COGNITIVE SCIENCE 4, 243–284 (1980)

Language and Memory*

ROGER C. SCHANK
Yale University

This paper outlines some of the issues and basic philosophy that have guided my work and that of my students in the last ten years. It describes the progression of conceptual representational theories developed during that time, as well as some of the research models built to implement those theories. The paper concludes with a discussion of my most recent work in the area of modelling memory. It presents a theory of MOPs (Memory Organization Packets), which serve as both processors and organizers of information in memory. This enables effective categorization of experiences in episodic memory, which in turn enables better predictive understanding of new experiences.

PREFACE

As an undergraduate, I naturally developed a simultaneous interest in the problem of cognition and its computer simulation. I also had a strong interest in language. Attempting to combine these three interests led me to the conclusion that there existed no academic discipline that could comfortably accommodate my interests. Linguists were not seriously interested in cognition. Psychologists were, but did not take seriously the idea of a computer program as the embodiment of a theory. Computer Science was still nascent and in many ways resistant to the "mushiness" of Artificial Intelligence (AI). Where AI did exist, concern with people as opposed to machines was frequently lacking.

In the last few years the situation in all three fields has begun to change. In AI, cognitive concerns have not only been accepted but are considered to be of prime importance. Many linguists have abandoned their overriding concern with syntax for a more balanced view of language phenomena. Psychologists are

*I would like to thank the following people for their help both in the writing of and the information of ideas in this paper: Wendy Lehnert, Christopher Riesbeck, Robert Abelson, Michael Lebowitz, Janet Kolodner, Mark Burstein, Lawrence Birnbaum, and Margot Flowers. I would also like to thank Donald Norman for his help in editing a previous version of this paper. This work was supported in part by the Advanced Research Projects Agency of the Department of Defense, monitored by the Office of Naval Research under contract N00014-75-C-1111.

learning how to build computer models themselves and have begun to run experiments to test hypotheses taken directly from work in AI.

What we are seeing is the beginning of Cognitive Science.

INTRODUCTION

In this paper I will attempt to outline some of the issues and basic philosophy that have guided my work and that of my students in the last ten years. I will end by outlining some of the problems that I am currently working on in the area of the modelling of memory.

My initial research focused on the representation of meaning as it would be used for the generation of natural language sentences. I believed (and still do believe) that because people could easily translate from one language to another and, in a sense, think in neither, there must be available to the mind an interlingual, i.e., language-free, representation of meaning. I was very interested in the problem of mechanical translation of language and hoped that any representation I developed would be useful for solving that problem. Since I was looking for this interlingual representation based upon the assumption that people actually thought with such a thing, I developed an intense interest in making any representation I came up with as psychologically correct as possible. Unfortunately, psychologists were at this point very concerned with phenomena that could shed light on the validity of transformational grammars (e.g., Fodor et al., 1966; Mehler, 1963). This work did not provide much in the way of evidence one way or the other for the things I was interested in, so, since I was not trained to do experiments myself, I had only my intuitions to rely on for psychological evidence.

I began to think about the problem of representing meaning; but, since my guiding interest at the time was mechanical translation, I was particularly interested in the computational properties of any representation that I came up with. I was especially concerned with the question of how a meaning representation could be of use in the generation of natural language sentences and in the parsing of natural language sentences.

The first representation that I developed looked a lot like English words with arrows connecting them. The arrows were taken from dependency theory, which had been written about by Hays (1964) and used quite a bit by Klein (1965) and, to some extent, Lamb (1964). My contribution, as I saw it at that time, was to make the representation more conceptual.

The main claim that Conceptual Dependency made at that time (Schank, 1969) had nothing to do with the primitives with which the work has been primarily associated in recent times. Conceptual Dependency theory claimed that there was a predetermined set of possible relationships that made up an interlingual meaning structure. These relationships (or conceptual rules as I termed them) could be used either to predict conceptual items that were implicit in a sentence or, coupled with syntactic rules, to inform a parser what was missing from a meaning and where it might be found in a sentence (Schank & Tesler, 1969). In generation, these rules could be used as the basis for generating meanings from which sentences could be formed. This changed generation to a process that was more realistic than one beginning with $S = NP + VP$ (Schank, 1968).

The key issue from my point of view, then—and my philosophy has not changed on this—was the creation of expectations about what slots needed to be filled in a conceptualization and the embodiment of those expectations to guide both parsing and generation. If information about the properties of a coherent meaning structure is available to parsers and generators, there seems no reason not to use it. Folk wisdom decided that I "didn't believe in syntax" because I pointed out the necessity of using meaning information to drive the parser. But, the only way to write such a parser is to use meaning-driven rules which have their basis in syntactic information about the input sentence. This is what we tried to do (Schank et al., 1973). The basic idea of slot filling and top down expectations drives our work today (Carbonell, 1979; DeJong, 1979; Gershman 1979; Riesbeck & Schank, 1976; Wilensky, 1978). The notion of scripts (Schank & Abelson, 1977), and to some extent frames (Minsky, 1975), uses this same basic philosophy.

As we began to work on building programs that mapped English into and out of Conceptual Dependency, we ran into a problem with ambiguous sentences whose resolution depended on world knowledge considerations. Prior to this time parsers were purely syntactic, so no good solution had been found for this problem.

As an example of the kind of issue I was concerned about at that time, consider the following sentences:

I hit Fred on the nose.
I hit Fred in the park.

In order to parse these sentences correctly it was necessary to know where a person can be located. Here, "correctly" depended on what had to be represented in CD. There was a locative relationship for entire conceptualizations and a "part of" relationship for objects, and either could be expressed in English with a locative prepositional phrase. To solve this problem I used the conceptual semantics I had invented for generation. [These were simple world knowledge rules that were tied to each CD conceptual rule (Schank, 1968). Thus the conceptual rule that actors can act would be modified by lists of what could do what according to semantic categories, such as "animals can eat," "planes can fly," and so on.] The rules that mapped from syntactic relationships to conceptual ones checked for acceptability according to the conceptual semantics each time a mapping was attempted.

Gradually, it became clear that the final parse of most sentences into Conceptual Dependency wound up adding information that was not in the original sentence explicitly. This took our work out of the domain of linguistics, since we had gone beyond language phenomena. This work was Cognitive Science, but since that field didn't then exist, a good home became a nontrivial problem.

CONCEPTUAL REPRESENTATIONS

In 1970 we started to make our representations more conceptual (Schank et al., 1970) than they had been. Until this point our supposedly language-free representations had a great deal of language in them. Our representation seemed to require us to put in a great deal more than was in the surface or deep structure representation of the sentence in order to make conceptual sense. There did not seem to be any way to avoid this introduction of elements that were not present in the utterance initially if we were to represent the meaning of what had been said. Examining our representations, we began the search for some regularities in the representation that would give us a more canonical form. What we had until that point was so free form that we could create anything at any time. This did not seem very sensible. In particular, there was a problem of determining which sense of the various multiple-sense verbs we had at any given time. We could not just continue writing "have" with subscripts to differentiate "have a soda" from "have ten dollars" from "have cancer." There had to be some underlying basic forms. Was "understand1" equal to "see3"? Which sense was more basic? And, more important, how many senses of a word would there turn out to be and what would their intersections be? In the case of partial overlap of senses, there was a definite problem with the subscript method.

As a side issue at this time, we attempted to clean up the mess in which we had left our representation of prepositions. We had been using an arrow to mean any prepositional relationship, in the faith that higher level processes that used our representations would figure out the true relationship that held between an action and its associated objects. We tried to think about what kinds of prepositional relationships there were.

We had already dealt adequately for our purposes with locations and "part of" relationships (Schank, 1969). Aside from these two classes of prepositions we found that there were only three kinds of prepositional relationships: instrumental, directional, and recipient. These relationships described the way an action could relate to an object in an event regardless of what preposition was being used. Since we were describing relationships and not prepositions, we realized that English could be considered to have a kind of null preposition denoting objective relationships. However, this objective relationship was not any less of a relationship between action and object than the others. We knew that Fillmore (1968) had said similar things about syntactic relationships in

English, so we christened our relationships "conceptual cases." The differences between the two systems were a lot greater than their names suggested and in retrospect this was probably a poor choice of names. [See Schank (1972) for a discussion of those differences.]

This new system of cases had immediate ramifications throughout our entire representation system. For example, we had previously represented "I want money" as:

```
           I <==>want
                 ↑
                 |         to
           money <==>go  < —  I
```

However, adding a recipient to this representation caused us to come up with the following representation:

```
I <==>want
      ↑                O          R I--->I
Someone <=>    ??    <--money< --I
                                   I---<someone
```

that is, we knew that we had a Recipient here and it had to be "I." Similarly there had to be an Object because what else could "money" be? It did not seem like an actor. The actor was unknown, but we knew he was the same person as the donor of the recipient case. Of course, the above diagram had a glaring hole. What was the action? Still this representation seemed to make a lot more sense than the first one where money was an actor doing the action "go."

What was needed at this point was a name for our unknown action, and since it obviously involved a kind of transfer of the money, we called it "trans."

"Trans" helped us with other problems as well. It solved the partial overlap problem in the meaning of words such as "give," "take," "buy," and "sell." Furthermore, it eliminated the need for elaborate meaning transfer rules of the kind Katz (1967) had been proposing for mapping words like "buy" into "sell." We began to wonder what other actions like "trans" were around.

We began at this point to look more closely at the concept of an action. We attempted to classify the verbs we had been using according to the cases they took and the properties of their objects. This left us with S(tate)-ACTS, P(hysical)-ACTs, E(motional)-ACTs, and so on (Schank et al., 1970). Using this classification for verbs, we could now predict the missing cases that were implicit and that thus had to be inferred. We continued to look for effective groupings that would facilitate inference. Thus, although we did not actually set out to discover primitives, the considerations that we had in representation issues forced us to come up with some workable classification of actions that fit within our framework.

Inference was not yet a major issue in this regard, but other problems forced us to focus on it. For example, consider the sentence "I fear bears" and our proposed representation of it at that time:

```
        I<=>fear
              ↑
bears<=>harm<--I
```

In the same paper where we were wrestling with the issue of representation of actions (Schank et al., 1970), we also introduced an idea we called "associative storage of concepts." In order to adequately represent sentences of the above type, it was necessary to have available a conceptualization that could serve as the object of the verb "fear." (At this point we viewed such a verb as a kind of stative ACT. We later realized such states were not ACTs but states of objects.) Obviously this conceptualization had to have in it both "bears" and "I" as part of the object of "fear." Here again we were faced with the question of what was the ACT? The answer we chose was an ACT called "harm." As we were not particularly interested in primitives, this should not seem strange. The focus of our interest was: how were we going to find the concept "harm" to add to our representation?

We tried using "associative storage of concepts." What we meant by this was that there had to be some connection between fear and bears that would allow us to infer "harm" as the missing ACT. Quillian (1966) had used the idea of a linked network of concepts that could be searched from two paths in order to find their shortest intersection. This idea had been used for disambiguation, but it now seemed that it could be extended for use here as well.

However, that seemed like a lot of work for so little. When we looked at other examples of the phenomenon we were trying to account for, an easier solution presented itself. For example, the sentence "I like books" clearly needed something about "I read books" inside the conceptualization that represented its meaning. It was obvious that this could be done simply by listing "books" in the dictionary as a "READ object." If we had an empty slot requiring an ACT, and an object of "book," we would simply infer "read." This depended on treating "like" as a state and not an ACT, of course.

This did not solve the problem when the object was not the source of the inference, however. A functional object like a "book" could well be listed as a "READ object," but what were we to do when "bears" or "Nixon" was the object of a stative ACT? Since these objects were not functional in the same way, it seemed that the missing ACT would have to be supplied as a part of the meaning of the word "fear." Here again we had, without quite intending to, decomposed the meaning of a word (fear) into more basic elements (fear plus expected harm). The reason for this was again attributable to the requirements we had placed on CD with respect to what slots there were in a conceptualization and how they were to be filled. So, we were left at this point with a representation like:

```
        I fear Nixon
        I<=>fear
              ↑
        Nixon<=>do
               ///
something <=>harm<——I
```

Thus at this point we were now freely adding to our representation concepts that were not present in the English sentence in the first place and, perhaps more importantly, concepts that were only probably part of the meaning. These were the first explicit inferences that we had.

INTENTIONS, BELIEFS, AND MEMORY

We began to focus on the problem of inferencing intentions (Schank, 1971). We got into this problem because of a peculiar use of language that we happened to come across, which we realized was crucial for a reasonable understanding system to handle. The example was:

Q: Do you want a piece of chocolate?
A: I just had an ice cream cone.

Clearly, it is necessary to understand the answer given here as meaning ''no.'' In attempting to figure out how to do this, we realized that it was necessary to fill out the structure of the conceptualizations underlying both sentences so that a match could be made from the answer to the question. To do this required inferences that were different from the ''fill in the ACT'' ones we had been working on. Thus we needed a structure like:

want → trans → eat → satisfied

To get this structure we had to postulate that when a ''trans'' was present, the object of the ''trans'' might enable an actor to perform the usual functional ACT done to this object. Furthermore we had to examine the result of this action, because whatever state it caused was the key for the pattern match. That is, a paraphrase of this question might be: ''Do you want me to 'trans' you an object which is edible so you can eat it so that it will make you feel some feeling (full, happy, etc.)?'' The answer would then be: ''I already have that feeling because I just did an action that resulted in that feeling.'' To do all this required a new set of resultative and enabling inferences, and caused us to begin to focus on the question of what kinds of inferences there were and where they came from.

One of the first issues, however, was the potential use of such inferences. Since we were primarily concerned with parsing at this stage, we focused initially on the issue of what expectations existed in processing that came from places other than the CD or syntactic expectations themselves.

We looked at an example of a conversation where a person, in a fit of anger at his wife, asks his friend for a knife and, when he is refused it, says:

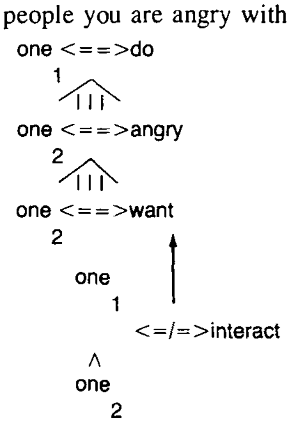
I think I ought to . . .

The question we asked was: What different kinds of things do you expect at this point? We isolated the following (Schank, 1971):

1. sentential	a verb is coming
2. conceptual	an entire conceptualization is coming
3. contextual	"ought to have fish" is excluded by the fighting context—something violent is expected
4. conversational	inference about the reason the person is talking—why talk about your future violence unless you want someone to stop it?
5. memory	what kind of person is John? should we take his anger seriously?
6. cultural	what happens in situations of this sort? memory structure inferences are used

These questions started us looking seriously at what else was going on in understanding besides parsing. Clearly we needed a memory full of facts about the world to do any sensible understanding. At this point our focus began to change. The issues of representation and parsing still existed of course, but memory, belief and inference were obviously at least as crucial.

Hemphill (1975) began to work on identifying how parsing was influenced by beliefs implicitly referred to in a text. I concentrated my efforts on representation; in particular, it was necessary, in order to handle the above example, to postulate a set of beliefs that could account for our expectations about an actor's behavior. To understand that John was not likely to want to sit down now and be friendly in the above example, we needed to know that when you are angry you do not like to be with the people you are angry with. This was represented as:



Beliefs of this sort were useful for predicting the future actions of an actor. Adding beliefs to the representation changed the idea of inference from that of simply additional information that would help in the parsing of a sentence. It suggested instead that we had to concentrate on problems having to do with the representation of information in memory and with the overall integration of incoming data with a given memory model. It thus became clear that natural language processing was a bit of a misnomer for our enterprise. What we were doing was not essentially different from what Colby (1967) or Abelson and Carroll (1965) were doing. That is, we had to deal with the problem of belief systems in general. But added to that was the problem of representation of meaning, knowledge, and memory.

The integration of all these problems caused us to deal with sentences whose meaning was a product of the combination of all these issues. For example, "He acts like Harry" means different things if Harry is a cat, a child, or an aged man. What is the correct representation for the meaning of such a sentence? Clearly it cannot be determined in any way apart from the memory structures its meaning relies on. Similarly, the sentence "He is dog-like in his devotion" means nothing if there is no belief available in memory about the devotion (or lack of it) of dogs.

We thus began to work on issues of memory and belief. But, in order to do this, we needed an adequate language for encoding beliefs and memory in general.

We returned to attempting to make the CD representations that we were using more rigorous so that we could better establish what was within the domain of a system like CD and what was outside of it. To do this, we considered the nature of the ACTs we had been using. At that point we had been using "trans" and a hodgepodge of others that suited us. To remedy this situation we looked at the mental verbs which we had, to this point, virtually ignored.

The significance of the primitive ACTs for us was that we could now be sure that we had a given agreed-upon representation for most of the sentences we were dealing with. This made our system usable by the large group of students who were beginning to concern themselves with programming systems that could communicate with each other. Further, we now knew what was in the bounds of the theory and what was not. We knew that to do the kind of work we were interested in, a canonical form was necessary. We were not as concerned with the ultimate correctness of that system as we were with its usability. No other canonical form existed, and transformational deep structure representations and predicate calculus, which were the major well-known alternatives, neither adequately represented meaning nor were in any sense canonical. The most important part of the primitives of CD for us was that they facilitated our getting on to the more interesting problems at hand. They did this because they gave us a language in which to describe those problems.

ORGANIZING INFERENCES

Eventually we began to realize that the most important problem in natural language processing was inference. The single most important fact about the primitive ACTs was that they helped to organize the inference problem (Schank, 1973). No primitive ACT meant anything in the system at all, other than the conceptualizations that might come to exist as inferences from it. Primitive ACTs served to organize the inference process, thus giving us a starting point from which to attack the problem.

We began to concern ourselves therefore with two principle kinds of inference: results from ACTs and enablements for ACTs. Then, having exhausted the inferences derivable from the ACTs themselves, we began to attempt to categorize the kinds of inferences that needed to be made in general. In Schank and Rieger (1974) we delimited 12 kinds of inference. Using these ideas, Rieger, Riesbeck, Goldman, and I began to design a computer implementation of these ideas in 1972 which resulted in the MARGIE system (Schank et al., 1973). During the implementation of these ideas our views on parsing, generation, and inference were altered by the task of attempting to specify precise algorithms for these processes. Rieger created a new classification of inferences based on his experiences with MARGIE (Rieger, 1975).

CAUSALITY

At this point we began to take seriously the problem of codifying the kinds of causal relations that there were. This work was crucial to the inference problem since we had come to believe that the major inferences were (forward) consequences and (backward) reasons. Thus the primary task of the inference process was to fill in causal chains. We identified four kinds of causal links: RESULT, REASON, INITIATE, and ENABLE. RESULT and ENABLE were the forward and backward causal rules for physical ACTs, and REASON and INITIATE were the forward and backward links for mental ACTs. We also added the rule that ACTs could only result in states and only states could enable ACTs. This had the consequence of making our causal chains, and thus our CD representations, both very precise and very cumbersome. The precision was of course important for any canonical form, but the cumbersomeness was obviously a problem that needed to be dealt with.

One of the advantages of all the detail necessary to connect all possible causal relations, aside from those already mentioned is that it provided a facility for tying together sentences in a text. Thus, a paragraph will frequently consist of a series of conceptualizations that can be related by their implicit causal connections.

THE REPRESENTATION OF TEXT

We began, therefore, to work on the problem of representing text. This was, after all, the major issue all along. We were not particularly interested in isolated sentences out of context. Dealing only with isolated sentences was probably the root of many of the problems involved with the theories proposed by transformationalists and computational linguists. People do not understand sentences in a null context. Why then did our theories try to deal with sentences out of context? The answer was obviously that this was thought to be a simplification that would facilitate research. But the problem was really significantly changed by this supposed simplification. Certainly parsing sentences in context is a more reasonable problem with respect to word sense disambiguation than is parsing out of context.

We had never dealt with texts of more than one sentence before because we just did not know how to represent them. Now, with the idea of causal chains, we could tie together texts in terms of their causal relations. Such causal chaining, when applied to real texts (Schank, 1975), helped to explain certain memory results (particularly those of Bartlett, 1932). Now we had a theory that said that a crucial piece of information had many causal connections and an irrelevant piece of information had no causal consequences.

The work on causal connectedness gave us a theory that was helpful in explaining problems of forgetting and remembering, and also helped tie together text. However, it could not explain how to tie together texts whose parts were not relatable by chains of results and enablements; something else was needed for those situations.

The something else was obvious once we thought about it. The answer was scripts. That is, scripts are really just prepackaged sequences of causal chains. Some causal chains are used so often that we do not spell out enough of their details for an understander to make the connections directly. Scripts are a kind of key to connecting events together that do not connect by their superficial features but rather by the remembrance of their having been connected before. The prototypical script we chose to examine described what goes on in a restaurant. In a story involving the setting of a restaurant, we cannot infer the causal connection to either ordering or paying from hearing simply that someone has entered a restaurant. However, speakers assume that you know this connection and they do not bother to mention it. There is a causal chain there, but inferring it bit by bit is impossible, which makes scripts necessary.

HIGHER LEVEL KNOWLEDGE STRUCTURES

We set about testing our assumptions about how scripts would facilitate the processing of connected text by building SAM (Script Applier Mechanism, described in Cullingford, 1978). While we worked on SAM we began to wonder about where scripts came from. In thinking about this we came up with the idea that plans gave rise to scripts and that goals gave rise to plans (Schank & Abelson, 1975). Meehan (1976) began to develop a story generator that served as a vehicle for developing our ideas about plans and goals. Wilensky (1978) developed a program to understand stories by tracking goals and plans. All this work is adequately described in Schank and Abelson (1977), so we will not deal with it here.

OUR PRESENT VIEW

The last four years have found us developing the system of plans, goals, themes, and scripts for use in understanding systems. This work produced many working systems (Carbonell, 1979; Cullingford, 1978; DeJong, 1979; Wilensky, 1978) and has greatly broadened our ideas about inference. We now believe the following:

There are a great many possible levels of description. Each of these levels is characterized by its own system of primitives and conceptual relationships. [For example, we have recently introduced a set of "basic social acts" (Schank & Carbonell, 1979) to account for actions that have societal consequences] inferences occur at each of these levels. Thus, for every set of primitives there exists a set of inferences that applies to it. Some of these levels have been described in Schank and Abelson (1977) and will not be dealt with in any detail here. We currently make use of the following kinds of inferences.

Micro-CD

All events in a story can be connected at a level where each event is connected to the events that follow from it, and to the states which enable it. This produces a very detailed causal chain made up of the events and states that were actually mentioned in the text as well as those that had to be inferred in order to complete the chain. The causal chain made by the low level expression of facts is one part of understanding. Thus, in order to read a magazine, you must: ATRANS it; OPEN it; ATTEND to it; and MTRANS from it. When any one of these events is discerned, the others must be inferred.

Macro-CD

Another type of causal chain exists at the macro-CD level. There, events connect to other states and events in the same way as they did at the micro-CD level, but the level of description is different. Thus, going to Boston enables eating in a Boston restaurant at the macro-CD level; but, at the micro-CD level, the locations would have to be further specified, such that going to Boston results in being in Boston which enables beginning to look for and go to a restaurant. This latter level of description can regress in infinite detail where, for example, walking is enabled by putting one foot in front of the other. The level of detail of inferences is extremely important and is dependent on the purposes the understander has in mind.

In the magazine situation mentioned above, Micro-CD is concerned with opening the magazine, holding it, turning the pages, etc. Each of those ACTs also uses causal chains but at a much more detailed level. Macro-CD simply involves having a magazine, which enables reading it. Neither one of these levels of description is more correct than the other.

For causal chaining, then, the needed inference types are:

What Enables
What Results
What are Reasons
What Initiates

These apply at both the macro level and the micro level.

Filling in Missing Information

For every object and person we hear about we are always tracking where they are, the state they are in, what they know and believe, and how they feel. All these inferences are possibly appropriate at any given time. Thus, other kinds of inference types that are necessary are:

Locational specifications
Object specifications
Emotional specifications
Belief specifications

Scripts

Inferring the presence of scripts and the unstated parts of scripts is an important part of the understanding process. The following kinds of inference are significant:

Filling in missing causal chains in a script
Inferring what script is being used
Inferring what unstated script was used instrumentally

Thus, when we hear that "John robbed the liquor store," it is appropriate to ask how he got there, how he got in, where he got his weapon, and so on. Such inquiries are a part of the inference process, since it is only by knowing what we do not know that we can seek to infer it.

One of the main problems with regard to inferences about scripts is the question of why a script is being pursued. This leads to the problem of inferring plans.

Plans

For any given event, it is often important to know the motivations and intentions of the actors in that event. This means knowing the plans being pursued by an actor. Thus it is necessary to make the following kinds of inferences:

Inferring the plan being used
Why was a particular plan chosen?
Inferring facts about an actor given
his choice of plans
Inferring other plans an actor is likely
to pursue to get his goal
Predictive inferences about future plans
What goal is he operating under?

This last inference leads to another class of information that spawns new inferences.

Goals

Detecting the presence of a goal causes the following goal-based inferences to be made:

Why was this goal chosen?
What might it be in conflict with?
Can it be subsumed?
Given this goal, what other goals can we infer?
Under what circumstances will it be abandoned?

Actually these inference types represent only the tip of the numerous kinds of goal-based inferences that have been isolated by Wilensky (1978) and Carbonell (1978).

Since goals are dominated by higher level structures which we call themes, detecting what theme is present and making the appropriate inferences is also necessary.

Themes

The theme-based inferences include finding out:

What kinds of goals is an actor likely to pursue?
What themes are likely to coexist with the given one?
Are there any conflicts in themes?
How might theme conflicts that are detected be resolved?
Where did a given theme come from?

HOW INFERENCES LOOK TO US NOW

Our current work has led us to believe there are, in general, six kinds of inferences. These inference types apply at all the levels of analysis we have worked on so far (i.e., scripts, goals, plans, themes, and some others):

1. *Specification:* Given a piece of an event, what else can be specified about the rest of the pieces?

2. *Motivation:* Why did an event happen? Why this event and not another? What did the actor believe he was doing?
3. *Enablement:* What was necessary for the event to occur?
4. *Results:* What are the results or effects of this event?
5. *Structure:* What higher level structure does this fit into?
6. *Other Events:* What other events are known to occur with this event? What could not have happened if this event happened?

Scripts, plans, and so on fit in as events in the above description. Thus we can ask for *Specification, Motivation, Enablement, Results, Structure* and *Other Events* for a script, a plan, a goal, a theme, or probably any other higher level structure we are likely to invent.

Inference, then, is the fitting of new information into a context that explains it and predicts other facts that follow from it. Since these explanations can occur at many levels, inference is a very complex problem and one we expect to continue working on in an attempt to find out how people understand and how computers could understand.

THE CURRENT SITUATION

Our work started out as a linguistic theory, albeit one with a computer-based bias. Some linguists have explicitly rejected it as a possible linguistic theory (see, for example, Dresher & Hornstein, 1976). In one sense they are right. The phenomena we have become interested in over the years are not particularly phenomena of language per se; rather they are phenomena having to do with the processing of language in general and the issue of the representation of knowledge in particular.

At the same time that *our* work was going on, the field of Artificial Intelligence had been evolving too. When I first arrived at the Stanford AI lab, the major issues in AI were theorem proving, game playing, and vision. Natural language was not considered to be a serious part of AI until Winograd (1972) presented the AI community with SHRDLU. This work contributed substantially to the evolution of AI. The major concern of AI would now seem to be the issue of the representation of knowledge, which of course makes the work in natural language processing quite central.

In the future I expect that many of the relevant fields will begin to become less separate. AI must come to terms with the fact that it is concerned with many issues that are also of interest to philosophers. I hope that the cooperation here will be of more use than was the head-butting that has gone on between AI people and linguists. (Recently this too has changed, however, as the more liberal forces in linguistics have become both stronger and more interested in AI.) Also, the interaction between psychologists and AI people hopefully will continue to flourish. The works of Bower, Black, and Turner (1979) and Smith, Adams, and

Schorr (1978) have already served to bolster the relationship between our group and cognitive psychology.

We now turn to where we are today. The mood in psychology has changed considerably since our early work. Psychologists have made various attempts to test experimentally some of the ideas that we have developed. As Cognitive Science develops, researchers whose original interests were in AI will have to take account of experimental results in their programs if they believe they are developing cognitive models. Of course, not all experiments necessarily reveal God's truth, but some undoubtedly will produce results that should cause Cognitive Scientists with computational orientation to alter their theories and thus their programs.

Our work on scripts has caused many people to use such notions for both programs and experiments. One piece of work in psychology that relates to scripts is that of Bower, Black, and Turner (1979). In addition to showing that script-like considerations are relevant in story understanding, one of the most valuable things to come out of that work was a problem it presented to us. Recognition confusions were found by Bower et al. to occur between stories about visits to the dentist and visits to the doctor. In no intuitive sense can this result be called surprising, since most people have experienced such confusions. But what accounts for it? Should we posit a "visit to a health care professional" script to explain it? Such a script is beyond our initial conception of what a script was, because it was not specific enough. We had always believed that scripts were rooted in actual experiences rather than in abstractions and generalizations from experiences.

The right question to ask at this point is: What phenomena are scripts supposed to explain? Previously we had used scripts, plans, etc., as data structures in terms of which we could make the right inferences to create connected causal chains. But we also always believed that scripts were more than just useful data structures. Scripts ought to tell us something about memory as well as processing. In Schank and Abelson (1977) we claimed that final memory representations for stories involving scripts would use the packaged scripts as the basis for those representations. For example, we would remember the RESTAURANT script (denoted $RESTAURANT) only and could "recall" INGEST by recognizing that INGESTING was a normal occurrence in $RESTAURANT. This is easily accomplished by saving the particular values of variables assigned to each script in a story. Under this view we remember only the salient new information and do not pay attention to the old stereotyped information. $RESTAURANT (lobster, John, Lundy's) should be enough to regenerate a rather dull story.

Our problem here, however, is not the final form of the story, but the initial form and level of the information that we use in understanding the story in the first place. If we used $DENTIST to interpret a relevant story, why should the remembrance of the story get confused with one that used $DOCTOR? If we used $HEALTHCAREVISIT, are we saying that there is no possibility of confusing a dentist story with a visit to an account's office? If we use $OFFICEVISIT, what kind of entity is that? Do we really have information stored at that level to help us understand stories? If we do, then understanding such a story becomes much more complex than we had initially imagined. We cannot get away with simply applying scripts. Rather we will have to consult many levels of information at once.

Why would we store new inputs about dentists in terms of a structure which might confuse it with a visit to an accountants? It seems unreasonable on the surface unless we simply do not have a dentist script available at all. Is it possible that there is no dentist script?

Why haven't we run headlong into this problem before? The answer is, I think, that whereas psychologists worry about recognition confusions in due course, as a part of their natural interest in memory, people working in AI never really concern themselves with memory at all. We have not been in the habit of actually remembering very much at all in our programs, so the issue has not really come up. Once the issue has been raised, however, it seems obvious that what we posit as a processing structure is likely to be a memory structure as well, and this has profound implications for what we do.

LEVELS OF MEMORY

The problem that we must deal with is the question of the kinds of knowledge available to an understander. Every theory of processing must also be a theory of memory. To put this another way, if psychologists show that recognition confusions occur between two entities in memory, this would have to be taken as evidence against a theory that said those two entities existed and were proceeded entirely separately.

Thus, in order to address the question of what kinds of processing structures people have, we should investigate the kinds of things people are capable of remembering (and confusing). We have to become a special kind of psychologist—one who has available to him computer and thought experiments, in addition to more standard methods.

To begin our discussion about memory, it seems clear that there are many types of memory. The first we shall discuss is Event Memory.

Event Memory (EM)

One thing that people remember is a particular experience, often in some detail. So, we postulate a level of memory that contains specific remembrances of particular situations—*Event Memory*. Examples of Event Memory include all the details of "going to Dr. Smith's dental office last Tuesday and getting your tooth pulled" and "forgetting your dental appointment and having them call you up and charge you for it." Events are remembered as they happened, but not for long. After a while, the less salient aspects of an event fade away (e.g., where you got the phone call or why you forgot your appointment). What is left are Generalized Events plus the unusual or interesting parts of the original event from Event Memory.

Generalized Event Memory (GEM)

A *Generalized Event* is a collocation of events whose common features have been abstracted. This is where general information about situations that have been experienced numerous times is held. Particular experiences are initially a part of Event Memory. However, when such particular experiences refer to a common generalized event, that generalized event is brought in to help in the processing of the new input. Once the connection between an event and the generalized event that it references is established, the event itself is liable to gradually fade away leaving only the pointer to the generalized event and the salient features of the event not dominated by the generalized event.

Situational Memory (SM)

Memory for generalized events relies in a similar way upon what we shall call *Situational Memory*. Situational Memory contains information about specific situations in general. Thus while information about dentists resides in GEM, Situational Memory contains more general information. "Going to a health professional's office" or "getting a health problem taken care of" rely on knowledge about such instances in general. SM contains the kind of knowledge we have about waiting rooms and other things that doctors and dentists share.

In the understanding process, information found in Situational Memory is used to provide the overall context for a situation. When we go to a dentist's office and something happens there (e.g., you are overcharged), the specifics of the dental part of the experience are unimportant in the same way that what telephone you were using is unimportant in the event given as an example for Event Memory above. Situational Memory serves as a repository for relevant contextual knowledge as well as the final storage place for the relevant parts of new events in memory; thus it contains relevant contexts and the rules and standard experiences associated with a given situation in general.

Intentional Memory (IM)

The next level of memory experience is *Intentional Memory*. Experiences are encoded in Intentional Memory in terms of the generalizations relevant behind the information encoded in Situational Memory. Information encoded in Intentional Memory would include that relevant to "getting any problem taken care of by a societal organization." What resides here are the rules for getting people to do things for you and other plan-like information. But the decomposition can go on as before. Thus, specific events would lose the particulars that were best encoded at other levels on their way up to the Intentional level.

People often cannot recall the full details of a situation they are trying to remember. Often they can recall just their goals and the resolution of those goals. The specifics of the situation are often more elusive to recall. This suggests that events can be decomposed into the pieces having to do with their intentional basis and these intentions can then serve as the organizational focus where the relevant parts of such experiences can be found.

THE PLACE FOR SCRIPTS IN THE ORGANIZATION OF MEMORY

Where do scripts fit into this partitioning of memory? In particular, what is the dentist script and where can it be found in memory? The answer is that there is no dentist script in memory at all, at least not in the form of a list of events of the kind we have previously postulated. A more reasonable organization of memory would allow for the following kinds of information:

EM — Particular dental visits are stored in event memory (EM). These visits decay over time and thus are not likely to last in EM for a very long time. Rather, what will remain are particularly unusual, important, painful, or otherwise notable visits or parts of visits. These particulars are stored at the EM level.

GEM — At the level of General Event Memory (GEM), we find the information we have learned about dental visits in general that is applicable only to dental visits. Thus, "sitting in the waiting room" is not stored at the GEM level. The reason it is not stored at that level is clear; the lack of economy of storage would be fearsome. We know a great deal about office waiting rooms that has little to do with whether or not they were part of a dentist's office. In addition, any abstraction and generalization mechanism that we posit is likely to be so powerful that it would not be likely to stop operating at any given level. Thus if commonalities between DENTIST and DOCTOR are brought to its attention, it would *naturally* produce this result.

What is particular to a dentist's office is, perhaps, the X-ray machine, or the dental chair, or the kind of light that is present, and so on. These items are not scriptal in nature. Rather, they are just pieces of information about dental offices that are stored as part of what we know about them. For example, one might expect to find a giant toothbrush in a dentist's office. Such information is stored at the GEM level. However, it is also available from the EM level in terms of those particular experiences that can be remembered at that level of detail. (Such memories fade fast, however.) That is, to answer questions about dental offices, there is nothing to prevent us from consulting our knowledge of dental offices in general (GEM) or of particular prior experiences (EM) to the extent that they still can be found.

So where is the dentist script? So far it has not surfaced. The next two levels complete the framework for allowing *dynamic creation of the pieces of the dental script that are applicable in a given situation for use on demand*. The dentist script itself does not actually exist in memory in one precompiled chunk. Rather, it, or more likely its needed subparts, can be constructed as needed. The economy of such a scheme is very important. Moreover, the memory use and probable psychological sensibility of such a solution is highly significant. Now we will consider how that might work.

SM In Situational Memory (SM) resides information about a situation in general. Here is where we find the kind of knowledge that would include facts such as "nurses wear white uniforms," "doctors frequently have many rooms so that they can work on lots of patients at once," "there are history charts that must be selected and updated by women in white outfits who might not actually be nurses," etc. We also find information about situations in general. This includes information such as the flow of events in an office, for example. Thus, the bare bones of the dentist script and, most importantly, many other scripts are found in SM. Here we have information such as: "If you need help you can go to the office of a professional who gives that help. You may have to wait for that help for a while in a waiting room. You may report your problem to an underling of the professional's. You will get a bill for services, etc."

IM Intentional Memory contains more goal-based memories. Trips, romances, improving one's health, and other general contexts whose immediate goals are known are IM structures. Intentional Memories organize inputs according to their reason for existence. As a consequence of this, memory confusions at the IM level involve different situations whose intentions are the same.

According to this view of the information in memory, then, scripts do not exist as extant memory structures. Script-like structures (corresponding to what we have called scenes or even parts of scenes) are constructed from higher-level general structures *as needed* by consulting rules about the particular situation from the three other levels.

The words "as needed" are very important. Why bring an entire script in while processing if it will not be used? Since scripts are being constructed rather than being pulled in whole from memory, only the parts that there is reason to believe will be used (based upon the input text) need to be brought in.

As for retrieval and storage of the incoming information, new stories are available at the EM level for only a very short time. In the course of processing an initial input, pointers are created that preclude the necessity of storing all the details. These pointers are not to the dentist script, but to the relevant subscenes that are to be found at the various memory levels. Thus, something that happens in the waiting room is stored with a pointer to the waiting room scene. However, and this is the main point, the waiting room scene came from knowledge about waiting rooms that was only picked up for the occasion from the highest level (IM). Thus, it was not connected to any dentist script initially, and whatever has happened in the waiting room, unless it was of particular interest, will be stored at the IM level, virtually disassociated from the dentist sequence. Under this scheme, recognition confusions will occur among various waiting room scenes with regard to the question of the original overall situation of which they were a part. The only time when it will be clearly remembered which waiting room scene belongs to which story will be when the continuity is provided between scenes by the story itself. For example, if something special happens in the waiting room scene that affects later scenes, the connection would be a causal link and such connections should be more easily remembered.

MEMORY FOR DENTIST INFORMATION

To see how what we have outlined would work, it is perhaps useful to look at a diagram of the structure of memory for a story (shown at the EM level) involving a dentist:

IM HEALTHPROBLEM
    FIND PROFESSIONAL + MAKE CONTRACT
    + PROFOFFICEVISIT

SM GO TO OFFICE + WAITING ROOM + ENTER INNER OFFICE
+ HELP + LEAVE + BILLSENT

GEM Dentist visits include:
  getting teeth cleaned—dentist puts
    funny tooth paste on teeth
    turns on machine
    etc.
  getting teeth drilled
    D does x-ray
    D gives shot of novocain
    D drills
    etc.
  also: Dentists fill the health care professional role in
    HEALTHCAREVISIT

EM The time I went to the dentist last week:
  I drove to the dentist.
  I read Newsweek. There were holes in all the pictures.
  I entered.
  He cleaned my teeth.
  He poked me in the eye with his drill.
I yelled at him.
  He didn't charge me.

The events in EM are remembered in terms of the highest level memory structures that were activated.

After some time, decay sets in and allows the magazine reading to be stored as part of the WAITING ROOM scene of PROFOFFVISIT. It thus gets disassociated from the rest of the event. Similarly, the "eye poking" gets stored under HELP and is thus disconnected from the magazine experience. But, since HELP is filled by specific Dentist information from GEM, it is remembered as part of a Dentist experience whereas the magazine experience can get completely confused with any other situation in which one might read a magazine.

The main point is that memory breaks down its new information into appropriately interesting pieces and stores those pieces in the context to which they are relevant, i.e., the context which originally recognized them and explained them.

MEMORY ORGANIZATION PACKETS (MOPs)

What we have been addressing here is the overall question of how information is organized in memory. The old issue of semantic memory versus episodic memory and the newer issue of what kinds of memory structures are available are the key elements with respect to memory organization. To summarize so far: we are saying that scripts are not data structures that are available in one piece in some part of memory. Rather, script application is a reconstructive process. We build pieces of scripts as we need them from our store of knowledge to help us interpret what we hear. Thus the next key question is: What is the organization of the knowledge store? This is another way of asking the question: What kinds of knowledge do we have and how is that knowledge represented and used in the understanding process?

There are thus two relevant questions to ask of memory: First, how does any given experience get stored so that it will provide a capability for understanding new experiences in terms of it? And second, why do recognition confusions occur at all?

Situational level memory structures help us understand the experiences we have. They do this by generating parts of scripts. But, in addition, they tend to enable recognition confusions in two ways: First, since new events are understood by using these structures to interpret them, a connection is established between the new event as entered in EM and the memory structure that was used to interpret that event. This connection is established by means of two different kinds of pointers. The first, a *processing pointer*, connects the memory structure with the new event in order to help in the processing of that event. The second, a *memory pointer*, is established because the memory structure is itself affected by the new event. We call memory structures at the SM level Memory Organization Packets or MOPs. MOPs both organize episodic memories and help to process new inputs; that is, they are the means by which an appropriate episode in memory can be accessed for aid in processing a new input. At the basis of a MOP at the SM level is an abstraction of a mass of input events that have been mushed together in memory. A MOP is a collocation of all the events that have come to be stored under it. Thus, memory pointers must be established from a relevant MOP to the detailed event which that MOP is helping to process at the EM level.

Of these two pointers, only the memory pointer needs to last very long. When the processing is finished, the processing pointer is easily forgotten since it can always be regenerated as needed. The memory pointer stays but decays over time. Details that are insignificant in the EM event are forgotten. Significant or interesting details are remembered by virtue of there being more than one memory pointer available; that is, if more than one Situational or Intentional structure has been accessed, then more than one pointer has been established. The number of the possibly relevant structures at these levels may be high since a new event may call many kinds of structures in to help interpret it during processing. For every structure that is called in during processing, a memory pointer is established. The combination of what these pointers point to is what remains of the event at the EM level of memory.

Thus, since each new piece of information is stored in terms of the high level structure that was needed to interpret it, two kinds of confusions occur. Connections between items in the same episode that are interpreted by different high-level structures will tend to break down. A waiting room scene will tend to disconnect from the dentist script of which it was a part because it was interpreted by a different MOP (one having to do with office visits perhaps) than other parts of the story.

The second kind of confusion will occur within a script. When a high level structure is deemed relevant, all inputs are interpreted in terms of the norm. This causes small details not normally part of a script to get lost and normalized. Normalization does not occur for very interesting or weird deviations from a script. The reason for this has to do with the answer to the first question above.

REMINDING

Sometimes during the processing of new inputs an interesting phenomenon occurs: You are reminded of a previous experience that is somehow similar to the new input currently being processed. Such reminding experiences are not random. Rather they are dependent upon the very nature of the understanding and memory processes we have outlined.

The answer to the question of why one experience reminds you of another is of primary importance to any theory of human understanding and memory. If people are reminded of things during the natural course of a conversation, or while reading, or when seeing something, then this tells us something of great importance about the understanding process. It tells us that a particular memory piece—that is, a specific memory—has been excited or "seen" during the natural course of processing the new input. We can then ask two important questions:

1. Why did processing naturally pass through this piece of memory? That is, what is there about the processing of new information that requires a particular related piece of information to be noticed?

2. How did such a mechanism as reminding develop? That is, what is the purpose of reminding? Why is this phenomenon available consciously when so many other processing phenomena are not?

We can begin to attempt to answer these questions by considering the kinds of reminding experiences that people have. For example, there is a restaurant in Boston where you pay first, then eat, called Legal Seafood. Going to another such restaurant, and saying, "This restaurant reminds me of Legal Seafood," would of course be quite natural. According to our view of memory, the restaurant script is merely a first approximation of where search should begin for the most appropriate memory structure to be used in processing a new input. Thus, initial access of the restaurant script merely serves to begin our search for the high-level structure that will be used to understand this new experience. Accessing the restaurant script just means finding a relevant entry point to memory. We have, rather than a discrete set of such high-level structures, a potentially infinite set. There is not one restaurant script but thousands. The various refinements on restaurants all serve as nodes in memory that help to reconstruct a needed high level structure. By saying to ourselves, "Gee, you pay first here," we have caused our minds to traverse a particular path within the information organized by restaurants in order to complete our search for the highest level structure (i.e., the structure that explains the most information). At the end of that path is Legal Seafood, so reminding occurs.

More important than reminding, however, is that all the predictions from that previous experience are now available to help interpret the new input. Such predictions function no differently if the new input calls up a once-seen prior relevant experience or a multitude of experiences expressed in terms of high-level generalizations such as "the restaurant script."

The logical consequence of all this is that there is a potentially infinite set of such structures and that most people's sets would be extremely large and idiosyncratic. For example, an expert at chess would be able to recognize "famous games" or positions that have been seen before. Such recognition depends on the use of a high-level structure in the first place of the kind we have been discussing that would have been part of the understanding process. That is, during understanding, we are constantly seeking the highest level of analysis we can get. This works for understanding chess as well as for anything else. Former chess understandings are stored at particular subparts of the appropriate knowledge structures.

UNDERSTANDING

We can now reevaluate what it means to understand. When we enter Burger King, having before been to McDonald's but never having been to Burger King, we are confronted with a new situation which we must attempt to "understand."

We can say that a person has understood such an experience (i.e., he understands Burger King in the sense of being able to operate in it) when he says, "Oh, I see, Burger King is just like McDonald's."

To put this another way, we might expect that at some point during his Burger King trip he might be "reminded" of McDonald's. The point I want to make is that understanding means being reminded of the closest prior experienced phenomenon and being able to use the expectations generated by that reminding. When we are reminded of some event or experience in the course of undergoing a different experience, this reminding behavior is not random. We are reminded of this experience because the structures we are using to process this new experience are the same structures we are using to organize memory. Thus, we cannot help but pass through the old memories while processing a new input. There are an extremely large number of such high level memory structures. Finding the right one of these, (that is, the one that is most relevant to the experience at hand) is what we mean by understanding.

Is it any wonder that we are reminded of similar events? Since memory and processing structures are the same, sitting right at the very spot needed will be the experience most like the current one.

But all experiences are not identical to all others. A key issue then is the creation of new structures. This is done in terms of the old ones. To return to our fast food example, when Burger King reminds you of McDonald's, what you are doing goes as follows: "Ah yes, Burger King is just like McDonald's except the waitresses wear red and yellow and you can have it your way." A new discrimination on the net that contains McDonald's is then made, creating a node in which Burger King is a high-level structure that shares most, but not all, of its properties with the old McDonald's node. The differences are significant in that they themselves may form the basis of reminding experiences.

In this view, then, understanding is finding the closest higher-level structure available to explain an input and creating a new memory node for that input that is IN TERMS OF the old node's closely related higher-level structure. Understanding is a process that has its basis in memory, particularly memory for closely related experiences accessible through reminding and expressible through analogy.

MEMORY DISCRIMINATIONS

Now the question is: How do we go about finding what is stored in memory? If there are "have it your way" discriminations, in what way could they be used? The answer clearly depends on effective initial categorization of the input.

Memory is highly idiosyncratic. One person's organization is not another's. How people categorize experiences initially is how they remember them later. If Burger King is seen as an instance of McDonald's, it will be stored in terms of whatever discriminations the understander noticed as relevant at that time. However, it is possible for a person to make multiple discriminations as well as multiple categorizations. Thus, a person can see Burger King as "something tasteless that kids love," a "place where red and yellow uniforms are worn," and a "place where you can have it your way." Each of these is used as a path by which Burger King can be accessed. A fight with one's child in a Burger King might be stored solely as an instance of a child fight, or as a fight in a restaurant, or as a fight in a Burger King. If the latter categorization were used, fights with a child in McDonald's might not be noticed. Thus, an intelligent understander stores his experiences as high up and as generally as possible so as to be able to learn from them, i.e., so as to make them available for use as often as possible or in as many situations as possible.

One question often asked of these ideas is: "Why, when you enter a restaurant, are you not reminded of all restaurants, or even reminded of that particular restaurant?" I believe the answer to both questions is that you most certainly are. When you enter Naples (a Yale hangout) you are reminded of Naples. You then use that reminding as the source of predictions about what will happen next; that is, you use the most particular script available to help you process what you are experiencing. When Naples reminds you of Naples, you do not experience the same sensation of reminding for an obvious reason. The more appropriate a reminding experience is, the less it seems like reminding. But reminding is simply the bringing to mind of highly relevant memories to help in the processing of new inputs. To say or to feel upon entering a new restaurant that "this place reminds me of a restaurant" is rather absurd, but remind you it does. If this were not the case, how would you know that it was a restaurant? Thus, reminding is not just a rather interesting phenomenon that I have been seeking to explain. Reminding, in a very serious sense, is the most significant memory phenomenon that there is to explain.

WHAT MEMORY LOOKS LIKE INSIDE

We are now ready to take a look at a specific proposal for handling memory (and handling scripts in particular) in order to account for the issues we have been discussing. For old times' sake, we will again use restaurants. The difference between what we have said in the past and our new view has to do with how restaurant experiences are organized in memory. Consider two restaurant experiences that are virtually identical except for what was ordered. The chance for confusion in memory here is enormous. Which waitress (if there were two different ones in the same restaurant) served which food might be confused, for example. Such confusions are not accounted for by our original conception of scripts where each story, including information about the waitress and the food eaten, is uniquely stored with its own copy of the script.

SCRIPT EMBELLISHMENT

Scripts are formed for actual use by building them up from scenes. However, in building up a script we are also allowing for the possibility of memory experiences being allowed to generalize and be stored at the high-level structure that best explains them. This causes the script that was temporarily built up for processing purposes only to be broken down again, thus causing memory confusions and a certain lack of connectivity between the scenes of a story as it is stored in memory.

Recall that our purpose is to integrate the structures of processing with those of memory. We desire to have episodes stored in such a way that each one of them can serve as a kind of script itself; that is, we want predictions to be available from *all* prior experiences, not just from those we have labelled officially as "scripts" or "script pieces." After all, people make predictions about what will happen next from past experiences. Are scripts the only kind of past experience that aids processing by making predictions and filling in causal chain inferences? Obviously this cannot be. A person who has experienced something only once will expect his second time around to conform to the initial experience and will be "surprised" in some sense every time the second experience does not conform to the first.

This is how scripts get put together in the first place: first one experience, then another on top of it, strengthening those areas of agreement and beginning to solidify a script. But obviously there are times when "new" experiences for which there is one or no prior experience can occur in the middle of an old, well-understood experience. Thus, when you go to Legal Seafood, you modify your restaurant script so as to indicate that the PAYING scene has been placed immediately after the ORDERING scene in memory. What I want to propose is that what is happening here is *not* the creation of a new part or "track" in a script. I see very little evidence for tracks at all. Rather, the entire memory experience is being stored under this PAYING interruption or abnormality following the ORDERING scene.

Two kinds of reminding exercises are accounted for by this. First, any other script readjustment occurring after ORDERING might remind one of Legal Seafood. Second, a new placement of the PAYING scene in the restaurant script might be expected to remind one of Legal Seafood. This reminding would occur as a result of having categorized Legal Seafood as weird with respect to PAYING. This gets placed as a part of what we know about PAYING. So, when PAYING is placed in a new spot, the Legal Seafood experience is brought in because it is hanging off a "PAYING reassignment" discrimination.

Now what does this reminding buy you? The reminding actually causes the whole rest of the reminded experience to be brought into memory at this point just as would happen with any new experience not accounted for by a welltrodden script. This experience is now used, just as any script is used, to predict what will happen next.

Thus, deep down inside the guts of a script, we find all the pointers to every specific memory experience we have had that has been organized in terms of that script and that has not been obliterated by multiple identical experiences. Thus script application is embellished by going down paths which the script itself organizes, that contain all prior deviant (that is, not completely standard) experiences. These experiences are functionally identical to scripts and thus are an integral part of the application process. This can occur within any script-piece at all.

As an example of all this, consider Figure 1, a picture of a possible set of memory experiences tied to, or organized by, the restaurant script.

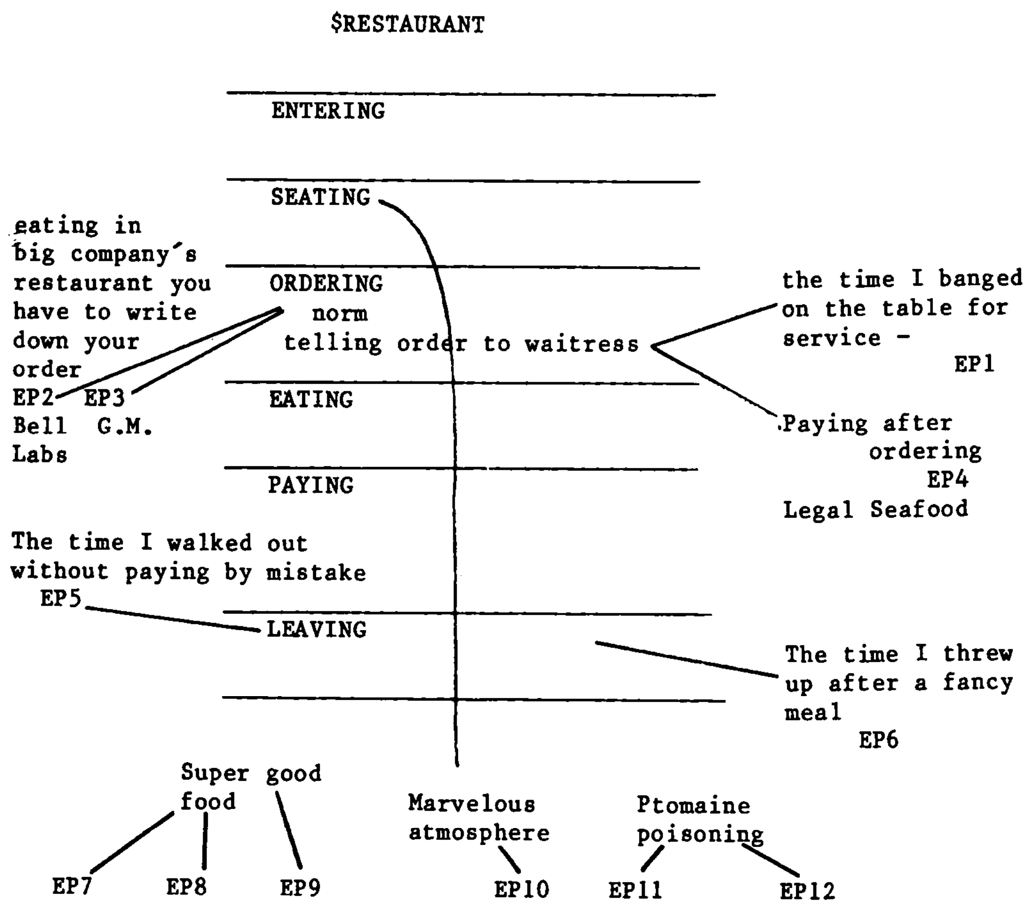


Figure 1.

From this diagram we can see that the ultimate purpose of scripts is as organizers of information in memory. The restaurant script that we have used in the past is no more than the standard default path, or basic organizing principle, that serves as the backbone for all remembered restaurant experiences that have been stored as restaurant experiences. Thus, we are saying that every deviation from the standard script is stored as a modification of the particular scene in which the deviation occurred.

So, one experience in Legal Seafood causes a deviation (these were previously referred to as being on the ''weird list'' in Schank & Abelson, 1977) in the ordering scene. This deviation serves as both the beginning of a reminding experience and the start of the script application process. As we have been saying, storage and processing must be taken care of by the same mechanism in order to get natural reminding to take place. In addition, the scheme that I am proposing allows for the use of all prior experiences in the interpretation of new experiences rather than a reliance on only standard normalized experiences (i.e., what we have previously called scripts).

If the new experience does have a counterpart, i.e., if similar deviations have been met before, *at some point* these experiences are collected together to form a scriptal subscene whose predictions are disembodied from actual episodes like the higher-level script itself. When enough of these are encountered, a new subscene is created that will *not* cause one to be reminded of the particular experiences that caused the creation of this subscene. As this subscene is created, all pointers to the relevant episodes that helped to create that subscene are erased, although other pointers not a part of this subscene but pointing to the same episode would still exist.

Memory collects similar experiences from which to make predictions; these are of general utility above some threshold of a number of prior experiences. Below that threshold, remindings of prior experiences serve as the source of relevant experiences. Thus uniqueness of experience, or, more accurately, unique classifications of experience, serve as a rich source of understandings about new experiences.

In the process of script embellishment via deviant paths we access entire episodes, many of which may have next to nothing to do with the story currently being processed. It is important to be able to separate the relevant from the irrelevant. On the other hand, it is hard not to be reminded of the parts of that experience that are connected to the scene that you have been reminded of. This is one aspect of *intelligence* that comes into play here. The discrimination of those experiences that are relevant for prediction and those that are irrelevant is one of the most formidable tasks facing an understander. *Such discriminations must be done at processing time since one cannot know beforehand where relevant similarities might lie for future inputs.* Thus, a very important part of understanding is the analysis of what is happening to you now with respect to the issue of how relevant newly encountered events might be for predictive purposes to handle future inputs. The collocation of arguments and "paying right after ordering" is not a useful category for experience-based prediction. Clearly we are not born knowing such things. What form such knowledge takes and how we go about acquiring it is, it seems to me, one of the very big issues for the future.

HIGH-LEVEL MEMORY STRUCTURES

The key point in the issue of what is and what is not a script has to do with where we expect information to be found in memory. From the point of view of processing it makes sense to talk about having a restaurant script available. We have shown that such scripts will facilitate the processing of stories; but just because an entity facilitates processing, it does not necessarily follow that that entity exists as a chunk that has been prestored in memory. It is quite plausible that such entities are constructed on demand from information that is stored directly in memory. The key question before us, then, is whether scripts and other high level structures have only a processing role, or whether they are also memory pieces, useful for storage of information that has previously been processed using those high level structures. That is, are high level structures solely processing devices or are they also memory devices?

If the chunks we have been calling scripts are not solely processing devices, the demands on them change. Just as we would not expect that a sensibly organized memory would have the fact that George Washington was the first President stored in fifteen different places, we would not expect that "you eat when you are hungry," or "you order after reading a menu," or "if you don't pay your dentist bill you can be sued" to be stored in fifteen different places either.

Once the requirement that you have to find *one* and *only one* place to store general information comes into play, the question of where that information is stored becomes extremely important. To decide that question, notions such as scripts must be tightened up considerably so that general information shared by any two scripts is held outside them in some other memory store. To do this requires ascertaining what it might mean for two scripts to share the same information, and finding out when such sharing is "realized" by memory and when it is not.

The Creation of MOPs

When a child discovers that its personal restaurant script is also shared by other people, he or she can resort to a new method of storage of restaurant information: a standardized restaurant script with certain personal markings that store idiosyncratic points in view; that is, the child can begin to organize experiences in terms that separate out what is unique and what is shared by the culture. For example, adults know that getting in a car is not part of the restaurant script, but this may be a very salient feature of the child's personal restaurant script. It is very important for the child to separate the car experience from the restaurant experience. The child must learn to reorganize memory according to cultural norms.

This reorganization of stored information can continue indefinitely. New experiences are constantly being reorganized on the basis of similar experiences and cultural norms. The abstraction and generalization process for experientially acquired knowledge is thus a fundamental part of adult understanding. When you go to the dentist for the first time, everything in that experience is stored as one chunk. Repeated experiences with the same dentist, other dentists, and vicarious experiences of others serve to reorganize the original information in terms of what is peculiar to your dentist, yourself in dental offices, dentists in general, and so on. This reorganization process never stops. When similarities between doctors and dentists are seen, a further reorganization can be made in terms of health care professionals. When doctors' and lawyers' similarities are extracted, yet another organization storage point emerges. The key to understanding is the continual creation of *Memory Organization Packets* (MOPs), which record the essential parts of the similarities in experience of different episodes.

The purpose of a MOP is to provide expectations that enable the prediction of future events on the basis of previously encountered, structurally similar events. These predictions can be at any level of generality or specificity. Thus, such predictions can come from nearly identical or quite different contexts or domains, since a context or domain can be described at many different levels of generality. The creation of a suitable MOP provides a class of predictions organized around the common theme of that MOP. The more MOPs that are available for processing a given input, the more prediditons will be available to help in understanding that input. The ability of MOPs to make useful predictions in situations for which there is no direct experience but for which there are relevant analogous experiences is crucial to our ability to understand.

Seen this way, a MOP is a kind of high-level script. The restaurant script is itself a kind of MOP, but it is also related to many different and more general MOPs. There is a MOP about social situations, a MOP about requesting service from people whose profession is that service, and a MOP about business contracts—to name three that are relevant to restaurants.

Viewed as a whole, then, memory is a morass of MOP strands, each connected at the base to the relevant abstractions and generalizations that are the base of the MOP. At the end of each strand are particular experiences (i.e., individual episodes) or groups of experiences (i.e., scripts).

Using MOPs

Consider the information relevant in a visit to a doctor's office. At least five MOPs are relevant to the construction of the processing structures necessary for understanding a doctor's office visit. They are: PROFOFFVISIT; CONTRACT; FIND SERVICE PROFESSIONAL; USE SERVICE; and FIX PROBLEM.

As we will see, these five MOPs overlap quite a bit. There is nothing wrong with that; indeed it should be expected that any memory theory would propose overlapping structures since they are the source of both memory confusions and the making of useful generalizations across domains.

When a script is available it can be used without really looking at a MOP. However, because storage of information needs to be economical, we would not expect what is best stored in a MOP to be found in a script as well. Thus, the doctor script would not have the doctor suing the patient for nonpayment of the bill directly in it. Neither would the bill itself be in the domain of the doctor script. Each of those is best stored as part of a MOP for a CONTRACT, with strands pointing to the doctor script. A doctor visit is perhaps not best viewed as a contract, but it is one nonetheless, and the CONTRACT MOP must help to construct what we can sloppily call the DOCTOR "script" that might actually be useful in processing.

It is important to mention that $DOCTOR is connected to the CONTRACT MOP by a strand of the MOP, but that $DOCTOR does not contain that strand, i.e., it does not contain information about payment other than the presence of that MOP strand. Thus, $DOCTOR is smaller than is obvious at first glance, since we have essentially taken the paying scene out of the script. The actual DOCTOR script that exists in memory contains only the doctor-specific parts of the doctor experience.

Thus, waiting room information is not part of $DOCTOR; waiting rooms are part of PROFOFFICEVISITS, which are also MOPs. But PROFOFFICEVISITS are different from CONTRACTs, which are different from HUNGER and PROFSERVICE. Each of these are MOPs, but they represent different kinds of MOPs. PROFOFFICEVISIT has a strong structure that it imposes on any situation to which it is applied. This structure is in essence a kind of search procedure that helps one sort through the strands in a MOP.

MOPs are memory organization packets. Thus PROFOFFICEVISIT will grab up particular experiences and store them. An experience in a waiting room of an office will get disembodied from the rest of the visit. What happens in the waiting room will be stored with the PROFOFFVISIT MOP, but the actual consultation with the doctor or lawyer will be stored with a different MOP.

WAITINGROOM is a content strand of the PROFOFFVISIT MOP. That is, it has a great deal of information attached to it, such as what a waiting room looks like, what is in it, what happens there, and so on. The HELP strand, on the other hand, is entirely empty. It is a kind of place holder, the only content of which is what is connected temporarily to it on either side. This is where $DOCTOR or $DENTIST comes in. Under this view, scripts are very particular structures about a situation that can fill in an empty strand in a MOP. Actually, these scripts are strands of different MOPs. Thus, just as WAITINGROOM is a contentful strand of the PROFOFFVISIT MOP, so DENTIST is a contentful strand of the HEALTHCARE MOP.

There are also a great many other relevant structures. Some are relevant because MOPs can themselves be strands of other MOPs. There is a *twining* mechanism that can cause strands from many MOPs to fill the same empty strands in another MOP. In the end, then, what we are doing is constructing a DOCTOR superscript (shown at the bottom line of Figure 2). This superscript is constructed for use as needed by taking the strands of relevant MOPs and ordering them by time and enablement conditions. Often multiple strands account for one scene in a superscript. Below, the DELIVER strand of CONTRACT as well as the HELP strand of HEALTHCARE and the SERVICE strand of PROFOFFVISIT all relate to $DOCTOR. That is, they each explain to some extent the role that the doctor is playing.

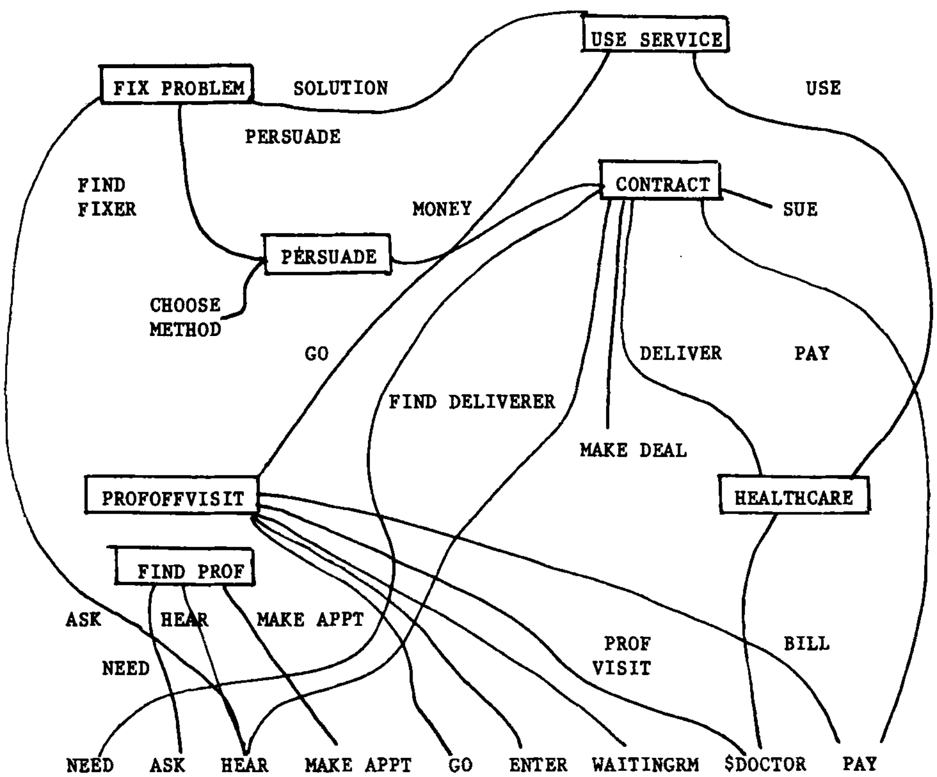


Figure 2.

Taking a look at Figure 2 for the construction of the doctor superscript, we see that a MOP has one clear characteristic: It organizes a class of information by the creation of sequences of slots that can be filled by various structures. In essence, the true differentiation of the kinds of MOPs that exist depends on the types of entities that can fill the slots in the MOP.

Processing Using MOPs

Memory Organization Packets serve as the basis of memory and of processing. In order to understand an input it is necessary to find in memory the structure or particular episode most like the new input. Reminding is one way of telling that such things are happening.

Processing an input means finding a relevant memory piece. When such a piece is found, expectations are created that some from all the pieces to which the initial memory piece is connected. Thus, when we receive an input, we look for a relevant MOP. Upon finding this MOP, we create expectations at every level to which that MOP is naturally connected. Expectations from scripts that have been activated create demands for certain conceptualizations. Expectations are simultaneously generated from the relevant script that filled the slots of the MOPs.

To illustrate how some of this works, consider a story beginning "My drain pipe was overflowing." Now for our purposes the point is not whether this is the first line of the story or not; rather, it is important that this simply be an input to a cognitive system. The fact that you might hear such a thing in everyday conversation is important also. The questions we need to address are:

1. What comes to mind upon hearing such a sentence?
2. What structures are being activated in memory that cause such things to come to mind?
3. What state is the mind in after having received this input?

At Yale in recent years and among researchers in general who are concerned with scripts or schemata, it has seemed plausible to answer these questions with something called the "plumber script." Such a script implies that any body of knowledge can be a script. Clearly a body of knowledge about plumbing can be assembled from the various corners of memory in which it resides in order to create such an entity as a plumber script. One issue is whether such an entity preexists in memory or is constructed, and, if the latter is true, then the real issue is "constructed from what pieces by what method?" A second issue is where are our episodic memories to be found that will help us respond to what we have heard? It seems unlikely that every experience we have had with plumbers is organized by $PLUMBER. Clearly a great many memory structures are likely to be active.

Thus far we have taken the position that to have precompiled chunks of memory such as a plumber script is unrealistic, particularly when we consider facts of memory such as recognition confusions, memory searches, and forgetting based upon the breaking up of an experience into chunks. A great deal of information can be retrieved about plumbers (e.g., what a plumber is likely to wear, the estimated size of the bill, etc.) that is in no sense a part of the script, so it seems safe to say that some reconstruction is going on, or at least that various pieces of memory are being searched when an input is being processed. In our discussion we will assume that there is no plumber script in anything but the simplest of all forms, and that the main problem in responding to an input such as the one above is the accessing of the memory structures relevant to the creation of the plumber superscript.

What kind of high level structures might be relevant to ''My drain pipe is overflowing''? Clearly at least the following information is relevant: Drains must be understood to be part of sinks in houses, thus determining the general location of the item in question. Such information is part of the meaning of ''drain.'' It is unlikely that there would be a ''drain MOP'' available with that information in it. The existence of such a MOP implies that memories about drains are organized together in one place. This seems unlikely; however, there is nothing immutable about what can be a MOP. Different individuals with different levels of expertise are likely to have different needs in memory organization.

''Drain'' points to information about bathrooms and kitchens, etc. Such information is stored in what Minsky (1975) refers to as a ''room frame.'' These frames contain primarily visual information rather than episodic information (although again the latter is possible). The visual information attached to the room frame here is helpful for understanding future inputs such as: ''To fix it I sat on the toilet,'' or ''The overflow ruined $20 worth of cleanser stored underneath.'' Such statements would be quite impossible to understand without these active and ready frames. But such frames are not MOPs, they are just, upon occasion, used by MOPs.

One way they are used by MOPs here is that, since these rooms are parts of houses, the combination of the implicit house and the possessive ''my'' causes information about HOMEOWNERSHIP to be activated. HOMEOWNERSHIP has information in it derived from preservation goals (P-GOALS; see Schank & Abelson, 1977) and, among other things, points to the FIX-PROBLEM MOP.

Of course people have drains in places they rent as well. This possibility is activated in the absence of knowledge to the contrary by activating D-AGENCY (Schank & Abelson, 1977). D-AGENCY points to knowledge about AGENCY relationships (which is what ''landlords'' are) so that HOMEOWNERSHIP can still be used although it would be mediated by D-AGENCY.

Until we see ''overflowing'' we do not really know what is being told to us. But, after we have seen it, a great many structures must become active. First, the conceptualization that is to be constructed here contains empty CD slots for what object is being PROPEL-ed (which comes from ''overflow'') and to where. The OBJECT defaults to ''water'' by consulting the ''normal contents'' part of the conceptual dictionary item for ''drain.'' The ''TO'' slot is filled by consulting the relevant frames, in this case the candidates being ''in house,'' ''on floor,'' or ''on carpet.''

The activation of FIXPROBLEM causes an attempt at the creation of a solution. FIXPROBLEM has as its strands FINDFIXER, PERSUADE, and SOLUTION. Each of these, it turns out, is a possible topic of conversation where the first input is our above sentence. Thus we might hear:

1. I know a good plumber.

2. My, that's going to cost a lot of money.
3. Have you tried Drain-Fixing Drano?

The fact that all these are quite possible as responses here is an important indication that all these structures are active in the mind of the understander of these sentences. To test further the validity of the active high-level structures in this way, consider other possible responses based on the other ones given above:

1. That sink of yours has been rotten for ages.
2. I told you not to get such an old house.
3. Boy, isn't it a pain to own a house?

Each of these is perfectly plausible as a response. We attribute this to the fact that some high-level memory structure would have to have been activated by the input. What other kinds of statements might be acceptable here? Some candidates are:

4. Oh, isn't that awful.
5. Did you have to stay home from work?
6. Water can be awfully damaging.
7. Do you know how to fix it?
8. Would you like to borrow my Stilson wrench?
9. And with your mother coming to visit, too!
10. This has certainly been a bad month for you, hasn't it?

Assuming that these too are all legitimate, what structures do they come from?

The ultimate questions here are what kinds of MOPs are there and how many of them are likely to be active at any given time? It seems plausible that the following high-level structures are likely to be active during the processing of our input sentence:

JOB; FAMILY RELATIONS
HOMEOWNERSHIP

FIX PROBLEM; PERSUADE
USE SERVICE

PROF HOME VISIT
FAMILY VISIT TO HOME
FIND PROFESSIONAL
MAKE CONTRACT

$PLUMBER

Are all these structures MOPs? Returning to our definition of a MOP, we can see that some of them clearly are and some of them fall into a rather grey area. Recall that a MOP is an organizer of information that can be used to create a superscript. Recall further that MOPs serve to organize terminal scenes that have within them a backbone of a sequence of events that are episodes from memory organized in terms of that backbone. These terminal scenes are either script-like (in which case they contain deviations from the normal flow of the script encoded as actual memories) or else they are locative in nature (in which case they contain actual episodes that are organized in a nonevent-based manner, possibly visually). Thus, a MOP is an organizer of terminal scenes or actual memory episodes. By this analysis, PROF HOME VISIT, FAMILY VISIT TO HOME, FIND PROFESSIONAL, and MAKE CONTRACT are all MOPs. They each organize terminal scenes such as PAY, PHONING FOR AN APPOINTMENT, FAMILY DINNER and so on. As we have said these terminal scenes are whether actual memories are to be found.

FIX PROBLEM, PERSUADE, and USE SERVICE are meta-MOPs; that is, they do not have memories in them directly. Rather, they are structures that serve to organize MOPs.

This leaves us with JOB, FAMILY RELATIONS, and HOMEOWNERSHIP as active knowledge structures that do not fit in with our previously established definition of MOPs and the structures that both organize and are organized by MOPs. What then are these structures and how do they differ from MOPs?

The first thing to notice about these labels is that we have a great deal of information about them in our memories. In fact, we have so much information and it is of such great importance to us (i.e., it concerns our high-level goals) that to begin to think that we can break such structures down into MOPs that organize terminal scenes is absurd. There is, for example, a JOB MOP that contains scenes about applying for jobs, getting paid, terminating employment, and so on. But there is a great deal more information about one's job or knowledge of jobs in general that could not be neatly contained in the JOB MOP. The point here is that such information is at a higher level than that of MOPs. We cannot begin to talk about that information here, since it is extremely complex, but later on we shall have a bit more to say about the role of the structures that are at a higher level than MOPs.

The MOPs that we have specified and the other high-level structures that we have not specified relate for this story as shown in Figure 3.

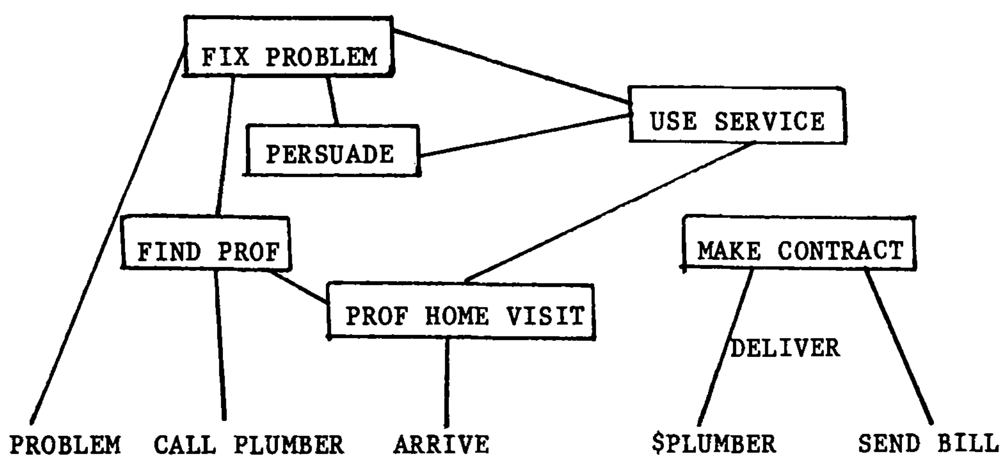


Figure 3.

All of the MOPs mentioned here are capable of making useful predictions about what is going on in this situation. The big questions, then, are exactly when such predictions are made, how they are called into play in processing, and where they come from.

MOPs REVIEWED

A MOP is a packet of knowledge that can be called into play for initial processing and memory storage. Thus, a MOP is a bundle of memories organized around a particular subject that can be brought in to aid in the processing of new inputs. All the subjects that we have considered so far have been events. Thus, MOPs are used insofar as we have described them for understanding and storing event-based information. The criteria we have been using for determining what can be a MOP has depended on the following questions:

1. Why is the information contained in that MOP contained there and not anywhere else?
2. How can we search that MOP?
3. What is the output (i.e., what is at the end of the strands) of that MOP?
4. How is that MOP known to be relevant and subsequently accessed?
5. What kind of processing help (i.e., what predictions are made) is available from having accessed that MOP?

First Conclusion

There are several technical conclusions we can make from what we have said here. In the next section I shall make some more general conclusions.

First, we are now in a position to see what a script really is. Scripts are a particular kind of MOP which we might call subMOPs. They are subject to temporal precedence search, produce Conceptual Dependencies, and contain memories—so they are obviously MOPs. But they are very particular. MOPs tend to organize information in general about an area and thus at one level of suborganization are methods of filling the various strands of the MOP. Scripts are standardized memory chunks that are particular methods of filling one or more strands in a MOP.

A second technical issue revolves around the kind of thing that is going on with respect to high-level structures in memory. We can see that there are basically three overall kinds of memory events: those that are classified uniquely, those that are mushed into a MOP for retrieval through that MOP, and those that have recurred so frequently that a MOP would be useless for aid in retrieval of these events.

Events that are uniquely classified can be retrieved by a variety of aids. For example, sometimes a particular word can have an episode attached to it in memory if that word is strongly identified with only one event. Such attachments can be made off of particular concepts as well. Such concepts are not MOPs, but they can and do have unique memories stored as a part of them.

At the other extreme, we have events that occur so frequently that they cannot be recalled at all. These would originally be grouped as MOPs, but a tremendous number of events can overwhelm a MOP and thus make that MOP effectively useless as a memory organizer. For example, "toothbrushing experiences" are likely to have been grouped as a MOP at one time, but eventually that MOP gets to be useless for retrieval.

A MOP, then, is in between these two extremes. A MOP must organize information in such a way as to provide useful processing structures (i.e., predictions) off the backbone or temporal precedences of the MOP and still have pointers to unique episodes that have been classified in terms of that MOP. As those unique episodes begin to get mushed with other episodes, they cease to be unique and become MOPs themselves. As those new MOPs begin to grow they develop pointers to unique episodes that they organize. But, if they grow too big, they lose their power as memory aids and become merely subMOPs, or scripts with no memory capabilities (what we had previously referred to as Instrumental Scripts in Schank & Abelson, 1977).

Second Conclusion

This paper has taken us through a number of issues in the area of the representation of language and memory. There seem to be two points worth mentioning—one theoretical and one methodological.

The theory I have been trying to build here is an attempt to account for the facts of memory to the extent that they are available. In order to do natural language understanding effectively (whether by humans or machine) it is necessary to have as part of the working apparatus of such a system as episodic memory. Scripts and other higher-level knowledge structures are not simply static data structures in such a memory. Rather, they are active processors as well as the organizers of memory. Processing and storage devices must be the same in order to account for the phenomenon of reminding. In order to account for the fact that reminding and recognition confusions in memory both can be disembodied from large notions of a script to much smaller pieces, it was necessary to restructure our notion of a script to be much more particular. Full blown scripts of the kind SAM used would have to be reconstructed by memory. This reconstruction implies a subsequent decomposition. Thus, we can expect pieces of stories or experiences to be stored in different parts of memory, commonly breaking the link between them. The advantage of this setup is to more effectively understand the world around us. This more effective understanding manifests itself in better predictions about what will happen in particular, well-constructed experiences that have been built up over time. But these predictions are only as good as the initial categorizations of the world that we make. Thus, an effective categorization of new experience is the major problem for an understander as well as the major research problem facing those of us who work on understanding.

The negative effect of this breaking up of experience in order to make more effective predictions about the world is imperfect memory. People have imperfect memories because they are looking to make generalizations about experience that will serve as a source of useful predictions in understanding. That imperfect memory is a by-product of predictive understanding capabilities is a very important point for those of us working in computer modelling. I do not believe that there is any other alternative available to us in building intelligent machines other than modelling people. People's supposed imperfections are there for a reason. It may be possible that in the distant future we will build machines that improve upon what people can do; but machines will have to equal people first, *and I mean equal very very literally*.

This brings me to my methodological point. It is absolutely crucial that AI researchers and psychologists, as well as cognitively oriented linguists, begin to work together on the issues facing us. To pretend that we are interested in different things is folly—we are all working on the nature of the mind. The fact that we bring different tools to bear on this subject is terrific. So much the better for getting potentially different results and thus learning from each other. The field of Cognitive Science can have its computer modellers, its experimentalists, its field workers, and so on. When we stop arguing (and reporting on) methodology and begin to listen to what we have to say to each other, Cognitive Science will really begin to exist.

REFERENCES

Abelson, R. P., & Carroll, J. Computer simulation of individual belief systems. *American Behavioral Science*, 1965, *8*,

Bartlett, R. *Remembering: A study in experimental and social psychology*. London: Cambridge University Press, 1932.

Bower, G. H., Black, J. B., & Turner, T. J. Scripts in text comprehension and memory. *Cognitive Psychology*, 1979, *1*, 177–220.

Carbonell, J. G. *Subjective understanding: Computer models of belief systems*. Ph.D. Thesis, Yale University, Computer Science Dept., Research Report 150, 1979.

Colby, K. M. Computer simulation of change in personal belief systems. *Behavioral Science*, 1967, *3*, 248–258.

Cullingford, R. E. *Script application: Computer understanding of newspaper stories*. Ph.D. Thesis, Yale University, Computer Science Dept., Research Report 116, 1978.

DeJong, G. F. *Skimming stories in real time: An experiment in integrated understanding*. Ph.D. Thesis, Yale University, Computer Science Dept., Research Report 158, 1979.

Dresher, B. E., & Hornstein, N. On some supposed contribution of artificial intelligence to the scientific study of language. *Cognition*, 1976, *4*, 321–398.

Fillmore, C. The case for case. In E. Bach & R. Harms (Eds.), *Universals in linguistic theory*. New York: Holt, Rinehart and Winston, 1968.

Fodor, J. A., Jenkins, J. J., & Saporta, S. Some tests of implications from transformation grammar. Unpublished study cited in R. Brown & R. J. Hernstein, *Psychology*. Boston: Little, Brown and Co., 1966.

Gershman, A. V. *Knowledge-based parsing*. Ph.D. Thesis, Yale University, Computer Science Dept., Research Report 156, 1979.

Hays, D. G. Dependency theory: A formalism and some observations. *Language*, 1964, *40*, 511–525.

Hemphill, L. G. A conceptual approach to automated language understanding and belief structures. Stanford University, Computer Science Dept., AI Memo AIM-273, 1975.

Katz, J. Recent issues in semantic theory. *Foundations of Language*, 1967, *3*.

Klein, S. Control of style with a generative grammar. *Language*, 1965, *41*.

Lamb, S. M. On alternation, transformation, realization, and stratification. *Monograph Series on Languages and Linguistics*, 1964, *17*, 105–122.

Meehan, J. *The metanovel: writing stories by computer*. Ph.D. Thesis, Yale University, Computer Science Dept., Research Report 74, 1976.

Mehler, J. Some effects of grammatical transformations on the recall of English sentences. *Journal of Verbal Learning and Verbal Behavior*, 1963, *2*, 346–351.

Minsky, M. A framework for representing knowledge. In P. H. Winston (Ed.), *The psychology of computer vision*. New York: McGraw-Hill, 1975.

Quillian, R. *Semantic memory*. Cambridge, Mass.: Bolt, Beranek and Newman, 1966.

Rieger, C. Conceptual memory. In R. C. Schank (Ed.), *Conceptual information processing*. Amsterdam: North Holland, 1975.

Riesbeck, C. K. & Schank, R. C. Comprehension by computer: Expectation-based analysis of sentences in context. Yale University, Computer Science Dept., Research Report 78, 1976. Also in W. J. M. Levelt and G. B. Flores d'Arcais (Eds.), *Studies in the Perception of Language*. Chichester, England: John Wiley & Sons, Ltd., 1979.

Schank, R. C. A notion of linguistic concept. Stanford University, Computer Science Dept., A.I. Memo 75, 1968.

Schank, R. C. Outline of a conceptual semantics for computer generation of coherent discourse. *Mathematical Biosciences*, 1969, *5*, 93–119.

Schank, R. C. Finding the conceptual content and intention in an utterance in natural language conversation. *Proceedings of the Second International Joint Conference on Artificial Intelligence*. London, England, 1971.

Schank, R. C. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 1972, *3*, 552–631.

Schank, R. C. Identification of conceptualizations underlying natural language. In R. C. Schank & K. Colby (Eds.), *Computer Models of Thought and Language*. San Francisco: W. H. Freeman, 1973.

Schank, R. C. *Conceptual Information Processing*. Amsterdam: North Holland, 1975.

Schank, R. C., & Abelson, R. P. Scripts, plans, and knowledge. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*. Tbilisi, USSR, 1975.

Schank, R. C., & Abelson, R. P. *Scripts, plans, goals, and understanding*. Hillsdale, N.J.: Lawrence Erlbaum, 1977.

Schank, R. C., & Carbonell, J. G. Re: The Gettysberg Address: Representing social and political acts. In N. Findler (Ed.), *Associative networks: Representation and use of knowledge by computers*. New York: Academic Press, 1979. Also Yale University, Computer Science Dept., Research Report 127, 1978.

Schank, R. C., & Rieger, C. Inference and the computer understanding of natural language. *Artificial Intelligence*, 1974, *5*, 373–412.

Schank, R. C., & Tesler, L. A conceptual parser for natural language. *Proceedings of the International Joint Conference on Artificial Intelligence*. Washington, D.C., 1969.

Schank, R. C., Goldman, N., Rieger, C., & Riesbeck, C. MARGIE: Memory, analysis, response generation and inference in English. *Proceedings of the Third International Joint Conference on Artificial Intelligence*. Stanford, California, 1973.

Schank, R. C., Lebowitz, M., & Birnbaum, L. Integrated partial parsing. Yale University, Computer Science Dept., Research Report 143, 1978.

Schank, R. C., Tesler, L., & Weber, S. SPINOZA: Conceptual case-based natural language analysis. Stanford University, Computer Science Dept., A.I. Memo AIM-109, 1970.

Smith, E. E., Adams, N., & Schorr, D. Fact retrieval and the paradox of interference. *Cognitive Psychology*, 1978, *10*, 438–464.

Wilensky, R. *Understanding goal-based stories*. Ph.D. Thesis. Yale University, Computer Science Dept., Research Report 140, 1978.

Winograd, T. *Understanding natural language*. New York: Academic Press, 1972.