Adaptive Planning: Refitting Old Plans to New Situations*

Richard Alterman

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

1. Introduction

This paper is about *adaptive planning*. The basic problem that adaptive planning addresses is the development of planning techniques for re-using old plans. The capability to re-use old plans suggests a way to avoid the problem of combinatorial explosion that is inherent in brute force planning methods. Moreover, it accounts for some of the flexibility of human planners: a planner than can re-use plans can plan about a wide range of phenomena, not so much because its depth of knowledge is consistent throughout that range, but because it can refit old plans to novel contexts.

A typical case of re-using plans is the situation when a planner is about to ride the NYC subway for the first time, and uses its experiences on BART (Bay Area Rapid Transit) to guide its planning activity. Consider the steps involved in riding BART (see figure 1). At the BART station I buy a ticket from a machine. Next, I feed the ticket into a second machine which opens a gate to let me into the terminal and then returns my ticket. Next I take the train. At the exit station I feed my ticket to another machine that keeps the ticket and then opens a gate to allow me 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. Given the BART Plan as depicted in Figure 1, there appears to be little in common between the two procedures. The problem with the BART Plan, as shown in figure 1, is that it represents this plan in isolation. In isolation the old plan does not provide enough information to refit it to the NYC subway situation. There is a great deal of background knowledge associated with the BART Plan that is not explicitly represented in the figure 1, but is needed in order to re-use the old plan. Without the background knowledge the BART Plan is practically useless in the construction of a plan for riding the NYC subway.

The key idea to understanding the adaptive planning approach to re-using old plans is:

> **Make explicit the content and organization of the background knowledge associated with the old plan.**

What is known about the BART Plan is not only the plan itself, but also that plan in relation to all the other planning knowledge that is available to the planner (see figure 2). Making explicit the background planning knowledge allows for a different kind of planner. Rather than planning by problem solving, it becomes possible to plan by *situation matching*. Rather than treating the old plan as a partial solution which is modified using weak methods, such as GPS (c.f. [1]), the old plan is used as a starting point from which the old and new situations are matched, and in the course of the matching a new plan is produced.

* 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.

** I would like to thank Robert Wilensky for getting me interested in the problems of commonsense planning and the role of memory in planning and some good discussion as the ideas developed. I would also like to thank the other members of the BAIR (Berkeley Artificial Intelligence Research) group for constructive input.

An adaptive planner called PLEXUS (see dictionary for explanation of name) has been constructed. PLEXUS' knowledge-base takes the form of a network. Its basic strategy is to match the old plan situation against the new context. Where differences occur, changes are made to the old plan. PLEXUS achieves both the detection of differences and subsequent changes to the old plan by exploiting the net that surrounds the old plan.

2. Related Work

Early artificial intelligence research on planning and problem solving emphasized weak methods that apply to situations where extensive knowledge is not available. The basic idea was to produce plans by manipulating and combining many low level operations. Typical areas of application were chess playing programs, theorem proving, and robot problem solving. Although these kinds of methods are appropriate in knowledge poor contexts, they suffer from the problem of combinatorial explosion and are therefore less than ideal in the knowledge rich domains that frequently occur in common-sense planning situations.

Macrops [2] was the first attempt to deal with the problem of re-using old plans. Their application domain was robot problem solving. Solutions to old problems were generalized by substituting variables for the arguments of some of the operators. During planning, if some portion of an old plan, in its generalized form, achieved some goal or subgoal of the new plan it was re-used.

The major limitation on macrops was that the goals and situation of the old plan, except for where variables were substituted for constants, had to identically match the new situation. For most real world problems it is rarely the case that the old and new plan situations, or some subportion of them, are identical. More recent developments in the area of re-using plans have attempted to increase the flexibility of this original approach, in terms of both plan retrieval and usage. Kolodner & Simpson [3-5] and Hammond [6] have proposed techniques, based on Schank's theory of Dynamic Memory [7], for indexing old plans. Georgeff [8] has suggested guidelines for variable substitution under analogical conditions.

Carbonell has developed a theory of what he refers to analogical problem solving. He has suggested two approaches to employing analogies. His first approach [1,9] applies a means-ends analysis to an old problem, gradually transforming it into a solution for a new problem. His second approach [10] dubbed derivational analogy, attempts to recreate the decision making process of relevant past problem solving situations, and apply that decision making process to the new problem situation.

Both of these approaches suffer from the problem of dealing with a plan in isolation. As alluded to in the introduction, without the content and organizational structure of the background knowledge the old plan will be practically useless in the construction of a new plan. Moreover there are more specific problems associated with each of these approaches. A problem with the first approach is the dependence of means-ends analysis on the creation of effective difference tables. A problem for the second approach is that in many cases a derivational history is not available either because it has been forgotten or because the plan was learned by rote and therefore a derivational history never existed.

3. The Plan Network

In adaptive planning, planning knowledge is represented in the form of a network. Associated with every plan in the network are a set of conditions which are used to determine if a plan, or a step in a plan, is appropriate for the current planning situation. Conditions include intentions, (sub)goals, pre-requisites, and expected outcomes. Adaptive planning is largely based on situation matching; the conditions provide a checklist for the planner to use in determining the applicability of an old plan, or step, to a new situation.

Each plan is represented as a sequence of steps, and, recursively, each step is a plan which, in principle, can be decomposed into a sequence of steps. For example, steps in the BART Plan include: buying a ticket, entering a BART station through a turnstile, riding the train, and exiting a BART station through a turnstile. Furthermore, there are substeps involved in buying a ticket: putting money into the machine and receiving a ticket in return.

There is an 'isa hierarchy' in the network, and as is the norm properties are inherited through the 'isa hierarchy'. For example, there is a general plan for buying tickets, and there are at least two specialized versions of that plan:

buying a ticket from a machine and buying a ticket from a teller. Moreover both of these descendants inherit from their common ancestors the normal goal for buying a ticket, which is to gain access to some service. Finally, as mentioned above, associated with each plan (or plan step) are a list of conditions. These can all be inherited as well.

4. The Adaptive Process

4.1. An overview

Having recalled a plan, PLEXUS must use the embedding context of the old plan in order to refit it to meet the demands of the new situation. PLEXUS refits the plan by using the conditions to match the old planning situation to the new one. In the event that differences occur, it is necessary for PLEXUS to modify the old plan to meet the new situation.

Some of the differences between old and new are *anticipated* at the outset, others grow out of the planner's *interaction* with the environment. For example, in the case where PLEXUS is adapting the BART Plan to the situation at the NYC subway, an anticipated difference is that the planner no longer expects to be purchasing a BART ticket, but instead a ticket for the NYC subway. A difference that occurs as a result of the planner's interaction with the environment is the realization that, for the NYC subway, one does not buy a ticket from a machine, but instead from a teller.

The basic procedure works as follows: PLEXUS adapts, in order, one step at a time, the steps of the old plan. For each step, it either anticipates or interacts with the environment to determine if the conditions associated with that step are met by the current environment. If the conditions of that step are met it adapts in a depth-first fashion the subsequences of that step. When PLEXUS bottoms out it moves on to the next step. If it can't apply a step of the old plan it tries to find an *abstraction* of that step which will work in the current context. If it succeeds in finding an abstraction, it next attempts to find a *specialization* of that step that will work in the current context. Whether in its abstraction phase or its specialization phase, PLEXUS is exploiting both the content and organization of the background knowledge associated with the old plan.

More specifically, the old plan is adapted to the new situation as follows:

1) Check the conditions of each step in the old plan. The conditions of a step include, via property inheritance, all of the conditions of the 'isa' ancestors of that step.
2) If the conditions are met then apply that step to the current situation.
3) If at least one of the conditions fails, try a more abstract version of that step by moving up the 'isa' hierarchy.
4) If a more abstract version of an old step works, try to specialize that step by moving back down the 'isa' hierarchy.
5) For each step, if the conditions of that step are met apply the same procedure in a depth-first fashion to the substeps of that step. When the procedure bottoms out move on to the next step.

In a sense PLEXUS' working hypothesis is that the new plan under construction and the old plan share a *significant* ancestor in the 'isa' hierarchy. By significant I mean that the steps of both the old plan and the new plan can be seen as specializations of the shared ancestor's steps. In fact, as the new plan is constructed, PLEXUS discovers the steps of the more abstract plan. These steps are composed of the successfully borrowed steps of the old plan (2) and the successful abstractions of steps from the old plan that failed to apply in the new situation (3,4).

Consider what happens when PLEXUS uses the BART Plan as a basis for constructing a plan for riding a NYC subway. Figure 2 shows a portion of net that is relevant to the problem of adapting the first step of the BART Plan to the problem of riding a NYC subway. The first step of the BART Plan is to buy a ticket from a machine, but in the new situation there is no machine from which to buy a ticket. By moving up the 'isa' hierarchy it finds a more abstract version of this step that will work in the current situation: through abstraction PLEXUS determines that because there is a place for buying a ticket the plan for buying a ticket will work. Next it moves back down the 'isa' hierarchy, specializing the plan to a plan for buying a ticket from a ticket office. In a similar fashion PLEXUS adapts the other three steps of the BART Plan. The following two sections will describe in greater detail what is involved in the processes of abstraction and specialization.

4.2. Abstraction - Some issues

This section will discuss several critical issues involved in abstraction. Associated with each of these issues is an answer that will fall out from the explicit introduction of the content and organization of the background knowledge.

First there is the issue of how to choose the correct abstraction. A given plan step can have any number of abstractions associated with it. Choosing the wrong abstraction can lead to the wrong action. For example, one abstraction of 'buying a ticket from a machine' is to 'use a machine' and a specialization of this is to 'use a candy machine' (see figure 2). In the NYC subway situation the planner needs to find the right abstraction of 'buying a BART ticket' else it may substitute 'buying a candy bar from a machine' for 'buying a ticket from a machine'. The planner can avoid this problem by ascending the 'isa' hierarchy that maintains the purpose of the step in the plan that is being refitted. In this case that means moving up the 'isa' hierarchy towards 'gain access'.

A second issue concerns knowing when to look for an alternate version of a step. For example, suppose the planner is trying to adapt its BART Plan to the Washington D.C. subway system. Like BART on the D.C. metro tickets are bought from a machine. Suppose the planner tries to buy a ticket from a machine but its dollar bill is rejected because it is too crumpled, the planner should not abandon the step to buy a ticket from a machine. By making the background knowledge explicit it becomes apparent why the step is not abandoned (see figure 2). The condition that is failing is not associated with 'buying a ticket from a machine', but instead with one of its substeps 'insert dollar bill'. The point is that the problem is not with the step but with the substep and it is the substep that needs to be refitted.

A third issue concerns the problem of when a planner should stop abstracting. In the NYC subway situation, when the planner discovers there is no ticket machine, the right abstraction to move to is 'buy a ticket'. In the case where the planner is trying to ride BART and the problem is s/he does not have any money, the planner needs to move above 'pay for access' to the abstraction 'gain access', thus allowing for the specialization 'break in' (see figure 2). Again the explicit introduction of the content and organization of the background knowledge allows for a simple solution: the planner needs to move up the 'isa' hierarchy to a position above the condition that is failing. In the case of the NYC subway situation the failing condition is that there exists no ticket machine, which is associated with 'buying a ticket from a machine', but not 'buying a ticket'. In the case of the BART situation the failing condition is associated with 'paying for access' but not 'gain access'. In either case the solution is to move to an abstraction that has no failing condition, but has a son with one.

A fourth way in which the structure of the background knowledge aids adaptation is that it partially orders alternate versions of a failed step. For example, 'gain access' is a more abstract version of 'buying a ticket from a machine' then is 'buying a ticket'. Consequently, in the NYC subway situation 'buying a ticket from a teller' will be available as an alternate plan before 'breaking in'.

5. Specialization - Some Issues

Where abstraction is initiated by the expectations formed from the old plan and driven by the planner's observations, specialization is not necessarily initiated by expectations. Again consider the case of adapting the BART Plan to the NYC subway situation. The expectation formed from the old plan is that there will be a machine from which to buy a BART ticket. This expectation is immediately amended, due to anticipated differences, to an expectation that there will be a machine from which to buy a NYC subway ticket. So the planner looks for a ticket machine, none exists, and therefore it continues abstraction. For present concerns, the key point is that when the planner observes its environment it is looking for a particular object, in this case a ticket machine. Now contrast that to the situation that confronts the planner when it attempts to form a specialization of the abstraction it has just determined. In the case of buying a subway ticket, through the process of abstraction, the planner determines that it still wants to buy a ticket, but there is no machine from which to buy it. The point is that specialization is not necessarily initiated by an expectation generated from the planner's memory. It is entirely possible that the planner looks around and notices a ticket booth, which thereby initiates the specialization of the plan 'to buy a ticket' to a plan 'to buy a ticket from a ticket teller'. On the other hand, it is equally possible that the planner first forms an expectation that a ticket booth might exist and then it looks for it.

Such considerations should suggest that the movement towards specialization is a more complicated blend of expectation and observation than in the case of the movement towards abstraction. Rather than applying an absolute rule for initializing specialization by one or the other means, it appears that the planner must deal with specialization on a case by case basis. However, it is possible to provide a criteria for choosing one method of initiation over the other. For example, it could be the case that an alternative is strongly suggested by the type of failure that occurs. So if a planner is trying to get change from a change-machine and the dollar is returned because it is to crumpled, an alternate plan is strongly suggested by the nature of the failure (i.e. flatten out the dollar bill and try again). There are also cases where the expectation that a particular alternate plan might work is dictated not so much by the nature of the failure, but rather by a bias towards the normal specialization of a particular abstraction. Such I believe is the case of buying a ticket for the NYC subway. When the plan to buy a ticket from a machine fails and the planner determines that it still wants to buy a ticket, the planner is biased towards the normal plan for buying a ticket which is to buy ticket from a ticket teller. On the other hand, when no alternative stands out, the value of observing the surrounding environment for clues greatly increases. Overall the planner's strategy is to try a likely specialization if it is somehow suggested by circumstances, and if that fails observe the planning environment for clues.

6. Summary and Conclusions

Adaptive planning is a novel approach to planning. As opposed to planning by weak methods, its basic procedure is to recall similar planning situations and refit them to meet the demands of the current context. The basic idea is that the network of relationships in which a planning procedure is embedded can be exploited to refit an old plan to a new situation. The differences in the current planning situation and the context associated with the old plan guide the adaptive process. By exploiting the content and organization of the background knowledge it becomes possible to plan by situation matching.

7. References

1. Carbonell, J. G., A computation model of analogical problem solving, *IJCAI 7*, 1981.
2. Fikes, R., Hart, P. and Nilsson, N., Learning and Executing Generalized Robot Plans, *Artificial Intelligence Journal 3* (1972), 251-288.
3. Kolodner, J. L., Maintaining organization in a dynamic long-term memory, *Cognitive Science 7* (1983), 243-280.
4. Kolodner, J. L., Reconstructive memory a computer model, *Cognitive Science 7* (1983), 281-328.
5. 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.
6. Hammond, K., Indexing and Causality: The organization of plans and strategies in memory., Yale Department of Computer Science Technical Report 351, 1985.
7. Schank, R. C., *Dynamic Memory*, Cambridge University Press, Cambridge, 1982.
8. Georgeff, M. P., Strategies in Heuristic Search, *Artificial Intelligence 20* (1983), 393-425.
9. Carbonell, J. G., Derivational analogy and its role in problem solving, *AAAI-83*, 1983, 64-69.
10. 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.

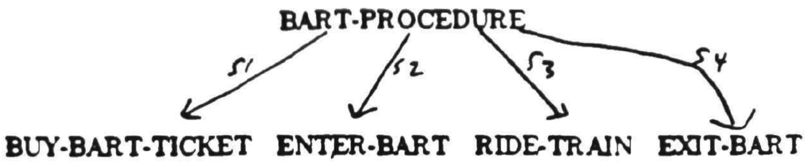

Figure 1. BART Plan in Isolation.

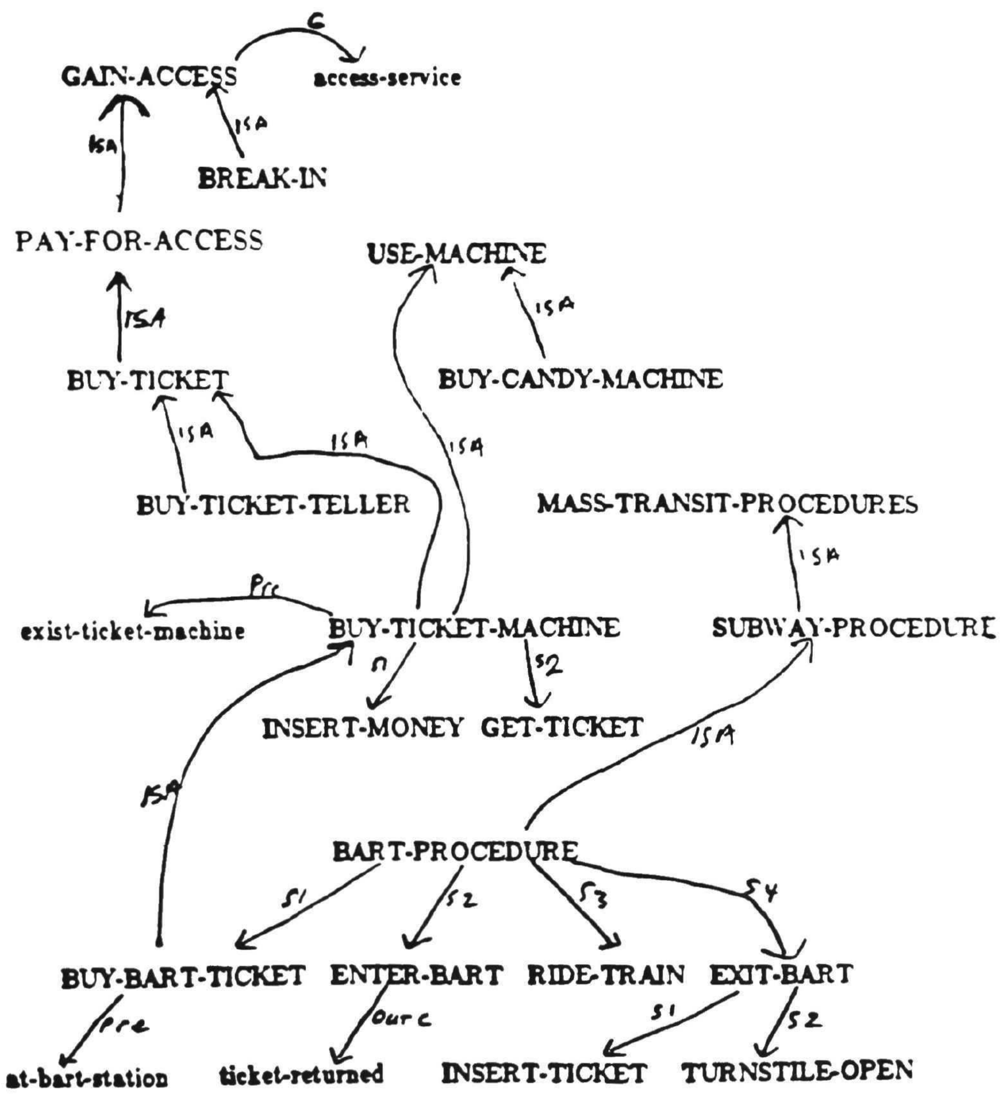

Figure 2. BART Plan with Background Knowledge Explicit.