Recognizing Novel Uses for Familiar Plans

Beth Adelson[1]
Dept. of Computer Science
Tufts University
Medford, MA 02155
adelson@tufts.cs.edu

Abstract

Analogical design and invention is a central task in human cognition. Often during the process the designer/inventor gets stuck; backs off from the problem; and only later, after having put the problem aside, discovers that some familiar plan can be used in a novel way to solve the problem. We describe a system which uses a causal *case memory* to check the side effects, preconditions, etc. of incoming events in order to model this phenomenon.

The method used makes this work relevant to case-based reasoning as well as design. It also forms a companion issue to execution-time planning.

1 Motivation

Analogical design/invention is a task which arises continually. Sometimes the task occurs in a form so simple as to pass almost unnoticed, as when picnickers find they have forgottten their knives. Sometimes the task requires intense and extended effort, as when computer engineers find they must develop a new architecture. Difficulty notwithstanding, the scenario often runs as follows: the designer finds that the tack he has been taking is not producing the desired solution; finding he is stuck, he puts the problem aside; he then recognizes in some later situation, that a familiar device, instrument or plan can be used in a novel way to solve the problem.

Problem Statement:
This scenario of the design process implies that cognitively based design systems need a mechanism that allows the recognition of situations that can contribute to the attainment of goals which previously could not be fulfilled.

The problem we are addressing is related to the problem of *execution-time planning*, in which a planning system needs to be able to recognize conditions that unblock previously blocked goals even though the system cannot predict when those conditions will occur (Hammond, 1989; Alterman, 1988; Firby, 1987; Georgeff & Lansky, 1987; Simmons & Davis, 1987). Hammond describes such a system in his work on *opportunistic memory* (1989): His system

[1]This work was funded by Carnegie-Mellon's NSF funded EDRC and by a grant from NSF's Engineering Directorate.

contains a detailed taxonomy of the conditions under which goals which were blocked for various reasons might later be fulfilled. As a result, under Hammond's model when a goal is blocked it is suspended and memory is tagged for the conditions which might allow its satisfaction. When these conditions are later encountered, becuase of the tags placed in memory, they are recognized as unblocking conditions and the goal is reactivated[2] and then pursued.

Design systems face a somewhat different problem. They need to recognize conditions that will contribute to goal satisfaction even though they can predict neither when the conditions will occur, nor what the conditions will be. In this paper we describe the mechanism we have developed for addressing this recognition problem. The mechanism is one that considers the implications (side effects, etc.) of well-known plans in order to discover their relevance to blocked goals. The effect of the mechanism is that the system is able to recognize novel uses for familiar plans.

2 The Need for Recognizing Plans During Design

2.1 Previous work: Debugging analogically acquired plans

We are developing our recognition mechanism in the context of our work on analogical learning and analogical design. We began by concentrating on analogical debugging. This work then led us to our current concern with recognition.

Our initial system (Adelson, 1989a&b) was developed to acquire plans for programming operations such as **pop**, **push** and **sort**. The plans took the form of executable models. This allowed the plans to be used in either of two ways: They could be run for purposes of debugging; or they could drive a module that generated both code and box-and-arrow drawings (Adelson, 1989a&b; Burstein & Adelson, 1987; Burstein and Adelson, in press).

As described on page 3, when the plans were being used for debugging they were passed to a mechanism that, via simulation, identified functionally analogous plan entities in the more and less familiar base and target domains. This entailed addressing: 1. The role of causal reasoning; and 2. The use of target domain knowledge in analogical learning.

When the plans were being used for debugging they were passed to a mechanism that, via simulation, identified entities in the target domain that were functionally analogous to entities in the original base domain plan[3]. This entailed addressing: 1. The role of causal reasoning; and 2. The use of target domain knowledge in analogical learning.

In addition to producing programming plans, the system has modeled Edison's description of modifying the phonograph to produce the kinetoscope and Morse's account of developing

[2]Hammond's model has the advantage of allowing goals to be quiescent during their suspension. Additionally, because of its detail, Hammond's indexing scheme results in both a high rate for hits, and a low rate for false alarms.

[3]The base and target domains are the more and less familiar domains. The reason for identifying functionally analogous entities is that one purpose of analogical debugging is to find target domain entities that should replace base domain entities in order to produce a model appropriate to the target domain.

telegraphic relay stations (Brian, 1926; Okagaki & Koslowski, 1987)[4].

Example: Morse's telegraphic relay station

The debugging behavior we have previously focussed on and the recognition behavior we turn to here are illustrated by the following example (Okagaki & Koslowski, 1987):

> Initially, in trying to transmit telegraphic signals across significant distances Morse tried the strategy of building successively stronger generators. He found however, that the signals still degraded with distance. The solution to the problem came to him in the following way. While riding on a train, he happened to look out of the window and notice a Pony Express depot, at which horses were being fed and watered. Morse realized that the *relay station* strategy constituted a solution to the telegraph problem as well.

There are two interesting pieces of reasoning in this example, recognizing that the Pony Express offers a useful analogical model; and then modifying the Pony Express model to fit the telegraphy domain. First we describe the modification performed by our system's debugger (Adelson, 1989a & in press) and then we go on to describe our recognition mechanism.

The debugger

In modifying the Pony Express model to fit the telegraphy domain, the system determines, via simulation (Schank & Riesbeck, 1989; Falkenhainer, Forbus & Gentner, 1988; Hammond, 1989a; Simmons & Davis, 1987), the causal effects produced by the use of horses traveling between relay stations (the information in the message can successfully be sent a long distance as a result of the medium of conveyance being repeatedly refreshed at each leg of the journey). In an attempt to construct a mechanism that will reproduce this effect in the target domain, the system again uses simulation to identify already existing appropriate pieces of mechanism in the target domain; ones that could be used in a larger schema for repeatedly refreshing and sending information short distances. This results in the system producing a model in which the existing generators are arranged in series at appropriate intervals.

3 Recognizing the analogy

As mentioned above, in order to produce the model of generators arranged in series, Morse must first recognize that the relay station strategy employed by his competitors can serve as a model for the telegraph problem. We propose a mechanism that accounts for this class of recognition phenomenon, in which it is noticed that a familiar case in memory; a plan, operation or device, can serve a purpose that is radically different from those which it has previously served. In essence, the mechanism is one in which the changing environment is examined for opportunities which allow the fulfillment of blocked goals (Birnbaum & Collins, 1984; Birnbaum & Selfridge, 1981). In this scheme, a blocked goal is placed on a list with other goals that have also been suspended because they were blocked. The system then sets about to parse any incoming plan in order to determine if the plan can unblock it[5].

[4]Although these may not be completely accurate accounts of particular discoveries, they persist because they are accurate accounts of our *experience* of the discovery process.

[5]Along with others any others already on this list.

At a very general level, under this model Morse realizes that the relay station strategy will meet his need because as he backs off from his plan of building stronger generators he starts looking at the features of other plans and assessing their utility in unblocking his goal. That is, when he encounters the Pony Express station he notices that it has the effect of sending messages long distances and so recognizes it as a plan that meets his goal.

The next section provides a detailed description of how this phenomenon is captured in our system. We begin with an overview of the system's function.

4 The Recognition Mechanism

4.1 System Function

Our system is designed to monitor the environment for plans which will be useful in satisfying blocked goals. In doing so the system takes the following steps:

1. Detecting and saving blocked goals:
   When the system is unsuccessful in meeting a goal, the goal is placed on a BLOCKED GOAL LIST.

2. Monitoring the environment for opportunities:
   The system then begins to evaluate events in the environment[6] in terms of their relevance to the blocked goal. In order to do so the system uses an EVENT LEXICON which can be thought of as a memory for causal relations. The EVENT LEXICON stores features of previously encountered events such as intended effects, side effects, etc.

   Because the system is looking at these effects to see if they fulfill a blocked goal, and therefore should be reproduced, each effect has a pointer to a plan which if run would give rise to the effect. That is, an event may produce a side effect which matches the desired effect of the blocked goal and if it does the system can reproduce the side effect in order to meet the blocked goal.

   The monitoriing process has two parts:

   (a) Parsing incoming events:
   When an event occurs, the system retreives the representation of that event from the EVENT LEXICON, instantiates it and places the instantiated representation in the system's short term memory.

   (b) Testing potential opportunities:
   The system now matches the features of the retrieved event representation against the blocked goal in order to see if some feature of the event will help in unblocking the goal (Hammond, 1989). If a match is found, the plan indexed under the feature is retreived.

[6] Events are provided as input to the system.

3. Seizing the opportunities:
At this point the goal is removed from the BLOCKED GOAL LIST. The plan satisfying the goal can now be instantiated and run. If the plan, when run, meets the goal it will be indexed under the goal in the system's PLAN MEMORY.

4.2 The Morse Example

In the Morse example this translates into the following steps:

1. Detecting and saving blocked goals:
The system is given the goal shown just below, of sending a telegraph message over a long distance:

```
(GOAL: send
       (OBJECT:  message
(TYPE: telegraph signal))
       (DISTANCE: >1,000 mi.))
```

The system finds that the goal cannot be satisfied by any of the existing plans in the system's PLAN MEMORY. The goal is therefore placed on the BLOCKED GOAL LIST.

2. Monitoring the environment for opportunities:
The system is then given a set of inputs representing the objects and events Morse encounters during the train ride: The indifferent dinner in the dining car, the elaborately feathered bonnet of his compartment mate, and the Pony Express relay station which he sees as he looks out the window[7].

   (a) Parsing incoming events:
   The system retreives its representation of the events from its EVENT LEXICON. The representations are instantiated and placed in the system's short term memory.

   (b) Testing potential opportunities:
   The system then finds that SIDE EFFECT1 in the representation for the Pony Express matches the blocked goal.

   Representation for Pony Express:

```
(EVENT: Pony Express
(GOAL: deliver mail)
(PRECONDITIONS: (...)...(...))
(SIDE EFFECTS:
       (SIDE EFFECT1: send
         (OBJECT:  message
```

[7]Not all inputs are given equal attention. We discuss the differences in attention given to different inputs in the context of the system's attention mechanism in Section 4.3 below.

```
(TYPE: mail))
                    (DISTANCE: >1,000 mi.)
                    (CAUSED-BY-PLAN: relay stations)))

(INTENDED EFFECTS: (...)...(...)))
```

That is, looking at the sbove EVENT LEXICON representation of the Pony Express, the system determines that through the use of relay stations, the Pony Express has the effect of sending messages across long distances.

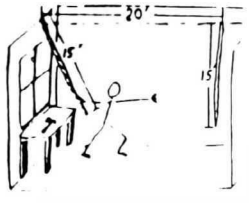
3. Seizing the opportunities:
   The system retreives the RELAY STATION plan indexed under SIDE EFFECT1 and passes it to the system to be run[8]. The plan is found to satisfy the goal and so it is indexed under the goal in the system's PLAN MEMORY.

4.2.1 Range of Examples: Einstellung and Archimedes

Our system has also been used to model a result representative of the results obtained in the classic gestalt *einstellung* experiments.

The example is as follows (Meyer, 1977):

Experimental Situation: A subject is shown into an otherwise bare laboratory, in which two cords are suspended from a ceiling and a hammer has been left on the windowsill. The subject is asked to tie the ends of the two cords together, however the distance between the cords is great enough that the ends of the cords cannot be brought together simply by grasping the first one and walking towards the second.



Result: The problem can be solved by tying the hammer to the end of the first string and then using the hammer as a pendulum to swing the first string towards the second.

Often subjects do not hit upon the solution themselves, In these instances the experimenter walks by one of the strings and 'accidentally' sets it in motion. This hint invariably allows the subject to solve the problem.

Our noticing mechanism, in modelling the above result, offers an account of the processes behind the insights which occur in these experiments. Under our model, at the outset of the experiment no plan for tying ropes which are more than an arm's length apart resides in memory. The subject gets stuck and the goal is placed on the BLOCKED GOAL LIST. When the rope swing occurs an underlying representation of the event is retreived. This is where the insight occurs; the representation of the rope swing contains bringing the ropes together as a side effect. Because this side effect matches the desired effect the rope swing is perceived as a solution to the problem. The retrieved representation also points to the

[8]Because the retreived plan is an analog to the one actually being sought it will be modified by the system's debugger before it is run.

'pendulum plan' which can reproduce the effect. When the plan is retreived, the hammer is seen as an appropriate weight for the pendulum.

This account includes an explanation of how in these situations problem-solvers come to see the environment differently after they have backed off from the problem-solving (Gick & Holyoak, 1980, 1983).

As a third example, and one which also illustrates the issue of attention, our system models Archimedes' realization that he could calculate the volume of the king's irregularly shaped crown by submerging it in water. Initially, because the system only has plans for calculating volumes of regularly shaped objects, the system places the goal of 'calculating the volume of the irregularly shaped crown' on the BLOCKED GOAL LIST. The system is then given 'Archimedes' bath' as an input event. None of the side effects of bathing: the cooling of the water, the increase in the volume of the tub's contents, etc., match the blocked goal of 'calculating the volume of the irregularly shaped crown'. However, the side effect of the increase in the volume of the tub's contents is treated as promising because it is relevant to the blocked goal's concern with volume. The system therefore pays more attention to it in the form of further processing: The system looks in EVENT MEMORY for the effects of increasing the volume of the tub's contents. It finds that one side effect is that the volume of the object that causes the increase can be obtained by calculating the difference between the new volume and the original. This side effect matches the blocked goal and is treated as a solution to the problem.

4.3 The attention mechanism

What attentional phenomena do we want a discovery system to capture and how should we implement the mechanisms giving rise to them? We have addressed this problem to a limited extent. At minimum we would would want to explain: Why Orville Wright but not Samuel Morse would notice a feathered bonnet; and why Morse certainly and Wright only perhaps would notice the Pony Express[9]. Although it is for a different reason, we also want to explain why Archimedes would notice the implications of the rise in water level but would not bother to trace the implications of the water cooling. In summary, we want to explain the fact that things which are either psychologically charged or conceptually relevant may receive more attention.

In order to model these effects we have implemented the system so that it does more retrieval if an event looks relevant to a blocked goal in either of these ways. That is, if an event mentions a competitor or a mentor the features not only of the event, but also, if needed, of the event's effects will be retrieved from memory and examined by the system. Additionally, as illustrated in the Archimedes example, if a feature of an event is conceptually relevant to the blocked goal the features of the feature will be retrieved and examined. As a result, we have made a first pass at attentional issues.

[9]At the time the Pony Express was Morse's competition.

5 Summary and Relevance to other Research

Analogical design and invention is a central task in human cognition. In our system a causal *case memory* is used to check the side effects, preconditions, etc. of incoming events, this allows the system to put familiar plans to use in novel ways. This can account for the reasoning that often is needed in analogical design and invention.

In addition to the relevance of our work to case-based reasoning, our design task is a companion task to execution time-planning. In execution-time planning, a planning system needs to be able to recognize conditions that unblock previously blocked goals even though the system cannot predict when those conditions will occur (Hammond, 1989; Hammond, Converse, & Marks, 1988; Kolodner, Simpson, & Sycara-Cyranski, 1985). Design systems need to recognize conditions that will contribute to goal satisfaction even though they can predict neither when the conditions will occur, nor what the conditions will be. It is our hope that the mutual relevance of these tasks will motivate work on the integration of design, cased-based reasoning and planning systems.

6 References

Adelson, Beth. Cognitive modeling: Uncovering how designers design. *The Journal of Engineering Design.*. Vol 1,1. 1989.

Adelson, B. The role of model-based reasoning in analogical learning. *Proceedings of the IJCAI-89 Workshop on Model-Based Reasoning.*

Adelson, B. Characterizing the nature of analogical reasoning. In Design Theory and Methodology. M. Waldron (Ed.). Springer-Verlag. NY. In press.

Alterman, R. Adaptive planning. *Cognitive Science*, Winter, 1988.

Birnbaum, L. and Selfridge, M. In *Inside Computer Understanding.* 1981. Erlbaum, Hillsdale:NJ.

Birnbaum, L. and Collins, G. Opportunistic Planning and Freudian Slips.

Brian, George. Edison: *The man and his work.* Knopf: Garden City, NY. 1926

Burstein, M. and Adelson, B. Mapping and Integrating Partial Mental Models. *Proceedings of the Tenth Annual Meeting of the Cognitive Science Society*, 1987.

Burstein, M. and Adelson, B. Analogical Reasoning for Learning. in *Applications of Artificial Intelligence to Educational Testing.* R. Freedle (Ed.) In press. Erlbaum: Hillsdale, NJ.

Firby, R.J. An investigation into reactive planning in complex domains. In: *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, Washington, July 1987.

Georgeff, M.P.& Lansky, A.L. Reactive reasoning and planning. In: *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle, Washington, July 1987.

Hammond, K. Opportunistic Memory. *Proceedings of the Machine Learning Workshop*, 1989.

Hammond, K. *Case-based planning: Viewing planning as a memory task.* Academic Press, Cambridge, Massachusetts, 1989.

Hammond, K., Converse, T., & Marks, M. Learning from opportunities: Storing and reusing execution-time optimizations. In: *Proceedings of the Seventh National Conference on Artificial Intelligence*, St. Paul, Minnesota, August 1988.

Kolodner, J. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society.* Boulder, CO: Cognitive Science Society, 1985.

Kolodner, J.L., Simpson, R.L., & Sycara-Cyranski, K. A process model of case-based reasoning in problem solving. In: *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, California, August 1985.

Okagaki, L. and Koslowski, B. Another look at analogies and problem-solving. *Journal of Creative Behavior*, 1987, *21*, 1.

Schank, R. and Riesbeck, C. *Inside Computer Understanding.* Erlbaum: Hillsdale, NJ. 1981.

Simmons, R., & Davis, R. Generate, test, and debug: Combining associational rules and causal models. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987.