UC Merced
Proceedings of the Annual Meeting of the Cognitive Science Society

Title
Towards a Memory Architecture that Supports Reminding

Permalink
https://escholarship.org/uc/item/7xb6b3fp

Journal
Proceedings of the Annual Meeting of the Cognitive Science Society, 8(0)

Authors
Kolodner, Janet L.
Cullingford, Richard E.

Publication Date
1986

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Towards a Memory Architecture that Supports Reminding

Janet L. Kolodner
Richard E. Cullingford

School of Information and Computer Science
Georgia Institute of Technology
Atlanta, GA 30332

Abstract

The phenomena of *reminding* has been receiving quite a bit of attention in the past few years. Researchers have been looking at how previous experience can help in understanding and problem solving. As much of this work has shown, reminding is a complicated process. In this paper, we present requirements on a cognitive architecture that can promote and support these observations. The architectural requirements are those needed by a machine that can use experience to reason. To motivate these requirements, we present observations of reminding that are based on analyses of both actual remindings and hypothetical cases.

Why Worry About Reminding?

The phenomenon of *reminding* has been receiving quite a bit of attention in the past few years (e.g., Kolodner, 1983; Lebowitz, 1983; Reiser, 1983; Schank, 1982). Researchers have been looking at how previous experience can help in understanding (Riesbeck, 1981; Schank, 1982) and problem solving (Bain, 1985; Carbonell, 1983; Hammond, 1984; Holyoak, 1985; Kolodner et al., 1985; Ross, 1982). As much of this work has shown, reminding is a complicated process. Our own research on problem solving (Kolodner, 1983b, 1985; Kolodner & Kolodner, 1985; Kolodner et al., 1984) has led us to analyze many instances of reminding, and those instances along with the ones already reported in the literature (in, e.g., Reiser, 1983; Schank, 1982), lead us to believe that there is more regularity to reminding than is evident on the surface.

The questions we ask in this paper are the following: What are the fundamental memory processes and structures that support reminding? What kind of architecture do these processes and structures require? In this paper, we review and add to the set of things known about reminding. With each observation about reminding that we present, we also present the architectural features necessary to support it.

This research is motivated by our research into the role of experience in problem solving and understanding. We have built memory programs that can be reminded (e.g., CYRUS (Kolodner, 1983a, 1984)) and problem solvers that can use the results of reminding to solve problems analogically (e.g., MEDIATOR (Kolodner, et al., 1985), PERSUADER (Sycara, 1985), Consumer Advisor (Turner, 1986)). Each of these problem solvers is able to solve problems in fewer steps than would be required if they were solving the problems from scratch. Each gets better over time as a result of experience. Each, however, is very slow, primarily because of the overhead of maintaining and accessing the memory for experience. We have therefore been forced to ask ourselves the questions: Where does this slowness originate from and is there a way to get around it? At first blush, the answer to this seems obvious: The memory model we are using has slowness inherent in its search procedures because they are parallel procedures implemented on a serial machine (Kolodner, 1983a, 1983b, 1984). Certainly, if we had a parallel machine, we could speed up the memory search. But this is not the whole answer, and parallelism is not the only property we want out of a cognitive architecture that can support reminding. It should also support memory's other functions. We have therefore spent some time analyzing several cases of reminding to find out what kind of memory architecture would best support it and to find out how the process can be speeded up.

An Example

We begin by putting reminding into context with respect to other reasoning processes. The dialog below is from a conversation we recorded recently while attempting to explore the process of advice giving. It illustrates an interesting reminding that we use for illustrative purposes throughout this paper.

Advisee: I need to *buy* a bookshelf to use in my study at home. The study is carpeted. Since I'm in school now, I can't afford anything expensive. I'm planning to use the bookshelf for textbooks and paperbacks. Can you suggest what kind of bookshelf I need to get?

Advisor: I don't know what bookshelves run for these days. Have you ever considered *building* your own? My roommate did that last year. He was running out of space for his books and went out and bought some lumber one weekend and built some bookshelves ...

This reminding is striking. Although the advisee was asking for advice about "buying", the reminding was of a "building" episode. There are two possible explanations for this. The problem solver inside the person could have been working on solving the original problem -- buying bookshelves. In this case, it would have attempted to elaborate its buying plan taking the advisee's needs into account. When it came up against a dead end (in this case, the fact that the advisee is a student and therefore may not have enough money to buy a durable set of shelves), it considered the goal the advisee was trying to achieve (acquisition of bookshelves), began to generate a plan more amenable to the situation (in this case, building), and in considering the "building" option, was reminded of the building episode above. We call that episode the "Pete's bookshelves" episode.

Our other explanation is the following: Perhaps the problem solver was not doing much work at all. Rather, memory (a separate processor) came up with the previous case, handed it to the problem solver, and the problem solver used it to derive an answer. In other words, the reminding of the "building" episode came through memory's consideration of the "buying" request itself or some part of the buying request.

While the first explanation makes the problem solver bear the brunt of the advisor's reasoning, the second divides the work between several different processes: the memory, a case-based reasoner, and the problem solver. How can multiple reasoning processes interact to make remindings happen and to use them appropriately? Part of the answer to this question lies in *memory's representations*. The memory must know that "buying" is a plan that is usually instrumental to achieving an acquisition goal. It must also relate the acquisition goal to the kinds of plans that can achieve it, e.g., "buying" and "building". The problem solver and memory must share these representations. The general problems to be considered here are what kinds of knowledge structures the memory has and what kinds of links there are between the knowledge structures. Another part of the answer has to do with *understanding processes*. Here we refer to the hypothesis that understanders are always doing *goal tracking*. In other words, they are always trying to determine why an actor would do the actions reported in a text or conversation. Thus, inferring an acquisition goal is automatic when hearing about a "buy" plan. Another part of the answer revolves around *memory's organization*. There is probably agreement that the knowledge structure "build" is in some way referenced by the knowledge structure associated with "acquisition". But what are the connections between the *building episode* reported in the reminding and the knowledge structure "build"? How are the knowledge structures and the individual experiences reported in remindings connected? The fourth part of the answer lies in defining the *processing a memory is able and allowed to do* on this representation. Which links can be passively traversed? Which require strategic processing? What kinds of processing must be easy for the memory to do? Finally, we must consider the *interactions between the memory and the reasoner*. The reasoner might be a language understander or a problem solver. Either way, it needs memory's knowledge to do its work. By the same token, memory, which gets reminded of previous cases in the course of reasoning must always be aware of the reasoning done by the reasoner. There must thus be means of communication between the memory and reasoning processes.

Reminding

We continue by going over previous hypotheses about reminding. Schank (1982) postulates that the *interesting* kinds of remindings arise from processing considerations: As a new event or text is being understood or analyzed, the knowledge structures that can provide the most specific explanations of the new episode are accessed. Sometimes those knowledge structures are generalized structures, while sometimes they are particular previous episodes. The particular previous episodes that are remembered in the course of understanding or analysis are what we call remindings, and they arise because of the need to make inferences.

Schank (1982) and others (Kolodner, 1983a, 1984; Reiser, 1983) describe a memory organization that supports such reminding. As new items are being processed by memory's structures (called MOPs), they are at the same time being indexed in those structures by a subset of their features not predicted by the generalized knowledge structures. That subset includes features that are predictive of other domain-related features (Kolodner, 1983a). This provides for several later functions. First, it allows a probe that describes an item to be used to retrieve that item from memory. Second, it stores the item such that if a later event that is distinct in ways similar to the first one is encountered, the first item will be found and available to inference mechanisms. Such a scheme allows an inferencer to recognize a situation that is prone to failure and also provides predictions of ways to get out of failure situations (Hammond, 1984; Kolodner et al., 1985; Riesbeck, 1981; Schank, 1982). It is thus useful for both understanding and problem solving.

CYRUS' "Diplomatic Meetings" MOP

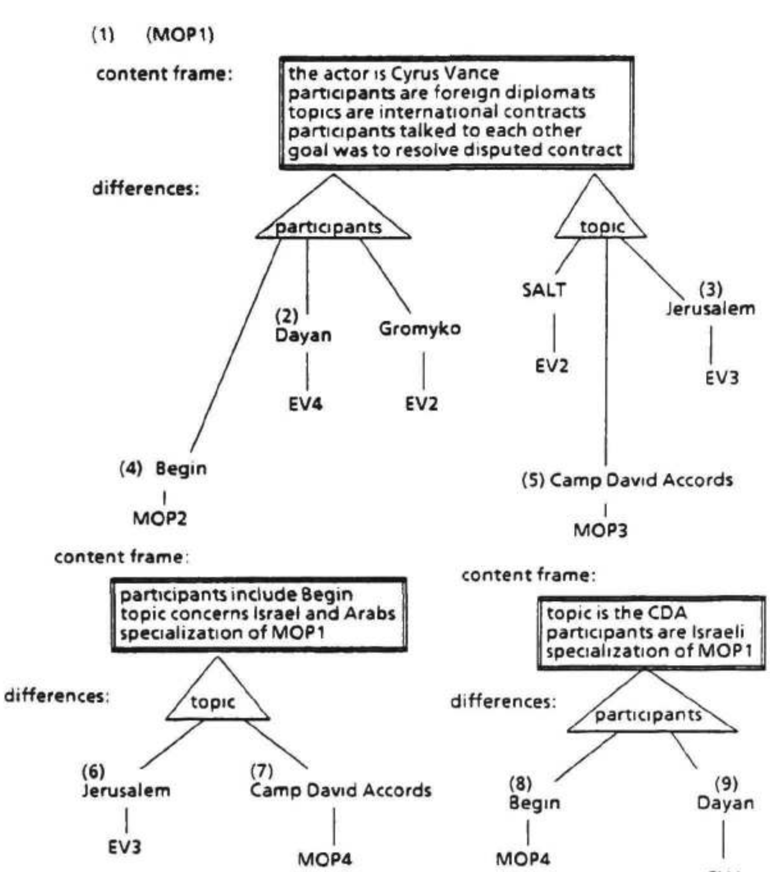

Figure 1

Figure 1 shows one of the MOPs used in CYRUS (Kolodner, 1983a, 1984). CYRUS kept track of the day to day events in the life of Cyrus Vance when he was U.S. Secretary of State. Thus, its memory structures correspond to the types of situations a secretary of state is involved in. This particular MOP organizes generalized knowledge about and individual instances of "diplomatic meetings." Since participants in such meetings can predict the topic under discussion, and topic in such situations can predict some other things, each is used for indexing in the "diplomatic meeting" context. This particular MOP holds meetings with Begin, Dayan, and Gromyko, and since there were several with Begin and all had something in common (their topics were similar (see MOP2 in the figure)), a sub-MOP of "diplomatic meetings" representing those with Begin was created. For a similar reason, there is a sub-MOP (MOP3) representing meetings about the Camp David Accords. Based on these generalizations, if an understander or problem solver with this MOP in its memory were to encounter a diplomatic meeting about the Camp David Accords, it would be able to predict some of the participants based on the knowledge available in MOP3 that participants are Israeli. If it were to encounter a diplomatic meeting about SALT, it would be able to make predictions based on the specifics of that meeting.

Before going into the intracacies of reminding, we consider the traversal process that works on these structures and the basic architectural requirements this process puts on the memory. As described by Kolodner (1983a, 1984) and implemented in the CYRUS system, memory traversal is a parallel process. A probe is received by the memory. The first thing that is done is to determine which memory structures are to be traversed. The probe is then transformed to fit the representation of each of those memory structures. These two steps make up a process called *context instantiation*. If memory were given a probe such as "Vance talked to Gromyko," for example, it might choose "diplomatic meetings" as one of the memory structures to be traversed and would transform the original probe into "Vance attended a diplomatic meeting at which Gromyko was also a participant."

The next step is *traversal of the chosen memory structure*. The probe is compared to the generalized knowledge associated with the chosen memory structure. Those of its features that are not expected by the memory structure but that are salient to the knowledge structure's domain are extracted from the memory probe. Indexes associated with those features are traversed. If an event is reached, it is recalled. If another knowledge structure is reached, either it is recalled or the process is repeated to find more specific events. For the Vance talk with Gromyko, features that are not expected by "diplomatic meetings" but that are salient to diplomatic activities are the facts that Gromyko is a Russian, that he was an ambassador, and that he is Gromyko. Links associated with each of these features would be traversed and if an event were found, it would be returned. Addition of a new item to memory works in the same way. Instead of merely traversing links, however, a new link is created for each of the features that does not already have a link associated with it. For those features that do have links associated with them, any collisions with other items already in memory leaves open the potential for generalization, while collisions with other generalized memory structures allow consistency checks and updating of generalizations.

Because memory links can only be traversed if their labels are available as memory cues, interesting problems arise when a memory probe is too general (i.e., it describes many events) and when the memory probe specifies features that are not associated with memory links. In these cases, memory does *guided elaboration* of the memory probe in an attempt to generate additional cues. Elaborations of the diplomatic meeting referred to above might include its possible place (Washington, New York, or the UN Building), its possible topic (one that concerns the US and USSR, possibly SALT), other participants, or a particular situational setting (e.g., a summit conference), among a host of other possibilities. After elaboration, memory can be probed with the generated cues.

Elaboration can be either strategic or automatic. An automatic elaboration is one that is made on the basis of a close association. Gromyko, for example, lived and worked in New York, so an elaboration of New York as a possible meeting place can be fairly automatic. Russia has a set of important political topics associated with it, of which SALT was the most important at the time Vance was Secretary of State. That too, then, would be an automatic elaboration. Strategic elaboration is necessary when hypothetical features are more remote. Generating a hypothetical topic for a meeting based on what was going on in the world at the time of the targetted meeting would be a strategic elaboration. Strategic elaborations can be arbitrarily complex and may require a fair amount of problem solving.

What does this require in terms of an architecture? First and foremost, it requires *parallel traversal of memory links*. This seems necessary because of the large number of cues that are normally available for any memory probe. If memory links emanating from knowledge structures were to be traversed serially, there would be serious timing problems. Traversal is an easy process, and this seems like a process fit for parallelism. The memory architecture described in CYRUS works such that traversal is done concurrently for each available cue. Second, and equally important, it requires *associative mechanisms that acknowledge "locks" on each of memory's links.* Rather than allowing indescriminate traversal of memory links as in a traditional spreading activation scheme, the MOPs scheme allows traversal only when a link's label is specified. This associative mechanism would fill the role of a quick pattern matcher. Third, it requires an architecture that can support both generalized structures and individual event descriptions at its nodes, and whose nodes can hold almost an arbitrary amount of generalized knowledge. This knowledge includes descriptive knowledge used for recognition and prediction as well as knowledge about salience and interestingness of features.

Observations about Reminding

The scheme presented above allows for reminding each time there are similarities between events. We continue by presenting several additional observations about reminding that add to the list of architectural requirements.

1. Remindings usually come from concrete scenes rather than abstract events.

Our analyses of remindings tell us that remindings tend to be of concrete situations. Remindings that seem to be of abstract events (e.g., a trip reminding someone of another trip) can usually be analyzed as coming from a concrete scene of the abstract episode (e.g., the reminding came after discussion of a particular taxicab ride that happened during one of the trips). To make this clearer, we must consider what an *abstract episode* is and what a *concrete scene* is. An abstract episode is made up of a variety of smaller events or scenes. It may be described in terms of an overall goal. Examples include trips and restaurant visits. A concrete scene, on the other hand, is usually relatively short in duration and strongly associated with a physical setting. It is a component of a larger episodic context and usually achieves a subgoal of that episode. The subgoal may be a contributor to the main goal (i.e., a precondition), the main goal itself, or a "clean-up" goal (i.e., a post-condition). Examples include checking in at a hotel and the paying scene in a restaurant.

To restate the premise in terms of knowledge structures, a reminding that seems to be based on an abstract knowledge structure (representing an abstract event) can usually be interpreted as a reminding based on a scene of that knowledge structure when analyzed in greater detail. While we observe people being reminded of whole episodes on the basis of similar episodes, further analysis usually leads us to conclude that the reminding derived from scene descriptions.

Consider, for example, one of Schank's (1982) restaurant examples. In this particular example, he goes for a visit to a company and eats in the company restaurant, where diners do their ordering by filling out a form. At a later time, he goes to another company, goes to eat in their restaurant, and must write down his order again. At this point, he was reminded of the previous restaurant experience. There are two different analyses of this event. In the first, one restaurant experience reminds him of another restaurant experience. In the second analysis, a particular scene deviation in one restaurant experience reminds him of the same deviation in another restaurant experience, and this scene reminding allows him to remember the rest of the previous episode. It is this second explanation that we claim is the more correct one, and Reiser's (1983) experiments bear this out.*

Schank's MOPs provide a structure that supports this kind of reminding. MOPs can have two purposes: they can package scenes and they can organize memories of events. Abstract knowledge structures, called MOPs (e.g., eating at a restaurant, going to the doctor, building bookshelves), are the ones that organize scenes. Scenes (e.g., ordering, sitting in the waiting room, selecting materials, installing the completed object), which are more specific and tend to be included in several different types of more abstract situations, organize memories of events.

This gives us one way of explaining the "Pete's bookshelves" reminding. Both "building" and "buying" of a large object require going to a store, selection of the object, installation of the object, etc. Each of these scenes is shared across both plans. "Installation" episodes, for example, whether they come from a "buying" or a "building" experience, are stored in the same place. If the advisor had been reminded of

*In fact, Reiser makes a stronger point in his experimentation. According to his experiments, scenes themselves are not enough for reminding. The situation in which the scene is embedded (i.e., the MOP) is also important. There is an implication in this for scene instantiation: scene instantiation is easy only when the abstract situation is available.

"Pete installing his bookshelves" or of some other "Pete's bookshelves" scene shared by "buying" and "building", then he would have been able to recall additional details of that episode, including details of doing the building. Figure 2 shows the memory structure that would allow this.

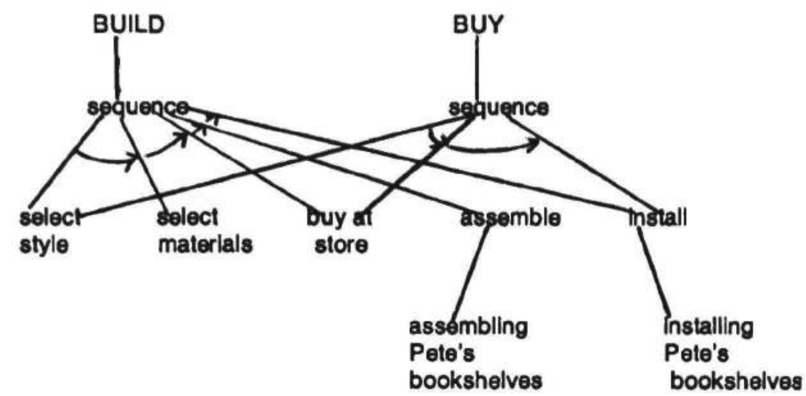

Figure 2

Now that we have the memory structure in place that can explain this reminding, we must consider the processing necessary to make the reminding happen at the right time. We have two competing explanations for this: One we will call the *elaborative hypothesis*, and the other we will call the *automatic traversal hypothesis*.

According to the *elaborative hypothesis*, some elaborative process is responsible for selecting a scene from the abstract situation (MOP).* That scene is then instantiated and traversed using the current scene description. In this case, the elaborative hypothesis might be applied several ways to result in the "Pete's bookshelves" reminding. The advisor might consider the sequence of buying a bookshelf, going through each of the major scenes, instantiating a hypothetical scene for this episode and reasoning about it. For each scene, he would take his scene description and use it to traverse the indexing network associated with the scene. If something similar to the hypothetical scene were in memory, it would be recalled at the scene level. In this case, there is a chance for reminding of "Pete's bookshelves" in any scene that "building" and "buying" have in common.

Alternatively, the advisor might be attempting to construct an image of a bookshelf with textbooks in it in a study. This in turn, might make him ask the question: How did it get there? This would lead to instantiation of the "installation" scene with the object being installed the one described by the advisee. At this point "Pete's bookshelves" would be remembered.

What does this require of a memory? It requires that *scene instantiation be fast.* This would allow an episode description at an abstract level to be described as specific scenes quickly. It also requires that the knowledge necessary for elaboration be easily available.

The alternative hypothesis, *automatic traversal,* introduces a new kind of parallel traversal of memory into the model. While previously, the only kind of traversal that was permitted was traversal of links whose labels were specified in retrieval cues, the new kind of traversal allows automatic traversal from abstract situations (MOPs) to well-known scenes. Under this hypothesis, when an abstract memory structure (MOP) is accessed, memory automatically instantiates each of the well-known scenes using the given specifications and traverses the structures of each in parallel.** This hypothesis places

*Particular strategies for elaboration are discussed and presented in (Kolodner, 1983a, 1984). Discussions of elaboration in general can be found in (Schank, 1982) and (Williams & Hollan, 1981).

While Schank explains all the remindings in his book based on the elaborative hypothesis, we have been able to find explanations for each based on the automatic traversal hypothesis also. The advantage of the automatic traversal hypothesis in these cases is that an explanation of a problem solving failure does not have to be constructed before reminding happens. Rather, the explanation can be derived from the reminding or can be constructed on the basis of two episodes. While Schank claims that explanation is guiding reminding, another requirement on the architecture: It requires that scene instantiation should be *automatic* in the architecture as well as fast.*

2. Remindings often seem very visual, but are only rarely of objects themselves. They are usually of objects in the context of some situation (i.e., a scene).

People often report remindings that are very visual (e.g., the legs on a particular table I saw while house hunting reminded me of the legs on my mother's dining room table). There are two ways to interpret such remindings: a view of an object may remind a person of a view of another object or a view of an object *in a situation* may remind a person of an object *in an analogous situation*. Our analyses of several such visual remindings lead us to believe that this second explanation is predominant. Usually the remindings people have of objects are not remindings of just the object itself, but of the object in a setting or scene. In the table example just given, standing in the position where the legs of the table could be viewed in a certain way reminded the person involved of a situation when she was standing in her mother's dining room and looking at the table legs int he same way.

Consider, again, the "Pete's bookshelves" example. One way we have tried to explain that reminding is by saying that the bookshelf description given by the advisee in the initial problem description reminded the advisor of a bookshelf he had *seen* that fit that description. But in that case, would the reminding come from a description of the bookshelves themselves, or from placing the given description of bookshelves in several situational contexts (e.g., "buying materials" "construction", "installation", "designing") and being reminded through one of those contexts.

We are more comfortable with the second description since visual reminding is then another example of reminding based on concrete scenes derived through elaboration. There are several hypotheses we can make based on this interpretation: (1) Organization of memories is rarely around objects themselves, but rather around situations the objects might be embedded in. (2) Visual memories might not be organized very much differently than other memories.

3. There may be many explanations of how a particular reminding happens. Remindings can be situation-specific (scene based) or thematic.

We again consider the reminding in the dialog above. Since it was not a reminding that one of us had, we cannot know exactly where it came from. We can, however, try to explain it reasonably, and we find that there are several reasonable explanations. The explanations we have given up to now were based on situation-specific knowledge structures (scenes) and in-context remindings. We can also explain the reminding in terms of thematic similarities between the advisee's problem and "Pete's bookshelves." Consider the following scenerio for reminding:

> The advisor knew that Pete had limited resources when he made his bookshelves, and additionally that he was a student or that his books were abnormally tall or heavy (like textbooks).

Our discussion so far has been about the usefulness of concrete situations represented by scenes for reminding. The scenerio above makes no references to concrete situations. Instead, it refers to a set of goals and conditions on those goals common to the two cases. We propose that the reminding comes through a common constellation of goals (acquisition of an object, preserve a limited resource) couples

that is not necessarily always the case. Rather, reminding can help with explanation. This is, of course, one of the roles experience should play in reasoning.

*** It may seem that we are inventing this piece of processing in an ad hoc way. Having invented it, however, we can explain its usefulness. Especially in planning, it is useful to be able to recall relevant planning failures as soon as possible. This is one way to enable that. Some may argue that doing it this way may result in a proliferation of remindings that we don't see in people. This is not necessarily the case, however, since instantiations of scenes will be fairly plain, and the only remindings we would therefore expect would be based on features of a scene if all ran according to the specifications so far.

with a set of common features novel to that goal set.

We explain this scenerio and others relying on plan and goal similarities by referring to thematic or cross-contextual knowledge structures in memory. The thematic structure that we call "Acquire Object; Limited Resources" (AO;LR) is important here. Such a knowledge structure represents knowledge about the interaction of goals, the interactions of plans and goals, and conditions on goal achievement. Schank (1982) calls these structures TOPs. This particular TOP represents knowledge about a particular kind of goal-goal interaction in which the normal plan for achievement of an acquisition goal violates a resource preservation goal.* In these cases, there is a goal of acquiring an object (usually done by "buy"), and there is also a goal of conserving a limited resource that would normally be used for acquisition (money or time). The kinds of situations represented by TOPs call for planning such that all conditions are taken care of, and there are often known ways of dealing with such situations.** In this case, buying second hand, buying at discount stores, and building are ways to conserve money, while using an agent or ordering from a catalog may be ways to conserve time.

TOPs organize plans that take multiple goals and conditions into account, but how? There are two ways. One relies on application of strategic planning knowledge. The other is more like the organization of events in MOPs: Events are organized by predictive differentiating features and found by automatic traversal of links whose values are specified in a memory probe.

Hammond (1984) gives a good explanation of the strategic organization. He claims that TOPs relate goal constellations to plans through complex questions asked of the input. The answers to the questions direct planning in appropriate directions. In AO;LR, the questions that might be asked are "what is the resource," "how important are the looks of the object," "is the object simple or complex in structure," "does the object have a lot of interacting components," "are cast-off objects of this type available," "can cast-off objects of this type be made to function appropriately," etc. These questions are the most important preconditions of the abstract plans that are available for acquiring objects with limited resources (e.g., buy second hand, buy from a catalog, build). Asking such questions of the advisee's problem description could result in either the "build" or the "buy second hand" plan being chosen. "Build" and "buy second hand" function here as abstract plans, but are examples of what we previously called *abstract situations*. They are represented by MOPs, and each has a set of more refined plan steps associated with it, represented by MOPs and eventually scenes at the most concrete levels. After choice of an abstract plan, the problem description would allow "Pete's bookshelves" to be remembered through one of those structures by traversing their scenes as described above.

According to this formulation of thematic structures, *at least some knowledge structures (thematic ones) must have strategic knowledge associated with them* that is made available to reasoning processes as soon as the thematic structure is accessed. This also gives an explanation of how situation-specific knowledge structures (MOPs) can be triggered, and provides an explanation of reminding of "Pete's bookshelves" in the course of considering several planning alternatives.

The second way TOPs organize plans is through indexing much like that done in MOPs. This organization provides an explanation for the reminding scenerio above. Here we assume that indexing in TOPs is according to features of a situation that make planning predictions. Thus, novel goals in tandem with those of the TOP would be used for indexing, as would other features (e.g., occupation, hobbies) that predict plans and plan steps appropriate to a particular individual. "Pete's bookshelves" in the scenerio above would thus be indexed in AO;LR by its novel conditions (e.g., advisee is a student). The reminding can be explained as coming from the TOP called "Acquire Object; Limited Resources" (AO;LR) by traversing an index associated with the other condition. "Pete's bookshelves," also indexed by that condition, would be remembered. If novel conditions or constraints of a new problem and an old one coincide, we expect reminding.* Figures 3 and 4 show some of the memory structures we have been discussing.

*It is a specialized case of a thematic situation in which plans for an achievement goal violate a preservation goal. Hammond (1984) calls this PVP.

**Hammond (1984) gives a good explanation of the use of these structures in planning.

*We do not consider here how TOPs are recognized or triggered.

Acquire Object; Limited Resources

Strategic knowledge:

What is the resource?

If time, "catalog buy"

How important are the looks of the object?

If not very important, and time is available, then "build" or "buy second hand"

Is the object simple or complex in structure?

If complex, rule out "build" unless actor has expertise

If simple, try "build"

Are cast-offs available?

If not, rule out "buy second hand"

Can cast-offs function appropriately?

If not, rule out "buy second hand"

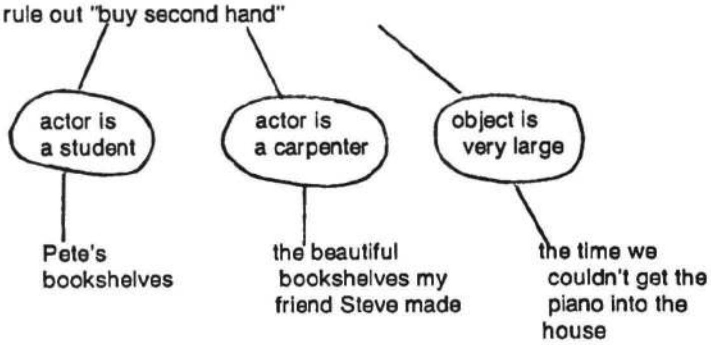

Figure 3

The BUILD plan

Preconditions:

Actor must be capable of building

Actor must have building instruments available

Sequence of events:

Select style (a plan)

Select types of materials (a plan)

Buy at store (a scene)

Assemble (a plan)

Install (a scene)

Figure 4

What kinds of processing are necessary to make all of this happen? The same ones required previously: parallel traversal of MOP/TOP indices, fast elaboration and scene instantiation, and the capability of associating almost arbitrary amounts of generalized knowledge with memory structures.

4. Remindings can be immediate or require a lot of effort.

Remindings can happen immediately when hearing of or reading about something, or might happen after a lot of understanding or problem solving effort has been expended. Remindings often require a fair amount of inference. Sometimes an explanation for an anomolous event must be derived, and reminding is on the basis of the explanation. Sometimes reminding doesn't happen until after something goes wrong, when the reasoner is attempting to figure out what went wrong. Sometimes it comes in the process of coming up with a plan in the first place. In this case, reminding can be of past successes or failures (Kolodner, 1985). Remembering failures allows a problem solver to avoid past mistakes. Remembering successes may provide shortcuts in problem solving. Sometimes reminding happens in the course of determining whether a plan to be proposed is a good one. Suppose, for example, that Pete was known as a poor handiman (contrary to the preconditions for using the "build" plan), and not much is known about the skills of the advisee. In this case, reminding of Pete's bookshelves would confirm that the "building" plan can be done with only minor prerequisites.

The point to be made here is that if reminding can happen almost any time an agent is reasoning, then memory traversal and reasoning must be happening at the same time. This requires *an architecture in which memory is always cognizant of any reasoning that is being done.* If memory is always attempting to integrate what it sees the reasoner doing into its structures, then it can be reminded at any time based on whatever knowledge is currently available. At the same time, memory must have a way of interrupting reasoning processes and telling them that it has been reminded of a good case.

Summary: A Cognitive Architecture

We have presented a number of observations about and explanations of reminding, and have presented a set of requirements on a cognitive architecture. We summarize by listing the set of constraints on a cognitive architecture that promotes and supports reminding:

1. It must allow for highly parallel traversal of memory links.
2. It must have associative mechanisms that acknowledge "locks" on each of memory's links.
3. Memory's nodes must be able to hold either generalized structures or individual event descriptions.
4. At least some memory nodes must be able to hold the arbitrary amounts of generalized knowledge associated with generalized structures. This includes knowledge necessary for automatic elaboration and for strategic processing.
5. Instantiation of concrete scenes from memory structures representing abstract situations must be fast. Under one hypothesis about memory traversal, instantiation must also be automatic.
6. Memory processing must be concurrent with other reasoning processes and memory must be cognizant of other processing being done.

There is currently interest among both psychologists and AI researchers in reminding and the use of remindings in reasoning. At the same time, there is considerable interest among researchers in both areas in massively parallel architectures. It seems that the time is right for researchers studying high level memory processing and those studying cognitive architectures and building massively parallel machines to start talking to each other. A cognitive architecture that supports reminding requires at least some of the things being built into massively parallel hardware, in particular, the massive parallelism. It also requires several things that the architecture people have not considered: fast instantiation, high storage memory nodes, and support for strategic processing within the architecture. With hardware of this type, more massive episodic memories could be built and more sophisticated work on case-based reasoning could be done. At the same time, in attempting implementations of episodic memories on massively parallel architectures, we may be able to make our memory models more sophisticated. We hope this paper will be the beginning of an attempt to combine the needs and aims of the two groups, so that additional research will be done by both groups to make sure architecture and processing needs match.

References

Bain, W. (1985). JUDGE: A Case-Based Reasoning System. In *Proceedings of the International Machine Learning Workshop,* Skytop, PA, pp. 1-4.

Carbonell, J. G. (1983). Learning by Analogy: Formulating and Generalizing Plans from Past Experience. In Carbonell, J. G., Michalski, R. S. and Mitchell, T. M., (eds.) *Machine Learning.* Tioga Publishing Co., Palo Alto, CA.

Hammond, K. (1984). Indexing and Causality: The Organization of Plans and Strategies in Memory. Yale Department of Computer Science Technical Report #351.

Holyoak, K. (1985). The Pragmatics of Analogical Transfer. In Bower, G. (Ed.), *The Psychology of Learning and Motivation,* Vol. 19, pp. 59 - 87.

Kolodner, J. L. (1983a). Reconstructive Memory: A Computer Model. *Cognitive Science,* October, 1983.

Kolodner, J. L. (1983b). Towards an Understanding of the Role of Experience in the Evolution from Novice to Expert. *International Journal of Man-Machine Systems,* November, 1983.

Kolodner, J. L. (1984). *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Kolodner, J. L. (1985). Experiential Processes in Natural Problem Solving. Technical Report No. GIT-ICS-85/23. School of Information and Computer Science. Georgia Institute of Technology. Atlanta, GA 30332.

Kolodner, J. L., and Kolodner, R. M. (1985). The Role of Experience in Clinical Problem Solving: Introduction and Framework. Technical Report #GIT-ICS-85/21. School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA 30332.

Kolodner, J. L., Simpson, R. L., and Sycara-Cyranski, K. (1985). A Process Model of Case-Based Reasoning in Problem Solving. *Proceedings of the International Joint Conference on Artificial Intelligence,* Los Angeles, CA.

Lebowitz, M. (1983). Generalization from Natural Language Text. *Cognitive Science,* Vol. 7, pp. 1-40.

Reiser, B. J. (1983) Contexts and Indices in Autobiographical Memory. Cognitive Science Technical Report No. 24. Yale University. New Haven, CT.

Riesbeck, C. K. (1981). Failure-driven reminding for incremental learning. In *Proceedings of International Joint Conference on Artificial Intelligence,* Vancouver, BC.

Ross, B.H. (1982). Remindings and Their Effects in Learning a Cognitive Skill. Cognitive and Instructional Sciences Series CIS-19. Xerox Palo Alto Research Center, Palo Alto, CA.

Schank, R. C. (1982). *Dynamic Memory: A Theory of Learning in People and Computers.* Cambridge University Press, London.

Simpson, R.L. (1985). A Computer Model of Case-Based Reasoning in Problem Solving: An Investigation in the Domain of Dispute Me liation. Ph. D. Thesis. Technical Report #GIT-ICS-85/18. School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA.

Sycara-Cyranski, K. (1985). Precedent-Based Reasoning in Expert Labor Mediation. Technical Report #GIT-ICS-85/22. School of Information and Computer Science, Georgia Institute of Technology, Atlanta, GA 30332.

Turner, R. (1986). Using Preconditions in a Case-Based Advisory System. *Proceedings of the 1986 IEEE International Conference on Systems, Man, and Cybernetics,* Atlanta, GA, October, 1986.

Williams, M. W. & Hollan, J. D. (1981). The process of retrieval from very long-term memory. *Cognitive Science,* Vol. 5, pp. 87 - 119.