)

Figure 3

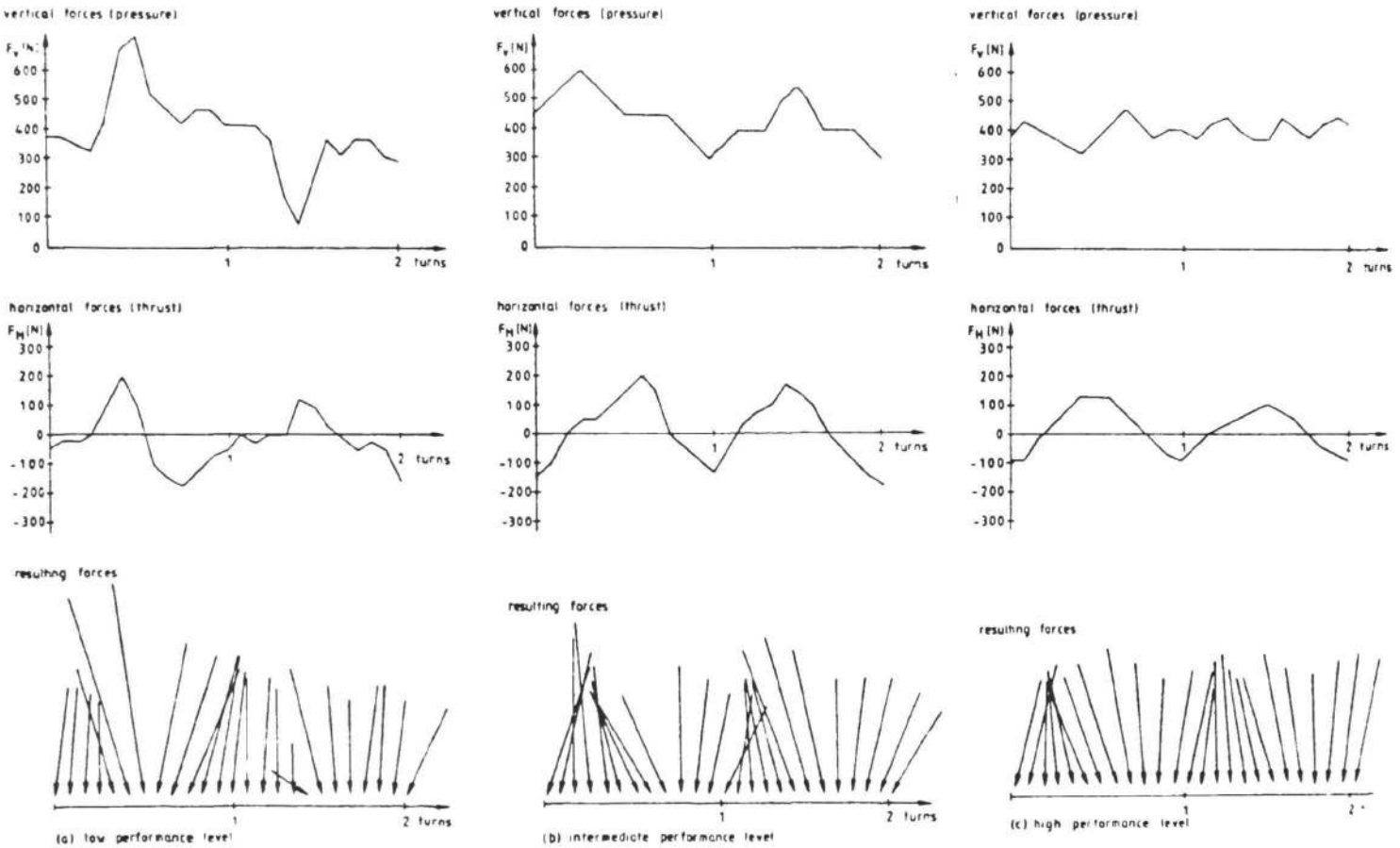

vertical forces (pressure)
F_V [N]
600
500
400
300
200
100
0
1
2 turns
horizontal forces (thrust)
F_H [N]
300
200
100
0
-100
-200
-300
resulting forces
(a) low performance level
(b) intermediate performance level
(c) high performance level

Figure 4

REPRESENTING COGNITIVE MAPS IN PARALLEL NETWORKS

David Zipser
Institute for Cognitive Science
University of California, San Diego

Maps are an important part of systems which represent world knowledge. They are used to provide information for guiding movement through the environment or for answering queries about the location of objects. The number of facts about the spatial relationships between objects, which maps must provide, is too great to be stored explicitly. What is needed is a representation that stores basic information in a manner that allows the use of inference to derive the appropriate facts when required. For example, in order to answer such queries as "am I headed towards home?", "how do I get from here to Carnegie Hall?", McDermott (1980) developed a spatial representation called fuzzy maps which has proven very useful in the symbol processing, serial computer environment. In this paper I will describe a different way to represent spatial knowledge called view-maps. The main motivation for developing view-maps was biological plausibility. I wanted a representation which was amenable to implementation in parallel networks of neuron-like units and whose properties corresponded to what we know about the neurobiology and psychology of spatial location. For a variety of reasons fuzzy maps do not suit this purpose.

To get a feel for what view-maps do, consider the problem of getting from your current location to an unseen goal, such as home. Assume that while you cannot see your home from your current location, you can see some familiar landmarks. Now suppose that at a previous time you had recorded in memory the location of your home relative to these landmarks. You can use this remembered information to locate your home and view-maps provide a mechanism to accomplish this. Roughly speaking, view-maps store a set of individual views, (Kuipers 1983) together with the locations relative to these views, of goals such as home, work, etc. When you want to go to one of these goals, you compare your current view with the set of stored views in the view-map. There will, in general, be no exact match, because you won't be at a precise point where you recorded a view. However, there will be similarity between your current view and views stored at nearby locations. These nearby views will differ from your current view only quantitatively in the values which indicate landmark location. It is possible to use these stored values of landmark location together with the values obtained from your current location to derive a transformation that will map the location of any object from the reference frame of a stored view to your current reference frame. By applying this transformation to the stored locations of a goal, you can find it in your current frame. View-maps use this general principle to guide an observer through the world.

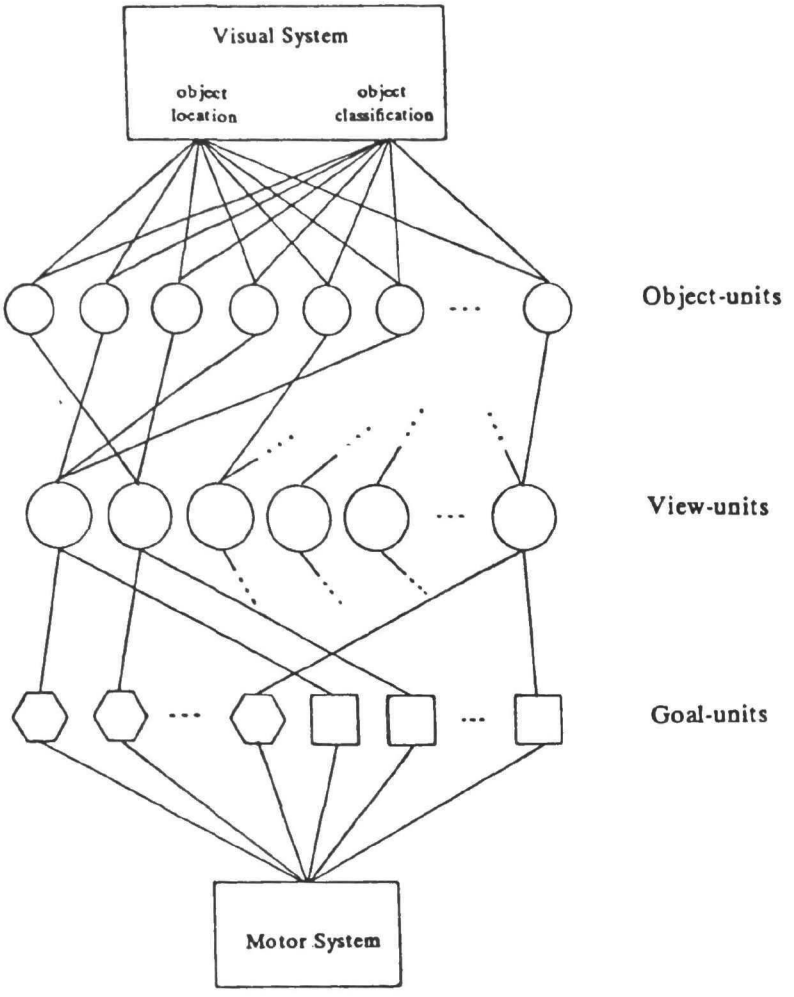
View-maps must first be built up by recording the location of goals at many viewpoints in the world. Then this information can be accessed by comparing the current view with the stored views to locate a desired goal. If an appropriate stored view does not exist in your view-map, it does not mean that you are lost. Often it is still possible to find a chain of views that overlap by several landmarks, so that a transformation can be generated to map the goal from a stored view to the current view, even though these two views do not contain the same landmarks. In this short paper I cannot discuss all the features of view-maps and the networks

---

This research was supported by a grant from the System Development Foundation.

that implement them. Rather I will describe a simple network which implements some key features of view-maps, such as view recognition and goal location.

In practice, it is often possible to locate goals without actually computing transformations or even reading out the stored locations of the visible landmarks. The network shown below was developed to demonstrate how this can be done.



The first layer consists of object-units which receive input from the sensory system and become active when a particular landmark appears at a specific location in the current reference frame of the viewer. The second layer consists of view-units, each of which receive input from several object-units. View-units recognize the fact that the observer is in the vicinity of a particular place. The activity of the view units can be used to answer such questions as "am I at the site where I buried that stuff?" The output of the view units are connected to a layer of goal-units which, when activated, can tell the motor system how to get from the viewpoint of a view-unit to the goal they represent, i.e., "how do I get home from here?" There must be a separate goal-unit for each goal at each viewpoint. Sets of goal-units representing different goals are shown with different shapes.

Each class of units in this network has an associated function which determines how its output activity varies with its inputs. For example, the activity function for the view-units is just the thresholded sum of their input activities. The value of the threshold is chosen so that

several landmarks must be recognized in approximately the expected locations for a view-unit to become active.

A more complex function is required for object-nodes because they must recognize the presence of a landmark and the degree to which its current location matches the expected location. This function should have a maximum when the viewer is in exactly the expected location and then fall off gradually as the viewer moves from this location. The function should be zero at all times when the expected object is not being viewed at all. A matching function I have used extensively, is:

$$\text{Object-unit activity} \begin{cases} = \exp\text{-}\left[(\text{current object location} - \text{expected object location})^2/\sigma^2\right] \\ \text{; if object detected.} \\ = 0\text{; if object not detected.} \end{cases}$$

The values of the threshold of the view-unit activation function together with the σ of the object-function determine the size of the region in space in which a particular view-unit will become active. If this region is small the observer will have accurate knowledge of position but it will be available over a small area. If the region is large, less accurate information will be available over a wider area.

I have used computer simulations (Rabin & Zipser, 1983) in which the movement of an observer is guided by networks of the type shown above to investigate how view-maps deal with problems such as recognizing a location as previously visited and getting from the current location to a goal. The first of these problems requires quantitative matching of the current view to all previously recorded views. Because of the structure of the network used, this match occurs in parallel so that each view-unit always indicates, by its output activity, the degree to which the current view matches the view at its viewpoint. Of course, at any particular location most view-units are inactive. The viewer determines if return to the desired location has occurred by sensing the activity of the appropriate view-unit. This is how, for example, the location of previously buried stuff can be located. The computer simulation demonstrated that there was a "place-field" around the location represented by each view-unit in which the activity of the unit increases as the viewer approaches the viewpoint. This makes view-units similar to the spatial field neurons in the hippocampus (Muller, Kubie & Ranck, Jr., 1983).

To answer such questions as "how do I get home from here?" requires that the location of a landmark not currently visible, i.e., home, be determined. If view-units which connect to goal-units for the desired unseen location have been recorded fairly evenly over the environment, then at any location the viewer will activate several view-units that refer to the goal. An estimate of the location of the goal can be made by forming an average of the location stored in each of these goal-units, weighted by the activity of its connected view-unit. Computer simulations showed that the use of this weighted average value of the goal location is an effective way to determine where the goal is when the viewer is far away from the goal. However, when the observer gets very close to the goal, there are serious difficulties because some of the viewpoints used are beyond the goal and thus give the wrong sign to goal direction. When the observer reaches the vicinity of the goal, motion becomes erratic. However, sooner or later the simulated observer generally finds a path into the goal. Behavior of the sort that occurred in these simulations is not too unreasonable since, when far away from a goal, an observer cannot see it and is then forced to use landmarks. When the observer gets close enough to actually see the goal, landmarks are no longer needed.

A more formal analysis of view-maps, not presented here, has been carried out which shows that they can serve as a robust representation of the spatial organization of objects. Several important issues have been analyzed to varying degrees and also studied with computer simulation (Zipser, 1983). Among these are: what is a landmark? How are object-units, goal-units and view-units learned? What features are used by the sensory system to localize landmarks? How is it possible to get to a goal whose location is not known to any view-unit currently active?

The object-unit to view-unit hierarchy provides a very general mechanism for representing the spatial location of objects. It has been shown here how it can be used to construct view-maps which represent the spatial organization of landmarks in the environment. It can also be used to represent the locations of features within a single object and in this way they may be useful for the recognition of objects. View-units can also be considered as schemata in which information about location is given major prominence (Brewer & Treyens, 1981). Viewed as schemata or frames, view-units can be thought of as having additional information besides that already discussed. For example, view-units might connect to units indicating the suitability of place fields, for example, dangerous or safe. The activation of a particular view-unit as an observer moves through the environment would then immediately give access to information about the desirability of remaining at that location. In general, any variable quantity connected to view-units would be accessible when the observer returned to a location in which the view-unit was activated. This corresponds to the common experience of recalling events or even thoughts that occurred at specific locations (Nigro & Neisser, 1983).

REFERENCES

Brewer W. F., & Treyens, J. C. (1981). Role of schemata in memory for places. *Cognitive Psychology, 13,* 207-230.

Kuipers, B. (1983). Modeling human knowledge of routes: Partial knowledge and individual variation. *Proceedings of the National Conference on Artificial Intelligence, August 22-26, 1983 (AAAI-83)* (pp. 216-219). Los Altos, CA: William Kaufmann, Inc.

McDermott, D. (1980). *Spatial inferences with ground, metric formulas on simple objects* (Res. Rep. No. 173). New Haven: Yale University, Department of Computer Science.

Muller, R. U., Kubie, J. L., & Rank, J. B., Jr. (1983). High resolution mapping of the 'spatial' fields of hippocampal neurons in the freely moving rat. *Society for Neuroscience* Abstract Number 191.4.

Nigro, G., & Neisser, V. (1983). Point of view in personal memories. *Cognitive Psychology, 15,* 467-482.

Rabin, D., & Zipser, D. (in preparation). *P3 - Parallel process programmer*. (Tech. Rep.). La Jolla: University of California, San Diego, Institute for Cognitive Science.

Zipser, D. (1983). *The representation of maps* (ICS Rep. No. 8303). La Jolla: University of California, San Diego, Institute for Cognitive Science.

A Model of Early Chemical Reasoning[1]

Jan Zytkow
Pat Langley
Herbert A. Simon
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213 USA

1. Introduction

During the 18th Century, one of the primary goals of chemistry was to determine the *components* of substances. This was a long and painstaking process, during which many different models were proposed and rejected. Throughout most of the 18th century combustion was believed to involve the decomposition of the combustible body, and this was one of the central tenets of the theory of *phlogiston*. Only in the last two decades of the 18th century was the phlogiston theory challenged and eventually replaced by the *oxygen* theory. In this paper we describe STAHL, a cognitive simulation that models the inferences made by early chemists. The system is named after G. E. Stahl (1660 1734), one of the principal formulators of the phlogiston theory. In the following pages we describe STAHL in terms of its component heuristics, and trace its reasoning on a number of episodes from the history of the phlogiston theory.

2. An Overview of STAHL

STAHL's input consists of a set of reactions, each represented by a simple schema. For instance, the reaction of sulphur and iron to form sulphuretted-iron would be represented as (react (input sulphur iron) (output sulphuretted-iron)). STAHL's goal is to determine the components of all non-elemental substances involved in the given list of reactions. In the table below we present six heuristics used by STAHL in inferring the componential models of substances.

STAHL's heuristics for inferring the components of substances.

INFER-COMPOSITION
If A and B react to form C,
or if C decomposes into A and B,
then infer that C is composed of A and B.

REDUCE
If A occurs on both sides of a reaction,
then remove A from the reaction.

SUBSTITUTE
If A occurs in a reaction,
and A is composed of B and C,
then replace A with B and C.

EQUATE-DECOMPOSITIONS
If A is composed of B and C,
and A is composed of D and E,
then infer that B and C react to form D and E.

IDENTIFY-COMPONENTS
If A is composed of B and C,
and A is composed of B and D,
then identify C with D.

IDENTIFY-COMPOUND
If A is composed of C and D,
and B is composed of C and D,
then identify A with B.

The most basic of the rules, INFER-COMPOSITION, deals with simple synthesis and decomposition reactions, and lets the system unambiguously infer the components of a compound. For example, given the sulphuretted-iron formation reaction, this rule would infer that sulphuretted-iron consists of sulphur and iron. Of course, the INFER-COMPOSITION rule can also deal with cases in which three or more substances unite to form a simple compound, and with similar decompositions.

---

[1]This research was supported by Contract N00014-84-C-50767 from the Office of Naval Research. J. Zytkow is a member of the Institute of Philosophy at the University of Warsaw and of the CMU Psychology Department, H. A. Simon is a member of the CMU Psychology Department, while P. Langley is associated with the CMU Robotics Institute.

For more complex reactions, STAHL employs other rules to transform these reactions into simpler forms, so they can eventually be matched by INFER-COMPOSITION. Thus, the REDUCE heuristic is responsible for "cancelling out" substances occurring on both sides of a reaction, leading to a simplified version. For instance, given the input (react (input calx-of-iron vitriolic-acid water) (output vitriol-of-iron water))[2], the REDUCE rule would produce the simplified description (react (input calx-of-iron vitriolic-acid) (output vitriol-of-iron)). In turn the INFER-COMPOSITION rule would conclude that vitriol-of-iron consists of calx-of-iron and vitriolic-acid.

The third heuristic (SUBSTITUTE) initially leads to more complex statements of reactions, but may enable the REDUCE rule to apply. This rule draws on information about the components of a substance that have been inferred earlier. Eg, suppose that, in addition to the last example, STAHL knows that (react (input iron vitriolic-acid water) (output vitriol-of-iron inflammable-air water)). Now, by REDUCE (applied to water) and then by SUBSTITUTE (applied to vitriol-of-iron) the system infers that iron consists of calx-of-iron and inflammable-air. This is in fact the conclusion drawn originally by Cavendish [1766], after he discovered hydrogen while dissolving metals in acids.

Let us now, based on the three rules described above, consider the origins of the theory of phlogiston early in the 18th century. G. E. Stahl adopted the ancient view that fire is a manifestation of a common principle which leaves a body during combustion. Therefore, any reaction involving combustion was viewed as a decomposition; for instance, burning charcoal was interpreted as decomposing it into phlogiston (another term for the matter of fire) and some residual ash. G. E. Stahl succeeded in proving the usefulness of the notion of phlogiston in explaining many reactions that were by no means reactions of combustion, and in justifying the presence of phlogiston in substances that were not combustibles.

Let us examine the path taken by STAHL in ariving at the initial conclusions of the human Stahl's theory of phlogiston. We present the system with two facts: (react (input charcoal air) (output phlogiston ash air)) and (react (input calx-of-iron charcoal air) (output iron ash air))[3]. Given both reactions, STAHL immediately applies its REDUCE heuristic to the first fact, giving the revised reaction (reacts (input charcoal) (outputs phlogiston ash)). The system then applies the same rule to the second fact, giving the reduced reaction (reacts (input calx-of-iron charcoal) (output iron ash)). After this, the first of these revisions, combined with the INFER-COMPOSITION rule, leads to the inference that charcoal is composed of phlogiston and ash, which was one tenet of the early phlogiston theory. Having arrived at this conclusion, STAHL applies SUBSTITUTE, generating the expanded relation (reacts (input calx-of-iron ash phlogiston) (output iron ash)). At this point, REDUCE is used to remove ash from both sides of the equation, giving (reacts (input calx-of-iron phlogiston) (output iron)). Finally, INFER-COMPOSITION leads STAHL to infer that iron is a compound composed of calx-of-iron and phlogiston. If, at this point, STAHL is given the reaction (react (input calx-of-mercury iron) (output mercury calx-of-iron)), in which neither phlogiston nor charcoal is explicitly present, the system infers that mercury consists of calx-of-mercury and phlogiston.

Now we are able to reproduce a historically valid application of EQUATE-DECOMPOSITIONS. Sulphur, as a combustible, was believed to contain phlogiston. To demonstrate that its remaining component is vitriolic-acid, G. E. Stahl refers to the following reactions [Partington, 1961, p. 671]:

(react (input vitriolic-acid potash) (output vitriolated-tartar)),
(react (input sulphur potash) (output liver-of-sulphur)),
(react (input vitriolated-tartar charcoal) (output liver-of-sulphur)).

---

[2]The following "dictionary" may be helpful in understanding our historic examples: metallic calxes are 18th century terms for metallic oxides, inflammable air is hydrogen, dephlogisticated air is oxygen, vitriol of iron is iron sulphate, vitriolated tartar is potassium sulphate, and liver of sulphur is a mixture of potassium polysulfides with potassium thiosulfate.

[3]Chemists of the early 18th century acknowledged the necessity of air in combustion as the acceptor of phlogiston and believed that combustion in a closed vessel stops at some point when air gets saturated with phlogiston and cannot accept more of this principle.

All three reactions match the INFER-COMPOSITION rule, and as the result, STAHL produces two different decompositions of liver-of-sulphur. This activates the EQUATE-DECOMPOSITIONS rule and now STAHL considers the additional reaction: (react (input sulphur potash) (output vitriolated-tartar charcoal)). In this reaction, SUBSTITUTION applies to both vitriolated-tartar and charcoal, creating (react (input sulphur potash) (output vitriolic-acid potash ash phlogiston)). Reduction of potash on both sides enables INFER-COMPOSITION to apply, and to conclude that sulphur consists of vitriolic-acid, and phlogiston (and ash, unless we ignore residual substances or use soot instead of charcoal, as the (almost) pure source of phlogiston. STAHL in its present form cannot deal effectively with residual substances or impurities).

The final two heuristics are responsible for postulating that two substances that were originally thought to be different are in fact identical. For instance, the IDENTIFY-COMPONENTS rule matches when STAHL learns that a compound can be decomposed in two different ways, where these decompositions differ by a only single substance. Our examples enable it to apply this rule to iron, for which STAHL has already inferred two different compositions: into calx-of-iron plus inflammable-air and into calx-of-iron plus phlogiston. Identification of inflammable-air with phlogiston, made at this point by STAHL, was indeed a historic fact, and was claimed from 1766 until the final rejection of the phlogiston theory in the 1790's.

Given a set of reactions as input, STAHL applies its heuristics to these reactions until it has made as many inferences as possible. Then the system halts, but it can accept additional input reactions and can make additional inferences. At any given point during the computation, the system's knowledge consists of some mixture of observed reactions, transformed reactions, and componential models. One of the system's interesting features is the manner in which its heuristics interact. Note that the SUBSTITUTION rule (and some other rules, too) requires knowledge of a substance's composition, so that some inferences about composition must be made before it can be used. We have also seen that complex reactions must be rewritten by the REDUCTION and SUBSTITUTION rules before some composition inferences can be made. This interdependence leads to a "bootstrapping" effect, in which inferences made by one of the rules enable further inferences to be made, these allow additional inferences, and so forth. This process begins with one or more simple reactions, but after this the particular path taken depends on the data available to the system.

3. Automated Self-Correction

Although STAHL's heuristics provide useful direction through the space of possible chemical models, they are not guaranteed to produce correct inferences. For instance, the system may apply the REDUCE rule when different quantities of a substance occur on both sides of a reaction, leading to errorful conclusions. However, similar confusions also occurred in the history of chemistry. As late as 1810, Gay-Lussac and Thenard [1810] argued that potassium was a compound of potash and hydrogen, contrary to Davy's claim that potash was a compound of potassium and oxygen. They based their argument on the following reactions:

(react (input potassium water) (output caustic-potash hydrogen water)),
(react (input caustic-potash water) (output potassium oxygen)),
(react (input potassium ammonia) (output hydrogen green-solid)),
(react (input green-solid water) (output caustic-potash ammonia water)).

Only the second reaction is not acceptable from today's point of view. The motivation for this description was Gay-Lussac's and Thenard's disbelief that potash, known to have an extremely strong affinity to water, can be totally destitute of water. Given these four reactions, STAHL is capable of inferring their conclusion about the composition of potassium in three independent ways: from the first reaction alone, from the second reaction (if it is known in advance that water consists of hydrogen and oxygen), and from the last pair of reactions.

Of course, chemists eventually realize their errors and recover from them, and STAHL incorporates strategies to do the same. If, during the computation, a reaction is transformed to the form of empty input: (react (input) (output A)), STAHL enters its error-recovery procedure. There are several other states of the system's knowledge that activate the same procedure: obviously (react (input A) (output)), but also, for

instance, the pair of componential models "B consists of A and X" and "A consists of B and Y". This particular error will be detected by STAHL. Suppose that, in addition to the reactions relevant to the theory of phlogiston and considered in this paper, the system is given the result of the famous Priestley experiment on the decomposition of calx of mercury: (react (input calx-of-mercury) (output mercury dephlogisticated-air)). Now the system knows that calx-of-mercury consists of mercury and dephlogisticated-air, but also, what was inferred earlier, that mercury consists of calx-of-mercury and phlogiston. Taken together, these two inferences imply that calx-of-mercury is composed of itself, plus phlogiston, plus dephlogisticated-air, a troublesome conclusion. Historically, the discovery of oxygen and the very same conclusion drawn by phlogistians stimulated an important transformation of the phlogiston theory, and was a starting point to the construction of the oxygen theory of combustion.

In order to recover from an error, STAHL searches through the list of reactions and collects the reactions which may be responsible for this particular error. Then STAHL processes these reactions once again, using the knowledge the system has already inferred. This may be enough to make conclusions that are free from the error. If not, as in the case of calx-of-mercury, then the system introduces a new conceptual distinction. Based on the observational-theoretic distinction STAHL replaces one of the occurrences of calx-of-mercury by the concept *calx-of-mercury-proper*. Such concepts are tautological when first introduced, but become respectable to the extent that they prove useful in dealing with other situations besides the one leading to their introduction. The compositional model of calx-of-mercury created by this move reads: "calx-of-mercury consists-of calx-of-mercury-proper, phlogiston, and dephlogisticated-air", and enables satisfactory explanation of all reactions within the framework of the phlogiston theory. If the second strategy does not allow recovery from the error, the system requests the repetition of the experiments that are responsible for the error. All these error recovery methods contribute signficantly to STAHL's status as a historical model, since such reformulations occurred many times in the early days of chemistry.

4. Conclusions

STAHL is a cognitive simulation that models significant portions of 18th century reasoning about the composition of substances . We have tested the system on reactions between acids, alkalis, and salts, and it has successfully modeled major developments in the domain of combustion and reduction. STAHL can reproduce successful reasoning, as well as failures which originally were believed to be successes. The system can work on incremental data, drawing conclusions from a given list of reactions, then processing another list of reactions, and so forth; each time it gains additional knowledge, and each time its performance depends on the knowledge it has already cumulated. This feature of STAHL makes it possible to simulate alternative theoretical developments, such as different responses of Lavoisier and phlogistians to the discovery of oxygen. All these findings enable us to conclude that STAHL captures a general method of 18th century reasoning about components of substances, and to refute the claim that establishing the oxygen theory of combustion was only possible because of the new chemical method developed by Lavoisier.

In our presentation of STAHL, we concentrated on the set of rules it incorporates, and, due to limited space, we discussed only briefly the control mechanism of STAHL and its automated self-correction strategy. However, the control mechanism is vital in assuring historically adequate and cognitively plausible behavior of the program. Therefore, the control strategy seems to constitute an important aspect of scientific reasoning, and as future programs come to incorporate more rules and simulate more aspects of theoretic activity in science, the control startegy may become a crucial point in programming such systems.

5. References

Cavendish, H., Three Papers,..., Phil.Trans., 56, 1766, pp.141-84.
Gay-Lussac, L.P., and Thenard, L.J., Annales de chimie, 75, 1810, pp.290-316.
Partington, J.R., A History of Chemistry, vol.2. London, 1961.

NEW AND FORTHCOMING BOOKS IN COGNITIVE SCIENCE

MEMORY AND THE BRAIN
By Magda B. Arnold
290-9 5/84 544 pp. $49.95

UNDERSTANDING EXPOSITORY TEXT:
A Theoretical and Practical Handbook for Analyzing Explanatory Text
Edited by Bruce K. Britton & John B. Black
412-X 11/84 c. 420 pp.

A HANDBOOK OF COGNITIVE PSYCHOLOGY
By Michael W. Eysenck
7016-9 5/84 c. 256 pp. (cloth)
7017-7 5/84 c. 256 pp. (paper)

THOUGHT AND KNOWLEDGE:
An Introduction to Critical Thinking
By Diane F. Halpern
514-2 6/84 c. 400 pp. $24.95 (paper)

ORTHOGRAPHIES AND READING:
Perspectives from Cognitive Psychology, Neuropsychology, and Linguistics
Edited by Leslie Henderson
7009-6 3/84 160 pp.

COMPARATIVE PERSPECTIVES ON THE DEVELOPMENT OF MEMORY
Edited by Robert Kail & Norman E. Spear
317-4 3/84 384 pp. $39.95

METHODS AND TACTICS IN COGNITIVE SCIENCE
Edited by Walter Kintsch, James R. Miller & Peter G. Polson
327-1 5/84 336 pp. $39.95

RETRIEVAL AND ORGANIZATIONAL STRATEGIES IN CONCEPTUAL MEMORY: A Computer Model
By Janet L. Kolodner
365-4 4/84 280 pp. $29.95

STORIES, SCRIPTS, AND SCENES:
Aspects of Schema Theory
By Jean M. Mandler
446-4 8/84 c. 160 pp. c. $16.95

NEONATE COGNITION
Edited by Jacques Mehler & Robin Fox
345-X 8/84 c. 480 pp.

ANIMAL COGNITION
Edited by H. L. Roitblat, T. G. Bever & H. S. Terrace
334-4 2/84 696 pp. $49.95 (cloth)
407-3 2/84 696 pp. $19.95 (paper)

PROBLEM SOLVING AND INTELLIGENCE
By Helga A. H. Rowe
347-6 5/84 c. 432 pp

ORIGINS OF COGNITIVE SKILLS:
18th Annual Carnegie Symposium on Cognition
Edited by Catherine Sophian
390-5 8/84 c. 420 pp c. $39.95

MEMORY CONSOLIDATION:
Psychobiology of Cognition
Edited by Herbert Weingartner & Elizabeth S. Parker
323-9 6/84 c. 288 pp. c. $29.95

HANDBOOK OF SOCIAL COGNITION
Edited by Robert S. Wyer, Jr. & Thomas K. Srull
Volume 1:
338-7 5/84 c. 300 pp.
Volume 2:
339-5 5/84 c. 300 pp.
Volume 3:
340-9 5/84 c. 300 pp.

See these and other works on display in our booth at the annual meeting

LAWRENCE ERLBAUM ASSOCIATES, INC., PUBLISHERS
365 Broadway, Hillsdale, New Jersey 07642

New and recent titles from

INDIVIDUAL DIFFERENCES IN COGNITION
Volume 1
Edited by RONNA F. DILLON and RONALD R. SCHMECK
With a foreword by ROBERT GLASSER
1983, 336 pp., $34.00/ISBN: 0-12-216401-6

SYMBOLIC PLAY
The Development of Social Understanding
Edited by INGE BRETHERTON
1984, 392 pp., $44.50/ISBN: 0-12-132680-2

STRATEGIES OF DISCOURSE COMPREHENSION
TEUN A. VAN DIJK and WALTER KINTSCH
1983, 448 pp., $38.50/ISBN: 0-12-712050-5

Two volumes in the ACADEMIC PRESS SERIES IN COGNITION AND PERCEPTION

VARIETIES OF ATTENTION
Edited by RAJA PARASURAMAN and D. R. DAVIEW
1984, 572 pp., $58.50/ISBN: 0-12-544970-4

EVALUATION OF DIAGNOSTIC SYSTEMS
Methods from Signal Detection Theory
JOHN A. SWETS and RONALD M. PICKETT
1982, 272 pp., $40.00/ISBN: 0-12-679080-9

NEURAL MODELS OF LANGUAGE PROCESSES
Edited by MICHAEL A. ARBIB, DAVID CAPLAN and JOHN C. MARSHALL
A Volume in the PERSPECTIVES IN NEUROLINGUISTICS, NEUROPSYCHOLOGY AND PSYCHOLINGUISTICS Series
1982, 592 pp., $54.50/ISBN: 0-12-059780-2

LANGUAGE PRODUCTION
Edited by BRIAN BUTTERWORTH
Volume 1: Speech and Talk
1980, 478 pp., $69.50/ISBN: 0-12-147501-8
Volume 2: Development, Writing and Other Language Processes
1983, 320 pp., $45.00/ISBN: 0-12-147502-6

ASPECTS OF CONSCIOUSNESS
Volume 3
Awareness and Self-Awareness
Edited by GEOFFREY UNDERWOOD
1982, 346 pp., $35.00/ISBN: 0-12-708803-2

Titles in the COMPUTERS AND PEOPLE Series

DESIGNING FOR HUMAN-COMPUTER COMMUNICATION
Edited by M. E. SIME and M. J. COOMBS
1983, 348 pp., $48.00/ISBN: 0-12-643820-X

PRINCIPLES OF COMPUTER SPEECH
IAN H. WITTEN
1983, 304 pp., $32.00/ISBN: 0-12-760760-9

INTELLIGENT TUTORING SYSTEMS
Edited by D. SLEEMAN and J. S. BROWN
1982, 368 pp., $39.50/ISBN: 0-12-648680-8

FUZZY REASONING AND ITS APPLICATIONS
Edited by E. H. MAMDANI and B. R. GAINES
1981, 384 pp., $39.50/ISBN: 0-12-467750-9

COMPUTING SKILLS AND THE USER INTERFACE
Edited by M. I. COOMBS and T. L. ALTY
1981, 499 pp., $55.00/ISBN: 0-12-186520-7

Two volumes in the INFORMATION INTEGRATION Series

METHODS OF INFORMATION INTEGRATION THEORY
NORMAN H. ANDERSON
1982, 464 pp., $44.00/ISBN: 0-12-058102-7

FOUNDATIONS OF INFORMATION INTEGRATION THEORY
NORMAN H. ANDERSON
1981, 440 pp., $41.50/ISBN: 0-12-058101-9

CLASSROOM COMPUTERS AND COGNITIVE SCIENCE
Edited by ALEX CHERRY WILKINSON
A volume in the EDUCATIONAL TECHNOLOGY Series
1983, 232 pp., $28.00/ISBN: 0-12-752070-8

HUMAN AND MACHINE VISION
Edited by JACOB BECK, BARBARA HOPE, and AZRIEL ROSENFELD
A volume in the NOTES AND REPORTS IN COMPUTER SCIENCE AND APPLIED MATHEMATICS Series
1983, 584 pp., $42.00/ISBN: 0-12-084320-X