An Adaptive Planner

Richard Alterman

Computer Science Division
University of California, Berkeley
Berkeley, California 94720

ABSTRACT

This paper is about an approach to the flexible utilization of old plans called **adaptive planning.** An adaptive planner can take advantage of the details associated with specific plans, while still maintaining the flexibility of a planner that works from general plans. Key elements in the theory of adaptive planning are its treatment of background knowledge and the introduction of a notion of planning by situation matching.

1. Introduction

A planner that has access to general plans (alternately abstract or high-level plans) is flexible because such plans will apply to a large number of situations. A problem for a planner working exclusively with general plans is that many of the details associated with more specific plans (e.g. sequencing information and causal relationships) must be recomputed. For a planner that works from more specific plans the situation is reversed: There is a wealth of detail, but there are problems with flexibility. I will refer to planners with the capacity to use a mix of old specific plans and general plans as **adaptive planners** [1-3]. Adaptive planners foreground specific plans, but gain flexibility, in situations where the old plan and the planner's current circumstances diverge, by having access to more general plans.

The adaptive planning techniques that will be described in this paper are sufficiently robust to handle a wide range of relationships between an old specific plan and the planner's current circumstances. For example, suppose a planner is about to ride the NYC subway for the first time, and attempts to treat an old plan for riding BART (Bay Area Rapid Transit) as an example to guide the current planning activity. Consider the steps involved in riding BART. At the BART station the planner buys a ticket from a machine. Next, the ticket is fed into a second machine which returns the ticket and then opens a gate to let the planner into the terminal. Next the planner rides the train. At the exit station the planner feeds the ticket to another machine that keeps the ticket and then opens a gate to allow the planner to leave the station. Compare that to the steps involved in riding the NYC subway: buy a token from a teller, put the token into a turnstile and then enter, ride the train, and exit by pushing thru the exit turnstile. There are a great number of differences between the BART Plan and the plan that the planner must eventually devise for riding the NYC Subway.

- In the BART case a ticket is bought from a machine, in the NYC subway case there is no ticket machine and instead a token is bought from a teller.
- In the BART case the ticket is returned after entering the station, in the NYC subway case the token is not returned after entry.
- In the BART case the ticket is needed to exit, in the NYC subway case the token is not needed to exit.

This paper will describe an adaptive planner called **PLEXUS** that can overcome these differences and in an effective manner use the BART Plan as a basis for constructing a plan for the NYC subway situation. Two versions of PLEXUS have already been constructed. This paper gives an overview of adaptive planning and PLEXUS. It includes a discussion of adaptive planning in relation to the literature, descriptions of four key elements of adaptive planning, and some details of PLEXUS' adaptation mechanism.

2. Adaptive Planning

There are four keystones to the adaptive planning position on the flexible utilization of old plans.

- An adaptive planner has access to the **background knowledge** associated with an old plan.
- In adaptive planning the exploitation of the background knowledge is accomplished by a process of **situation matching.**
- An adaptive planner foregrounds **specific plans.**
- Adaptive planners treat the failing steps of a plan as **representative of the category of action** which is to be accomplished.

Adaptive planning makes the **background knowledge** associated with an old specific plan explicit. Previous approaches to re-using old plans have dealt with an old plan in relative isolation and therefore the task of re-using an old plan has been considerably more complicated. By making the content and organization of the background knowledge explicit, it becomes possible to re-use an old plan in a wider variety of situations. Background knowledge includes general plans, categorization knowledge, and causal knowledge.

Exploitation of the background knowledge is accomplished by a process of **situation matching.** Adaptive planning uses the position of the old plan in a planning network

This research was sponsored in part by the Defense Advance Research Projects Agency (DOD), Arpa order No. 4031, Monitored by Naval Electronic System Command under Contract No. N00039-C-0235. This research was also supported by the National Science Foundation (IST-8514890).

as a starting point for finding a match to the planner's current circumstances. The **interaction** of planning knowledge and the current situation determine a plan which fits the current context and realizes the goal. The interaction works in both directions. In the direction of planning knowledge to situation, the old plan serves as a basis for interpreting the actions of other agents and the various objects in the new situation. Moreover, it provides the planner with a course of action. In the direction of situation to planning knowledge, it is the situation which provides selection cues that aid the planner in determining an alternate course of action when complications arise.

Adaptive planning foregrounds **specific plans.** It has been previously argued by Carbonell [4] that the importance of being able to plan from more specific plans is that many times a more general plan is not available. But there are other reasons why the capacity to work from more specific examples is important. Many times a more specific plan is tailor-made for the current planning situation. Furthermore, the more specific plans make available to the planner previously computed causal and ordering relationships between steps. For a more general plan these can not be determined until that steps are instantiated. Consequently, even in the cases where the more specific plan must be re-fit, many times the cost of such changes are much less than the cost of dealing with the subgoal and subplan interactions inherent in a process that works by instantiating more general plans.

Adaptive planning treats the failing steps of the old plan as **representative of the category of action** which is to be accomplished. In the case of the BART-NYC planning problem, each of the failing steps is representative of the category of action the planner eventually wants to take. An adaptive planner uses the category knowledge, as represented by the failing step, to access more general versions of that step and also to determine its eventual course of action. For example, the first step of the BART Plan, 'buying a BART ticket', is representative of the planner's eventually course of action - adapting a plan to 'buy a theatre ticket'.

3. PLEXUS - An adaptive planner

For PLEXUS the background knowledge associated with an old plan is determined by the old plan's position in a knowledge network. The network includes **taxonomic, partonomic, causal,** and **role** knowledge; the network acts as a structural backbone for its contents. PLEXUS uses the taxonomic structure not only for the purposes of property inheritance, but also as a basis for reasoning about categories. The partonomic structure (i.e. step-substep hierarchy) is used to aid in determining the pieces of network which need to be refitted in a given situation. The causal knowledge serves several functions: The **purpose** relation identifies the abstraction which maintains the purpose of a step in a plan. The **precondition, outcome, and goal** relations act as appropriateness conditions. The **reason** relation provides dependency links between a step and its justification (c.f Stallman & Sussman, 1977) [5]. Roughly, in PLEXUS, **purpose** is synonymous with 'intent', **goal** with 'aim', and **reason** with 'justification'. The purpose of 'buying a BART ticket' is to 'gain access', the goal associated with it is to 'have a ticket', and the reason for doing it is that it makes it possible to 'enter the BART station' (see figure 1). Associated with roles are type constraints on the types of objects which can fill them. The **role** relations are used by PLEXUS for both cross indexing purposes and to control inferencing. For further arguments on the importance of background knowledge see Alterman (1985), and for more details on the representation of the background knowledge see Alterman (1986) [3].

PLEXUS uses the old plan to interpret its course of action in its current circumstances. It considers the steps, one step at a time, in order. If a step is not an action it adapts substeps in a depth-first fashion before moving onto the next step in the plan. When a given step of the old plan has been adapted to the current circumstances, PLEXUS simulates a planner taking action on that step before moving onto the next step in the plan - thus, as did NASL (McDermott, 1978 [6]), PLEXUS **interleaves planning and acting.**

Associated with each step (substep) in a plan are **appropriatness conditions.** The appropriatness conditions are intended to be suggestive that a particular course of action is reasonable to pursue. Before a step is applied, PLEXUS treats the preconditions and goals of the old plan as appropriateness conditions. After a step has been applied, PLEXUS treats the expected outcomes as appropriateness conditions. Appropriateness conditions are checked by testing the type constraints associated with each of the roles attached to the appropriateness condition. The type constraints are interpreted in terms of the network.

A rough outline of the top-level decision procedure is shown below:

1) Are any of the before conditions associated with the old plan failing?
    a) Is this a case of step-out-of-order?
    b) Is this a case of failing precondition?
2) Has the current circumstances aroused a goal not accounted for by the current step?
    a) This is a case of differing goals.
3) Is the current step an action?
    a) If yes, perform the action.
    b) If no, proceed to adapt substeps.
4) Are any of the outcomes associated with the current step failing?
    a) This is a case of failing outcome?
5) Adapt next step.

If one of the before appropriatness conditions fails, or the current circumstances indicate a goal not accounted for by the old plan, one of three different types of **situation difference** is occurring: failing precondition, step-out-of-order, or differing goals. There is a fourth kind of situation difference, failing outcome, that occurs when one of the expected outcomes of a given step fails to occur. Associated with each of the types of situation difference are varying strategies that will be briefly described in the fifth section of this paper. PLEXUS does not always consider the steps in order, under certain circumstances it looks ahead to the latter steps of the plan and adjusts them in anticipation of certain changes - thus PLEXUS has an element of **opportunism** (Hayes-Roth & Hayes-Roth, 1979) [7].

The core of PLEXUS are the matching techniques it uses for finding an alternate version of a step once it determines that the step needs to be refit. To find an alternate matching action for a given situation, PLEXUS treats the failing step as **representative of the category** of action it needs to perform, and then it proceeds to exploit the background knowledge in two ways.

> By a process of **abstraction** PLEXUS uses the background knowledge to determine a category of plans in common between the two situations.

> By a process of **specialization** PLEXUS uses the background knowledge to determine an alternate course of action which is appropriate to the current circumstances.

PLEXUS accomplishes abstraction by moving up the categorization hierarchy until it finds a plan where all the before appropriatness conditions are met. PLEXUS accomplishes specialization by moving down the categorization hierarchy until it finds a plan that is sufficiently detailed to be actionable.

4. Core of the Matcher (Managing the Knowledge)

There are at least two important considerations concerning the control of access to knowledge. One consideration is that there is a danger of the planner becoming overwhelmed by the wealth of knowledge (c.f. **saturation,** Davis 1980 [8]) that is available. The problem is that there are potentially too many plans that the planner might have to consider, and consequently, the planner could get bogged down in evaluating each candidate plan. Somehow the planner needs to be able to selectively consider the various alternatives available to it.

Another consideration in the control of access to knowledge comes form the cognitive science literature and is referred to as the problem of **enumeration** (e.g. Kolodner, 1983 [9]). The problem of enumeration is that humans do not appear to be capable of listing all the instances of a category without some other kind of prompting. When asked to list the states of the union, human subjects do not accomplish this by simply listing all the members of the category of states. For the concerns of adaptive planning the problem of enumeration comes in a slightly different guise. Given an abstract plan it is not reasonable to assume that a human planner could enumerate all of the specializations of that abstract plan.

The first of these considerations dictates that PLEXUS be **selective** in its choice of planning knowledge to use. The second of these considerations acts as a sort of **termination condition:** sometimes the planner knows the right plan but circumstances are such that it cannot find it. As a result of these considerations, PLEXUS' abstraction and specialization processes must be constrained. While moving up the abstraction hierarchy PLEXUS maintains the function of the step in the overall plan. Movement down the abstraction hierarchy, towards more detailed plans, is controlled by the interaction between the planner's knowledge and the current circumstances.

4.1. Abstraction

The way to think about abstraction of a plan is that it removes details from that plan: if a particular plan fails to match the current situation, some of the details of that particular plan must be removed. Moving up the abstraction hierarchy removes the details that do not work in the current situation while maintaining much of what is in common to the two situations. Effectively, the movement of abstraction is discovering the generalization which holds between the old and new situations given that a difference has occurred.

A given plan step can have any number of abstractions associated with it. Choosing the wrong abstraction can lead to the wrong action. The planner can avoid this problem by applying the following general rule:

> Ascend the abstraction hierarchy that maintains the **purpose** of the step in the plan that is being refitted.

By moving up the abstraction hierarchy that maintains the purpose of the step, PLEXUS attempts to maintain the function of the step in the overall plan and thereby mitigate the propagated effects of changes.

In general PLEXUS uses two techniques for moving up the abstraction hierarchy.

- If a plan is failing due to the existence of a particular feature of a plan, move to the point in the abstraction hierarchy from which that feature was inherited.
- Incremently perform abstraction on a failing plan.

The first technique applies in situations where there is a specific feature in the old plan that does not exist in the current situation. The second technique of abstraction applies in situations where there is no identifiable feature which has to be removed. In such cases, PLEXUS incrementally moves up the abstraction hierarchy. In either case, for each abstraction it tries to find a specialization that will work in the current context. If it fails to find a specialization for a given abstraction, it moves to the next abstraction in the abstraction hierarchy.

4.2. Specialization

Via the process of specialization PLEXUS moves from a more abstract plan towards more specific examples. PLEXUS navigation thru the network is dependent on the planner's current circumstances. PLEXUS descends down the classification hierarchy one step at a time. PLEXUS tests the applicability of a specialization by checking the before appropriateness conditions; if one of these conditions fails the movement is rejected. At each point in the hierarchy PLEXUS is faced with one of five options:

1) Is the plan sufficiently detailed to act on?
2) Is there a feature suggested by the type of situation difference which cross indexes some subcategory of the current category of plan?
3) Is there an observable feature which cross indexes some subcategory of the current category of plan?
4) Is there an observable feature with an abstraction that cross indexes a subcategory of the current category?
5) Is there a salient subcategory?

PLEXUS stops descending the categorization hierarchy when it gets to a leaf node (option 1). If the node is not a leaf it continues to descend (options 2-5). Sometimes the type of situation difference suggests cues for subcategory selections (option 2). Sometimes 'observable features' act as cues for subcategory selection (options 3-4). These 'observable features' can either directly cross index some subcategory of plan (option 3), or have an abstraction which cross indexes a subcategory of plan (option 4). Certain subcategories are salient regardless of context and can always be selected (option 5).

Many of these techniques are employed in the following example: Suppose a planner wants to transfer between planes at the Kennedy Airport in NYC. The planner's normal plan for transferring between planes is to walk from the arrival to the departure gate. But when the planner arrives at Kennedy Airport the arrival and departure gates turn out to be in different terminals. Suppose the planner decides that the walk between terminals is too strenuous, and thus a new goal is aroused: preserve energy. The detection of this goal has no correspondent in the old plan and it is determined that the plan must be adjusted to account for this goal; this is a case of the **differing goals** type of situation difference. By a process of abstraction, PLEXUS moves up the categorization hierarchy from the plan to 'walk' to the more general plan of 'travelling'. Next PLEXUS must determine an alternate plan, within the category of 'travelling', from which to act. The newly aroused goal acts as a cue for selecting 'vehicular travel' as a potential subcategory of plan from which to act (option 2). Suppose the planner has never used a shuttle before at an airport, but it sees (observable feature) a sign concerning 'airport shuttles'. An abstraction of 'shuttle' acts as a cue for selecting 'mass transit travel' as a subcategory of 'vehicular travel' (option 4). Moreover, 'shuttle' is a cue for selecting 'shuttle travel' as a subcategory of 'mass transit travel' (option 3). 'Shuttle travel' is sufficiently detailed for PLEXUS to attempt to adapt (option 1). See Alterman (1986a) [3] for further details and a trace of PLEXUS handling this planning problem.

5. Four Types of Situation Difference

PLEXUS currently recognizes four kinds of situation difference: failing precondition, failing outcome, different goals, step-out-of-order.

A **failing precondition** situation difference occurs when one of the preconditions of a step (plan) fails. For failing preconditions PLEXUS moves up the abstraction hierarchy, according to the **purpose** of the step, to a point at which the failing condition has been abstracted out. In the event that PLEXUS cannot find a specialization of that category of plans, it continues to incrementally move up the abstraction hierarchy indicated by the **purpose** relation. For failing preconditions either of PLEXUS specialization techniques are appropriate.

A **failing outcome** situation difference occurs, if after applying a plan (step) PLEXUS discovers that one of the expected outcomes of that plan was not achieved. There are three courses of action available. The obvious course of action is to try the plan again. A second course of action, is to use the **reason** relation to determine the other steps of the plan which are effected by the failed outcome, and determine, via abstraction and specialization, if the planner can continue on its course action because there is an alternate interpretation of the latter step which does not require the failed outcome. If all else fails, the third option available to the planner is to find and perform an alternate version of the failing step. For failing outcomes, if the current plan step is being re-interpreted, abstraction occurs incrementally. If PLEXUS is trying to re-interpreted a step related to the current step by a **reason** relation, abstraction occurs using the failing outcome as a feature to abstract out of the plan. For the second and third cases PLEXUS uses both of the specialization techniques available to it.

A **differing goal** situation difference occurs if the planner's current circumstances arouse a new **goal** not accounted for by the old plan. For this kind of situation difference, abstraction occurs incrementally, and specialization requires that the new plan be indexed under both old and new goals.

A **step-out-of-order** situation difference occurs, when PLEXUS encounters a situation where it needs to apply a step out of order. There are two adjustments that are possible when a step-out-of-order situation difference occurs, PLEXUS can either delete the intermediate step(s), or reorder the steps of the old plan. If a step can be applied out of order, PLEXUS uses abstraction and specialization in an attempt to find an alternate version of the plan with the correct ordering of steps. Under such a situation, PLEXUS can use the new ordering constraint as an index for specialization purposes. In the event an alternate plan with a different ordering of steps can not be found, PLEXUS performs the step-out-of-order, removes it from the sequence of steps, and proceeds with attempting to apply the failing step.

6. An example

The BART-NYC subway planning problem provides examples of three of the types of situation difference (see figure 1).

Adapting buy a BART ticket.
The first step of the BART plan fails in the NYC subway situation because there is no ticket machine. This is a case of **failing precondition,** and therefore PLEXUS abstracts out the failing condition, 'exist ticket machine', and specializes, using the salient subcategory, to 'buy theatre ticket', which it proceeds to adapt to the NYC subway situation. During the process of adapting this step 'ticket' gets bound to 'token'.

Adapting enter BART station.
The second step of the BART plan involves entering the station. The first substep of this step is to insert the token into the entrance machine, which the planner successfully accomplishes. The next step of 'BART enter' is that the ticket is returned by the machine. But in the NYC subway situation the ticket is not accessible, but it is possible to push thru the turnstile (the third step of 'BART enter'). Hence this is a case of **step out of order.** Having accomplished the last step of 'BART enter', PLEXUS must determine whether it should act on the intermediate step or instead delete it.

Re-interpreting BART exit.
In order to delete intermediate steps PLEXUS must treat the outcomes of each intermediate step as a case of a **failing outcome** and test to see if the latter steps in the plan effected by the failing outcome can be adapted. In this case there is only one intermediate step, 'ticket returned'. The *outcome* associated with this intermediate step is that the planner 'has the ticket' (or in this case 'token'). PLEXUS applies the second strategy associated with the situation difference type **failed outcome:** Find an alternate interpretation of the situation where that outcome is no longer necessary. PLEXUS uses the *reason* relation associated with 'ticket return' to determine which of the latter steps are effected by the failing outcome. In this case, the reason that the ticket is returned is so it can be used when exiting the station. PLEXUS must try to re-interpret 'BART-EXIT' in such a manner that it can exit without a ticket. This leads to a situation of **failed precondition** for the step 'BART-EXIT'. Via abstraction PLEXUS extracts that 'exiting an institution' is what is in common between the old plan and the new situation. PLEXUS 'observes' the exit turnstile and uses it as a cue for determining 'exit_building' as an alternate plan for 'exiting the station', where 'exit turnstile' plays the role of 'locking door'. Since it can find an alternate interpretation to 'exiting the station' that does not involve using a ticket, PLEXUS treats the **step-out-of-order situation** that occurs during execution of the plan 'BART enter' as a case of deletion. For a more detailed discussion of this problem and a trace see Alterman (1986) [3].

7. Discussion

Like the early general problem solving planners [10,11] adaptive planning is concerned with the problems of **generality** and **flexibility.** Unlike them it explores these issues in the context of increased amounts, and larger chunks of, knowledge. Where the early general problem solvers accomplished generality and flexibility by working with a small number of atomic operators, adaptive planning works with increased amounts of knowledge and achieves these twin goals by exploiting the **structure** of that knowledge. Like the work on MACROPS [12], adaptive planning is concerned with larger chunks of actions, but adaptive planning extends their utilization to planning problems like the BART-NYC subway problem. Adaptive planning is concerned with **tasks** [6] and **commonsense planning** [13] problems. It is **knowledge-based** in that its approach to refitting old plans is based on the accessibility of the structure and content of the **background knowledge** associated with an old plan. As in the case of other knowledge-based planning approaches [8,14,15], adaptive planning is concerned with **control of access to knowledge**; its approach is dependent on the interaction of the planner's knowledge with the planner's current circumstances. Like the work on analogical planning [4,16,17], adaptive planning attempts to re-use old specific plans, but its strategies take greater advantage of the available knowledge, exploit categorization knowledge, and its processing is novel in that it takes the form of **situation matching**. Where other researchers have emphasized the problem of initial retrieval of old plans [18-21], the work on adaptive planning balances that view by investigating issues concerning flexibility and usage. Although knowledge acquisition is not the focus of the current research, adaptive planning does provide a framework for dealing with these issues. It promises to promote **additivity** because its procedures are largely based on the structure of the knowledge and not its content. Moreover, as a by-product of abstraction and specialization, PLEXUS discovers the generalizations over the steps of the old plan and the steps of the new plan, and consequently it provides a framework for the planner to do automatic re-organization and generalization [22-25].

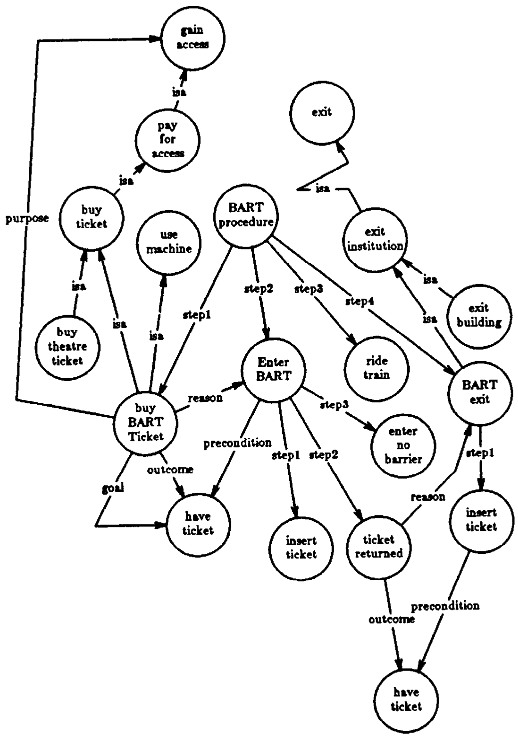

Figure 1: BART Plan with some background knowledge.

References

1. Alterman, R., Adaptive Planning: Refitting old plans to new situations, in *The seventh annual conference of the cognitive science society*, 1985.

2. Alterman, R., Adaptive Planning: Refitting old planning experiences to new situations, *Second Annual Workshop on Theoretical Issues in Conceptual Information Processing*, 1985.

3. Alterman, R., Adaptive Planning: A case of flexible knowing, Technical Report, University of California at Berkeley, 1986.

4. Carbonell, J. G., Derivational analogy and its role in problem solving, *AAAI-83*, 1983, 64-69.

5. Stallman, R. and Sussman, G., Forward reasoning and dependency-directed backtracking in a system for computer-aided circuit analysis, *Artificial Intelligence 9*, 2 , 135-196.

6. McDermott, D., Planning and Acting, *Cognitive Science 2* (1978), 71-109.

7. Hayes-Roth, B., A cognitive model of planning, *Cognitive Science 3* (1979), 275-310.

8. Davis, R., Meta-Rules: Reasoning about Control, *Arificial Intelligence 15* (1980), 179-222.

9. Kolodner, J. L., Reconstructive memory a computer model, *Cognitive Science* 7 (1983), 281-328.

10. Ernst, G. and Newell, A., *GPS: A case study in generality in problem solving*, Academic Press, 1969.

11. Fikes, R. and Nilsson, N., STRIPS: a new approach to the application of theorem proving to problem solving, *Aritificial Intelligence 2* (1971), 189-208.

12. Fikes, R., Hart, P. and Nilsson, N., Learning and Executing Generalized Robot Plans, *Artificial Intelligence Journal 3* (1972), 251-288.

13. Wilensky, R., *Planning and Understanding*, Addison-Wesley Publishing Company, 1983.

14. Wilensky, R., Meta-Planning: Representing and using knowledge about planning in problem solving and natural language understanding, *Cognitive Science 5* (1981), 197-233.

15. Stefik, M., Planning and meta-planning, *Artificial Intelligence 12*, 2 (1981), 141-170.

16. Carbonell, J. G., A computation model of analogical problem solving, *IJCAI 7*, 1981.

17. Carbonell, J. G., Learning by analogy: formulating and generalizing plans from past experience, in *Machine learning, and artificial intelligence approach*, Mitchell, M. C. (editor), Tioga Press, Palo Alto, 1983.

18. Kolodner, J. L. and Simpson, R. L., Experience and problem solving: a framework, *Proceedings of the sixth annual conference of the cognitive science society*, 1984.

19. Kolodner, J. L., Simpson, R. L. and Sycara-Cyranski, K., A process model of cased-based reasoning in problem solving, *Proceedings of the ninth international joint conference on artificial intelligence*, 1985.

20. Hammond, K., Indexing and Causality: The organization of plans and strategies in memory., Yale Department of Computer Science Technical Report 351, 1985.

21. Hendler, J., Integrating Marker-Passing and Problem Solving, in *The sevent annual conference of the cognitive science society*, 1985.

22. DeJong, G., Acquiring Schemata through Understanding and Generalizing Plans, *IJCAI 8*, 1983.

23. Schank, R. C., *Dynamic Memory*, Cambridge University Press, Cambridge, 1982.

24. Kolodner, J. L., Maintaining organization in a dynamic long-term memory, *Cognitive Science* 7 (1983), 243-280.

25. Lebowitz, M., Generalization from natural language text, *Cognitive Science* 7 (1983), 1-40.