THE USE OF REMINDINGS IN PLANNING*

Kristian Hammond
Department of Computer Science
Yale University

Abstract

In recent years, much research has been aimed at the study of episodic reminding and its relationship to the process of understanding in both man and machines. Because of the ill defined nature of the functionality of episodic remindings in the understanding process, however, little progress has been made in uncovering the nature of the memory organization that supports these remindings or how they are used to help in understanding.

This paper will present a functional view of episodic reminding from the perspective of planning rather than understanding. Because of the clearer functionality of planning, a more straightforward view of the use of these remindings and the memory organization that supports them is possible. This paper will discuss three situations in which episodic memories can be used in planning: problem anticipation, plan construction and plan repair. The memory organization that each of these uses implies will also be discussed. This view of memory will be presented in terms of a case-based planner, CHEF, which makes extensive use of episodic remindings in constructing new plans.

1 Episodic Remindings in Understanding and Planning

Interest in the occurrence and use of episodic remindings abounds in both Artificial Intelligence and psychology. Studies of analogical reasoning, learning from experience and episodic memory organization all center on the notion of using features in one situation to access memories of a past one. In AI, the focus of interest has been on the structures that are required to simulate these remindings in machines. In psychology, the approach has been to experimentally clarify the features that are used to access them. Unfortunately, both of these approaches seem to ignore the most important aspect of these remindings: their function.

Without a clear idea what function episodic remindings serve, there are strong limits on what can be said about the memory organization that supports them and about the information that is available to any process that accesses them. Because of this, most of the discussion about remindings has been reduced to examining lists of the common features between an episode and the reminding it elicits. There has been little discussion about how functional needs determine the vocabulary that may be used to index episodes in memory or about the processes that chose the features in an input episode to use in accessing past episodes in memory. There is a great deal of discussion of remindings that ignores the constraints that would rise out of a theory of what purpose they serve.

One reason that many of these theories lack a functional view of remindings is that most of them come out of the context of natural language understanding in which the use of episodic

*This report describes work done in the Department of Computer Science at Yale University. It was supported in part by ONR Grant #N00014-85-K-0108.

remindings is somewhat ill-defined. One way around this problem is to shift the study of episodic reminding into an area where the task is clearer and the possible functionality of the remindings is more straightforward: planning. The goals and requirements of a planner are somewhat better defined than those of an understander. Likewise the situations in which episodic remindings would be of use to a planner are also better defined. In planning there is a clearer notion of when a reminding would be of use, what it would be used for and what information the planner might have at hand to find it. This clarity provides strong constraints on the vocabulary used to store episodic memories in the first place and on the choice of the features in an situation that are going to be used to search for a past episodes for use in that situation.

The discussion in this paper will center around the memory structures used in the computer program CHEF, a case-based planner in the domain of Szechwan cooking that makes heavy use of its past planning experiences to deal with current goals. The organization used by this planner provides memories for use in initial plan construction, problem anticipation and plan repair. These memory organizations provide what we call *directed remindings*, because they are intentionally sought by processes actively seeking solutions to present problems using past memories.

2 Remindings in Understanding

In 1980, Schank [Schank 82] proposed a view of remindings as the natural by-product of goal tracking while understanding stories. He suggested a set of memory structures that would account for the association of certain episodes in memory and allow access of one from the other. In his initial work, however, he did not suggest a model of how these remindings could be used in the understanding process. Following this proposal, Dyer [Dyer 82] designed an implementation of many of the organizational ideas suggested by Schank. His understander, BORIS, was "reminded" of memory structures similar to Schank's, but did not make any use of them in actually understanding stories. More recently Dyer [Dyer 85] has continued to look at computer simulations of understander's that have remindings but still has done no more in the way of explaining their use.

In Schank's current work [Schank 86], he has suggested using remindings for verification of explanations. He also has students working on reusing past explanations to fill in causal gaps in stories being read by an understanding program. Even in this work, however, there is some difficulty in defining the function of the remindings, which undercuts the effort to demonstrate how and when they arise.

In psychology, a great deal of work on analogical reminding has been done by Gentner [Gentner and Landers 85] and Holyoak [Gick and Holyoak 83] in trying to pin down the features used to access memory. They have both taken the approach of providing subjects with understanding and planning tasks that elicit remindings and noting the commonalities between the reminding and the initial episode. The object of this work has been to derive the features that are used to organize memory.

Unfortunately, both Gentner and Holyoak seem to ignore the fact that a reminding of one episode from another is a reminding out of all of memory. They forget that their task is not to just explain why one episode was found but also explain why others weren't. They are satisfied to explain a reminding from an episode to one in the past by simply listing the features that they have in common. But the major problem with this work is that it avoids the fact that memory is used for something. Both Gentner and Holyoak seem to be looking at memory and reminding as though it were a phenomenon in a vacuum, without any function. Because of this, they have failed to ask the questions about the purpose to which the reminding they have studied are being put, and thus have failed to ask why and how they were retrieved in the first place.

A striking exception to the non-functional approach to the study of remindings in understanding is the work of Seifert [Seifert 85]. While her experimental paradigm is similar to Gentner's, her materials were aimed at discovering the processing situations that evoked remindings as well as the features used to index them. Her initial work seems to indicate that remindings are sought to fill in explanatory gaps in narratives and that the process that retrieves the remindings is more directed than it has previously been thought to be.

The general problem of discussing episodic remindings from the perspective of understanding is that the tasks involved in understanding are somewhat ill-defined. Because of this, the function of remindings in those tasks is equally ill-defined. Without an idea of the function of remindings, the theories that have arisen to explain them have lacked any notion of when the remindings will occur, what features are actually used to access them and what form the memories themselves will have to take in order to be useful. The similarities between episodes that have been noted within remindings have been interesting, but they only tell part of the story.

3 Remindings in Planning

The task of planning is somewhat more straightforward than the task of understanding. As a result, the specific functions to which an episodic memory can be put are somewhat more well defined than those in understanding. In planning, remindings of past situations can be used to suggest planning strategies, repair faulty plans, provide complete plans and warn of potential failures. These functions are such that they actually define what the planner needs and what it has on hand to search for a reminding to satisfy those needs. In planning, a theory of reminding is more than a list of the similarities between two situations. It is a theory that includes the vocabulary required to get the reminding, the process which chooses the features in one situation that are used to search for one and a description of the kind of information a planner is trying to get from it.

Carbonnel [Carbonell 83] has suggested one of the the more successful approaches to the use of remindings in planning and problem solving in his model of problem solving through analogy. His model centers around the idea that past solutions to problems can provide strategic information which can be applied to new problem solving situations. The vocabulary used to store past instances is based on a partial processing of problems. If the processing steps in one problem match those taken in a past one, the past steps are accesses from memory and applied to the current situation. The problem solver is always trying to find past solutions that can be applied to the present case. The information that the problem solver transfers from the old situation to the new consists of the the steps taken in the old case. While Carbonell's view of remindings gains a great deal from the function served by the memories retrieved, he suggests only one use that these memories serve.

4 Remindings in CHEF

CHEF is a case-based planner that tries to use past memories in all aspects of its planning activity. There are three areas in which this has been most successful:

- Plan Construction, which makes use of a memory organization of past plans indexed by the goals that they satisfy and the problems they avoid.
- Problem Anticipation, which makes use of a memory of the planner's own failures, indexed by the goal features that predict them.
- Plan Repair, which makes use of a memory of past repairs, indexed by descriptions of the problems that they solve.

Before looking at how CHEF makes use of remindings in planning, it is important to get an idea of how it plans in general.

CHEF's input is a set of goals for different tastes, textures, ingredients and types of dishes and its output is a single recipe that satisfies all of its goals. Its basic algorithm is to find a past plan that satisfies as many of the most important goals as possible and then modify that plan to satisfy the other goals as well.

Before searching for a plan to modify, CHEF examines the goals in its input and predicts any failures that might rise out the interactions between the plans for satisfying them. If a failure is predicted, CHEF adds a goal to avoid the failure it to its list of goals to satisfy and this new goal is also used to search for a plan. For example, if it predicts that stir frying chicken with snow peas will lead to soggy snow peas because the chicken will sweat meat into the pan, it searches for a stir fry plan that avoids the problem of vegetables getting soggy when cooked with meats. In doing so, it finds a past plan for beef and broccoli that solves this problem by stir frying the vegetable and meat separately. The important similarity between the current situation and the one for which the past plan was built is that the same problem rises out of the interaction between the planner's goals, although the goals themselves are different.

4.1 Plan Construction - Memory of Past Plans

CHEF's approach to plan construction is somewhat different than most planners'. Rather than build plans up from primitive steps, it searches for a past plan in memory that satisfies as many of its goals as possible and then modifies it to satisfy all of the planner's other goals. Because CHEF also has the ability to anticipate planning problems (see Section 4.2) before they arise, it tries to find past plans that avoid the problems that it anticipates while also satisfying its active goals.

The plan memory that is defined by this function is straightforward: past plans are indexed by the goals that they satisfy and the problems that they avoid. A CHEF plan for stir fried beef and broccoli, for example, is stored by the fact that it is a stir fry plan, that it includes beef and broccoli and by the fact that it avoids a problem endemic to stir frying in which the liquid produced by the stir fried meat makes the vegetable fried with it soggy.

CHEF begins its search of plan memory with two sets of features in hand, the goals that it needs to achieve and the problems it wants to avoid while doing so. These goals are ranked so that CHEF is able to search for plans that satisfy its most important goals first. This ranking is based on the difficulty with which it can add the steps for a goal to an existing plan.

Because it is rare that a past plan will satisfy all of the goals that CHEF has in hand, it also needs a memory organization that reflects similarity between different goals and the different plans that satisfy them. A plan to satisfy the goal to stir fry beef, for example is similar to the plan to stir fry pork. As a result, a complex plan for beef can easily be modified to satisfy the goal to include pork. This notion of similarity is implemented in an abstraction network of goals, in which similar goals are close together in the network.

CHEF's plan memory is implemented as a discrimination net, in which plans are indexed by the goals that they satisfy and the problems that they avoid. Access to this memory is controlled by CHEF's plan retriever, a module that takes a set of goals, including the goals to avoid problems, and retrieves the past plan in memory that satisfies as many of the most important goals as possible. The planner first uses its knowledge of the relative difficulty of modifying plans to prioritize the goals it is handed. It then uses the highest priority goal to drive into memory and abandons the search on the basis of that goal only after it has failed to find a plan that even partially satisfies it.

When a goal fails to match any of those satisfied by an existing plan, the plan retriever backs off on the goal and tries to find a plan that at least satisfies a more general version of it. If there are multiple plans that do this, the planner further discriminates between them on the basis of individual features of the goals that the plans satisfy. The features it chooses to do this discrimination are dynamically determined by past failures that are associated with them and by a static ordering of features that is used if none of the features is problematic. After a goal has been used to discriminate through the plan memory, the next highest priority goal is called on to do any further discrimination that has to be done.

The remindings that CHEF gets from its plan retriever reflect the needs of the planner. They are plans that satisfy many of the planner's goals because its plan memory is organized around those goals. Because it also wants to avoid problems, the fact that a plan does so is also used to index plans. Because it can use plans that partially satisfy goals it has a notion of similarity that is also used to organizes this memory. And because it wants to minimize the work it has to do to modify a past plan to satisfy its current goals, it searches for past plans indexing on the most difficult to achieve goals first.

In dealing with a planning problem of making a stir fry dish with chicken and snow peas, CHEF anticipates the problem of the liquid from the chicken ruining the texture of the snow peas and so adds a goal to avoid this problem to the list of goals it has to achieve. Once this is done, it searches for a plan that satisfies as many of the goals as possible. Because the most difficult goal to achieve is the one to avoid the problem of the liquid ruining the snow peas, this one becomes the highest priority goal for this search. As a result, a plan that satisfies this goal and partially satisfies the others is used rather than another plan that is a stir fry dish for chicken but does not have the changes that would allow the planner to easily avoid the problem it has anticipated.

```
Searching for plan -

Placing goals in order of difficulty -
   Make a stir-fry dish.
   Avoid failure of type SIDE-EFFECT:DISABLED-CONDITION:CONCURRENT
 exemplified by the failure 'The broccoli is now soggy' in recipe
```

```
 BEEF-AND-BROCCOLI.
   Include chicken in the dish.
   Include snow pea in the dish.


Driving down on: Make a stir-fry dish.
 Succeeded -
Driving down on:
  Avoid failure of type SIDE-EFFECT:DISABLED-CONDITION:CONCURRENT
  exemplified by the failure 'The broccoli is now soggy' in recipe
  BEEF-AND-BROCCOLI.
 Succeeded -
Driving down on: Include chicken in the dish.
 Failed - Trying more general goal.
  Driving down on: Include meat in the dish.
   Succeeded -
Driving down on: Include snow pea in the dish.
 Failed - Trying more general goal.
  Driving down on: Include vegetable in the dish.
   Succeeded -

Found recipe -> REC9 BEEF-AND-BROCCOLI

Recipe exactly satisfies goals ->
    Avoid failure of type SIDE-EFFECT:DISABLED-CONDITION:CONCURRENT
exemplified by the failure 'The broccoli is now soggy' in recipe
BEEF-AND-BROCCOLI.
    Make a stir-fry dish.

Recipe partially matches ->
    Include chicken in the dish.
          in that the recipe satisfies: Include meat in the dish.

Recipe partially matches ->
    Include snow pea in the dish.
          in that the snow pea can be substituted for the broccoli.
```

In CHEF, remindings of past plans serve a function and that function defines the memory organization that houses the plans and the access process that retrieves them. As a result, the planner can use its memory to find the plans that fit its needs. These plans best achieve the goals that CHEF has been given and avoid the problems that it has anticipated. They partially satisfy goals can be found along with those that completely satisfy them. CHEF's plan memory is designed to meet the needs of the planner at large and the features that are used to access it are those that best describe the problems that the plans searched for have to solve.

4.2 Failure avoidance - memory of past failures

To avoid failures, CHEF has to be able to anticipate them, to notice that they are going to occur in order to give the plan retriever the goal to avoid them. The process that predicts the occurrence of failures is CHEF's problem anticipator which uses a knowledge base of CHEF's memory of failures.

The function of a problem anticipator is to predict failures on the basis of the surface features of a situation that have caused similar failures in the past. It predicts, for example, that the plan for the goals of including a meat and a crisp vegetable in a stir fry dish will lead to the vegetable becoming soggy after being stir fried with the meat. As the last section demonstrated, once this prediction has been made, a plan that avoids this problem can be found. But before the problem can be solved, it must be anticipated. And to anticipate these failures, CHEF's memory of them has to be organized so as to link its memory of past failures to the features of the goals that predict them.

The organization of CHEF's memory of failures is less complex than its organization of plans. It is simpler because its function and the requirements of the process that uses it are simpler. The task of the anticipator is to infer the possibility of a failure from a set of surface features. The memory of failures that it uses, then, has to be organized in a way that connects those features to the failures they predict.

CHEF's memory of failures is a simple network of nodes, in which particular failures are connected to the goal features that predict them. In CHEF, the surface features of a situation all stem from the goals that it is asked to plan for. The goals that CHEF is planning for define its situation, so it makes no sense to link memories of failures to anything but these goals. The failure in the beef and broccoli situation is linked to the goal to include meat, the goal to include any crisp vegetable and the goal to have a stir fry dish.

When a particular feature of a goal can be identified as participating in a failure, a test is built for that feature and is associated with the most specific version of the goal that allows the most general use. If all members of a class of items is associated with the failure, a link is made directly from that class to the memory of the failure itself.

When a new set of goals is handed to CHEF, its first step is to activate the features associated with each of the goals which in turn spreads activations to any memoriies of past failures that are associated with those features. If all of the features that are required to predict any failure send markers to its node, it activates itself and the planner builds a goal to avoid that failure and adds it to the planner's goal list. In looking at the goals to have a stir fry dish that includes chicken and snow peas, then, the fact that the chicken is a meat and the snow peas are a crisp vegetable activates the memory of a past stir fry failure stemming from these features.

```
Searching for plan that satisfies -
   Include chicken in the dish.
   Include snow pea in the dish.
   Make a stir-fry dish.

 Activating features:
  Activating: The dish STYLE-STIR-FRY.

  Activating: The item a MEAT.

  Activating: The item a VEGETABLE.
              The TEXTURE of item CRISP.

     Chicken + Snow Pea + Stir frying = Failure
```

```
"Meat sweats when it is stir-fried."
"Stir-frying in too much liquid makes vegetables soggy."
 Reminded of BEEF-AND-BROCCOLI.
 Fired demon: DEMONO

Based on features found in items: snow pea, chicken and stir fry
Adding goal: Avoid failure of type
SIDE-EFFECT:DISABLED-CONDITION:CONCURRENT exemplified by
the failure 'The broccoli is now soggy' in recipe
BEEF-AND-BROCCOLI.
```

CHEF's memory of failures is organized around the features of the goals and states that predict them. This lets CHEF go directly from a set of goals to be planned for to a memory of those failures that it should be trying to avoid. By also associating the failures with the actual features that caused them, features that may not have been directly attached to a goal that the planner was working on when the failure occurred, it is also able to be reminded of failures in situations that have surface features different from those of the original situation. So CHEF's memory of failures can provide appropriate predictions of problems in situations that are similar to those it has seen before at the level of initial goals and at the level of the actual causes of particular problems.

4.3 Plan modification - memory of past repairs

After CHEF is reminded of a past plan, it has to modify it to satisfy any goals that it does not already achieve. To do this CHEF uses a table of standard modifications that tell it how to add new goals to existing plans. In some case, however, CHEF needs to make changes that go beyond the scope of the modifications suggested. These are cases where past experience has shown that the standard modifications will lead to a failure.

Because CHEF is able to anticipate problems that it has encountered before, it is able to predict when these cases will arise. It is also able to use this prediction to find the past repair it used to solve the problem in the earlier instance. This is because it stores the memory of the repair it made in memory, indexed by the problem that it solves. Like complete plans that are indexed by the problems that they avoid, specific plan alterations are indexed by the problems that they fix.

One example of this occurs when CHEF is building a plan for Duck Dumplings. After running a basic plan that just replaces duck for pork in an existing recipe, it finds that the fat from the duck makes the recipe too greasy. It repairs this problem in its new plan by removing the fat from the duck before grinding it. It also stores the memory of this repair under the prediction of the failure. Later, when it is planning for a pasta dish with duck, it predicts that the same problem will occur but cannot find a pasta dish that avoids it. The best it can find is the recipe for Ants Climb a Tree, a pasta dish with pork, that it puts the duck into. Because it has predicted a problem with grease, however, it has access to the past repair that deals with it and can apply it before the failure occurs again.

```
Modifying recipe: DUCK-PASTA
```

```
to satisfy: Include duck in the dish.

Role substitution of duck for pork in recipe DUCK-PASTA.

Placing some duck in recipe DUCK-PASTA


Considering reminding:
After doing step: Bone the duck
 do: Clean the fat from the duck
 because: The duck is now fatty
 avoiding: SIDE-FEATURE:GOAL-VIOLATION
- Reminding applied.
```

By storing the memory of the past repair in terms of the problem that it solves, the planner can be reminded of it when the problem arises again and use it to solve the problem before it leads to an actual planning failure.

5 Expanding the Scope of Remindings

While CHEF makes good use of *directed remindings* there are still many other ways in which remindings of past situations could be put to good use. Two of these are in the area of plan repair. Briefly, *directed remindings* could be used to build explanations as to why a plan has failed out of past explanations of similar failures. It also could use memories of past repairs to fix current failures.

Just as CHEF reuses past plans rather than chaining together new plans out of primitive actions, an explainer could reuse past explanations from similar situations rather than chaining together new ones to explain current failures. A plan repair mechanism could also use memories of past experiences to avoid the replanning operations normally associated with plan repair.

6 Conclusions

The CHEF planner makes use past memories of its own experiences, in the form of *directed remindings*, in order to do much of its planning work. The view of reminding that it presents is an active one in which the needs of the planner define the different memory organizations that it uses. There is a relationship between the function to which a memory is going to be put and the features that are used to find it. The features are not random features that are taken from a situation and used to search memory, they are actively chosen by a planning process and reflect the needs of the process. As a result the remindings that occur also reflect those needs.

References

[Carbonell 83] Carbonell, J., Derivational Analogy and its Role in Problem Solving, *Proceedings of the National Conference on Artificial Intelligence,* AAAI, Washington, DC, August 1983.

[Dyer 82] Dyer, M., *IN-DEPTH UNDERSTANDING: A Computer Model of Integrated Processing For Narrative Comprehension,* Technical Report 219, Yale University Department of Computer Science, May 1982.

[Dyer 85] August, S. and Dyer, M., Understanding Analogies in Editorials., *Proceedings of the Ninth National Conference on Artificial Intelligence,* AAAI, Los Angeles, CA, August 1985, pp. 845–847.

[Gentner and Landers 85] Gentner, D. and Landers, R., Analogical Reminding: A Good Match is Hard to Find., *Proceedings of the International Conference on Systems, Man and Cybernetics,* Tuscon, AZ, November 1985.

[Gick and Holyoak 83] Gick, M. and Holyoak, K., *Schema induction and analogical transfer,* Cognitive Psychology, 15 (1983), pp. 1–38.

[Schank 82] Schank, R., *Dynamic memory: A theory of learning in computers and people,* Cambridge University Press, 1982.

[Schank 86] Schank, R., *Explanation Patterns: Understanding Mechanically and Creatively,* In preparation, 1986.

[Seifert 85] Seifert, C., McKoon G., Abelson R., and Ratcliff R., *Memory Connections Between Thematically Similar Episodes,* Journal of Experimental Psychology: Learning, Memory and Cognition., (1985).