A Model for Planning in Everyday Situations*

Robert Wilensky

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

1. Introduction - Four Tenets for a Theory of Planning

Much previous work on planning and problem solving has been concerned with either very specialized systems or with highly artificial domains (e. g., consider Fikes and Nilsson (1971), Newell and Simon (1972), Sussman (1975), Shortliffe (1976)). More recently, there has been an increase in attention given to planning in commonplace situations. For example, Rieger (1975) has proposed a set of "common sense algorithms" for reasoning about everyday physical situations; Hayes-Roth and Hayes-Roth (1979) are concerned with how a person might schedule a day's activities; and Carbonell (1980) POLITIC's program reasons dogmatically about political decisions. On another front, Sacerdoti (1977) and McDermott (1978), while operating perhaps in the more traditional problem solving context, have proposed some powerful approaches to problem solving in general.

1.1. Everyday Planning is Reasoning about Interactions Between Goals

We have been developing a theory of planning that is concerned with reasoning about everyday situations. A central tenet of this theory is that most of the planning involved in everyday situations is primarily concerned with the interactions between goals. That is, planning for individual goals is assumed to be a fairly simple matter, consisting primarily of the straightforward application of rather large quantities of world knowledge. The complexity of planning is attributed to the fact that most situations involve numerous goals that interact in complicated ways.

Thus while traditional problem solving research has been concerned with finding the solution to a single, difficult problem (e. g., finding the winning chess move), most everyday problem solving consists of synthesizing solutions to fairly simple, interacting problems. For example, a typical everyday situation that involves the sort of planning we are interested in might be to obtain some nails, and also buy a hammer. The plan for each goal is straightforward: One simply goes to the hardware store, buys the desired item, and returns. The problem lies in recognizing that it is a terrible idea to execute these plans independently. Rather, the seemingly simple common sense plan is to combine the two individual plans, resulting in the plan of going to the hardware store, buying both items, and then returning.

Simple as this situation may be, most conventional planners are ill-equipped to handle it. Although some planning programs have mechanisms for removing redundancies from a plan, they generally lack a mechanism for even noticing this sort of interaction if these plans are derived from heretofore unrelated goals. Perhaps more importantly, the interaction between plans may have more complex ramifications. For example, if enough items are to be purchased at the hardware store, then a better plan might be to take one's car, while walking may do otherwise. Thus a good part of planning involves detecting the interactions between goals, figuring out their implications, and then deciding what to do about them.

1.2. Planning Knowledge Should be Equally Available for Understanding

The second tenet of our theory of planning is that it should be equally usable by both a planner and an understander. That is, while a planner uses its planning knowledge to bring about a desired state of affairs, an understander may need to use this same knowledge to comprehend the actions of a person it is watching or of a character about whom it is reading. For example, a planner with the goal of keeping fit might take up jogging; an understander might use the same knowledge to infer that someone who has taken up jogging may have done so because he had the goal of staying in shape. Planning and understanding are rather different processes, and this will of course be reflected in our planning and understanding mechanisms. However, our theory of planning specifies that knowledge should be represented in a fashion so that it is usable by either mechanism.

1.3. Meta-Planning is Used as the Driving Principle

The third salient feature of our theory is that it is based on *meta-planning*. By this I mean that the problems a planner encounters in producing a plan for a given situation may themselves be formulated as goals. These "meta-goals" can then be submitted to the planning mechanism, which treats them just like any other goals. That is, the planner attempts to find a "meta-plan" for this meta-goal; the result of successful application of this plan will be the solution to the original planning problem.

A typical example of a meta-goal is the goal RESOLVE-GOAL-CONFLICT. A planner would presumably have an instance of this goal whenever it detects that some of its "ordinary" goals are in conflict with one another. The meta-plans for this goal are the various goal conflict resolution strategies available to the planner.

Meta-planning is described in more detail in Wilensky (1980). Here, we give only a brief characterization of its main features and advantages.

Meta-goals are organized by *meta-themes*. These are very general principles of planning that describe situations in which meta-goals come into being. We summarize these briefly:

Meta-themes

1) DON'T WASTE RESOURCES
2) ACHIEVE AS MANY GOALS AS POSSIBLE
3) MAXIMIZE THE VALUE OF THE GOALS ACHIEVED
4) AVOID IMPOSSIBLE GOALS

*Research sponsored in part by the National Science Foundation under grant MCS79-06543 and by the Office of Naval Research under contract N00014-80-C-0732.

As an example of how these function, the meta-theme "ACHIEVE AS MANY GOALS AS POSSIBLE" is responsible for detecting goal conflicts. That is, if the planner intends to perform a set of actions that will negatively interact with one another, this meta-theme causes the planner to have the goal of resolving the conflict. If this meta-goal fails, i. e., the planner could not find a way to resolve the conflict, then the meta-theme "MAXIMIZE THE VALUE OF THE GOALS ACHIEVED" springs into action. This meta-theme sets up the goal of arriving at a scenario in which the less valuable goals are abandoned in order to fulfill the most valuable ones. The details of the meta-plans involved in these processes are described in length in the last two sections of this paper.

Meta-planning has a number of advantages over other approaches to planning; these advantages are summarized below:

1.3.1. Meta-planning knowledge can be used for both planning and understanding

As meta-goals and meta-plans are declarative structures in the same sense as are ordinary goals and plans, they may be used to understand situations as well as plan in them. Thus an understander with access to this knowledge would be able to interpret someone's action as an attempt to resolve a goal conflict, for example. In contrast, planning programs that have the equivalent knowledge embedded procedurally would not be able to conveniently use it to explain someone else's actions.

1.3.2. The same planning mechanism can apply to more difficult tasks.

Meta-planning knowledge generally embodies a set of strategies for complicated plan interactions. By formulating this knowledge in terms of goals and plans, the same planning architecture that already exists for simpler planning can be used to implement more complicated planning involving multiple goals, etc.

1.3.3. More general resolution of traditional planning problems

Traditional planners usually treat problems such as goal conflicts by special purpose means by the introduction of critics, for example (Sussman 1975, Sacerdoti 1977). This is equivalent to having the general problem solver consult an expert when it gets in trouble. The meta-planning allows the general problem solver to call a general problem solver (itself) instead. Thus all the power of such a system can be focussed on planning problems, rather than just relying on a few expert tactics. Of course, all the specific knowledge usually embodied in critics would still be available to the general problem solver. But the meta-planning model allows this knowledge to interact with all other knowledge as it now take take part in general reasoning processes.

1.3.4. Representational advantages

The meta-planning model also provides more flexibility when no solution can be found. Since a meta-goal represents the formulation of a problem, the existence of the problem may be dealt other than its being fully resolved. For example, the problem solver may simply decide to accept a flawed plan if the violation is viewed as not being too important, or decide to abandon one of the goals that it can't satisfy. By separating solving the problem from formulating the problem, the problem may be accessed as opposed to treated, an option that most other problem solving models do not allow.

1.4. Projection is Used to Infer Goals and Debug Plans

The fourth significant feature of our planning model is that it is based on *projection*. That is, as the planner formulates a plan for a goal, the execution of this plan is simulated in a hypothetical world model. Problems with proposed plans may be detected by examining these hypothetical worlds.

Projection not only enables the planner to find problems with its own plans, but it also enables it to determine that a situation merits having a new goal. For example, sensing an impending danger requires the planner to project from the current state of affairs into a hypothetical world which it finds less desirable. Having done this projection, the planner can infer that it should have the goal of preventing the undesirable state of affairs from coming into being.

Projecting hypothetical realities also allows a general "goal detection" mechanism to work for meta-goals as well as for "ordinary" goals. When proposed plans for goals are projected, interactions will appear in the hypothetical world. Since such interactions generally indicate that some important planning principle is not being adhered to, the occurrence of this hypothetical negative interaction is usually a signal to the planner to achieve some particular meta-goal.

Working with projected universes entails some liabilities as well, as does the notion of meta-planning and of using highly declarative representations. However, our claim is that the prices associated with these ideas are prices that must be paid anyway. By putting them together in the manner described here, a deal of power is obtained for no additional cost.

In the next section, I discuss the general structure of a planning mechanism based on these assumptions. This is the structure used in PANDORA (Plan ANalysis with Dynamic Organization, Revision, and Application), a planning system now under construction at Berkeley. The sections following show how these mechanisms function together in reasoning about *goal conflict situations*. As we have noted, we intend these ideas to be applicable to understanding as well as planning, and in fact, they are being used in a new implementation of PAM, a plan-based story understanding system. While we do not discuss the structure of PAM here, the analysis of goal conflicts is presented in a form in which its use in understanding as well as planning may be seen.

2. The Design of a Planner Based on Meta-planning

This section described the overall architecture of a planner based on the four tenets just considered. The planner is composed of the following major components:

1) The Goal Detector

This mechanism is responsible for determining that the planner has a goal. The goal detector has access to the planner's likes and dislikes, to the state of the world and any changes that may befall it, and to the planner's own internal planning structures and hypothetical world models. The goal detector can therefore establish a new goal because of some change in the environment, because such a goal is instrumental to another goal, or in order to resolve a problem in a planning structure that arises in a hypothetical world model.

2) The Plan Generator

The plan generator proposes plans for the goals already detected. It may dredge up stereotyped solutions, it may edit previously known plans to fit the current situation, or it may create fairly novel solutions. The plan generator is also responsible for expanding high-level plans into their primitive components to allow execution.

3) The Executor

The executor simply carries out the plan steps as proposed by the plan generator. It is responsible for the detection of errors, although not with their correction.

The importance of the goal detector should be emphasized. Most planning systems do not worry about where their goals come from; high-level goals are generally handed to the planner in the form of a problem to be solved. However, a planning system needs to infer its own goals for a number of reasons: an autonomous planner needs to know when it should go into action; for example, it should be able to recognize that it is hungry or that its power supply is low and what goal it should therefore have. It should be able to take advantage of opportunities that present themselves, even if it doesn't have a particular goal in mind at the time. It should be able to protect itself from dangers from its environment, from other planners, or from consequences of its own plans.

The goal detector operates through the use of a mechanism called the *Noticer*. The Noticer is a general facility in charge of recognizing that something has occurred that is of interest to some part of the system. The Noticer monitors changes in the external environment and in the internal states of the system. When it detects the presence of something that it was previously instructed to monitor, it reports this occurrence to the source that originally told it to look for that thing. The Noticer can be thought of as a collection of IF-ADDED demons whose only action is to report some occurrence to some other mechanism.

Goals are detected by having themes and meta-themes asserted into the Noticer with orders to report to the goal detector. *Theme* is a term used by Schank and Abelson (1975) to mean something that gives rise to a goal; a meta-theme, similarly, is responsible for generating meta-goals. For example, we can assert to the noticer that when it gets hungry (i. e., when the value of some internal state reaches a certain point), the planner should have the goal of being not hungry (i. e., of changing this value), or that if someone is threatening to kill the planner, that the planner should have the goal of protecting its life. On the meta-level, we might assert that if a goal conflict comes into existence, then the planner should have the meta-goal of resolving this conflict.

Note that the presumption of a goal detector coupled with meta-planning creates a system of considerable power. For example, no separate mechanism is required for detecting goal conflicts or for noticing that a set of proposed plans will squander a resource. The need to resolve conflicts or conserve resources is expressed by formulating descriptions of the various situations in which this may occur, and the appropriate meta-goal to have when it does. By asserting these descriptions into the Noticer to detect meta-goals, goal conflicts and other important goal interactions are handled automatically.

The planner component of our model itself consists of three components:

1) The Proposer,
   which suggests plausible plans to try
2) The Projector,
   which tests plans by building hypothetical world models of what it would be like to execute these plans
3) The Revisor,
   which can edit and remove parts of a proposed planning structure

The Proposer begins by suggesting the most specific plan it knows of that is applicable to the goal. If this plan is rejected or fails, the Proposer will propose successively more general and "creative" solutions. Once the Proposer has suggested a plan, the Projector starts computing what will happen to the world as the plan is executed. The difficult problems in conducting a simulation involve reasoning about "possible world" type situations which are not amenable to standard temporal logic (McCarthy and Hayes, 1969). However, we finesse this issue by defining hypothetical states in terms of what the planner thinks of in the course of plan construction. In other words, our solution is to let the system assert the changes that would be made into a hypothetical data base, in the meantime letting the goal detector have access to these states. Thus if the plan being simulated would result in the planner dying, say, this would constitute a hypothetical undesirable state, which might trigger further goals, etc.

As the Projector hypothetically carries out the plan, and other goals and meta-goals are detected by the goal detector, the original plan may have to be modified. This is done by explicit calls to the Revisor, which knows the plan structure and can make edits or deletions upon request. The modified plan structure is simulated again until it is either found satisfactory or the entire plan is given up and a new one suggested by the Proposer.

Actually, the function of the Projector is somewhat more pervasive than has so far been described. The Projector must be capable of projecting current events into future possibilities based both on the intentions of the planner and on its analysis of those events themselves. For example, if the planner sees a boulder rolling down the mountain, it is the job of the Projector to project the future path that the boulder will traverse. If the projected path crosses that of the planner, for example, a preservation goal should be detected. Thus the Projector is a quite powerful and general device that is capable of predicting plausible futures.

3. Reasoning about Goal Conflicts

In the next two sections we give a more detailed analysis of one particular part of our planning model, namely, the resolution of goal conflicts. The problem is important in its own right; however, the presentation that follows is aimed at demonstrating the kind of "strategy architecture" to which the model is conducive. In particular, the section illustrates a number of important meta-goals and the meta-plans for them, and describes how they would be invoked and utilized by the model. The section also emphasizes the utility of meta-planning for the application of planning knowledge to understanding goal conflicts as well as to planning for them.

Since it is desirable to achieve all of one's goals, a planner faced with a goal conflict will probably attempt to resolve that conflict. We express this by saying that the state of having a goal conflict is a situation that causes the meta-theme "ACHIEVE AS MANY GOALS AS POSSIBLE" to become active. In such a situation, this meta-theme creates the meta-goal RESOLVE-GOAL-CONFLICT. This is a meta-goal because resolving the conflict can be viewed as a planning problem that needs to be solved by the creation of a better plan. In this formulation, the resolution of the goal conflict is performed by the execution of a meta-plan, the result of which will be a set of altered plans whose execution will not interfere with one another.

The knowledge needed to replan around a goal conflict is quite diverse, and may depend upon the particular goals in question and on the nature of the conflict. However, the meta-plans with which this knowledge is applied are rather general. To see why, it is necessary to ask how it is possible for goal conflicts to be resolved at all. There appear to be two ways in which goal conflicts can come about that determine how they may be resolved:

1) The conflict detected is based on the plans for one's goals, rather than on the goals themselves. In this case, it may be possible to achieve the goals by other, non-conflicting plans.
2) The conflict depends upon some additional circumstance or condition beyond the stated goals or plans. The conflict might therefore be resolved if this circumstance is changed.

We therefore define two very general meta-plans, RE-PLAN and CHANGE-CIRCUMSTANCE. Of course, to be effective, we need to supply these meta-plans with more information; if we use RE-PLAN blindly, for example, we might end up enumerating all possible plans for each conflicting goal, although many of these plan combinations will present the same problem that caused the original goal conflict.

3.1. RE-PLAN

There are a number of different re-planning strategies applicable to goal conflict situations. They are given here in order of decreasing specificity. This is in accordance with our belief about the order in which such plans would actually be used, i. .e, the most specific one first, then progressively more general ones, until a satisfactory set of plans is found. In this respect, meta-plans are entirely analogous to ordinary plans insofar as the planning process is concerned.

The order of plan application is just a corollary of the First Law of Knowledge Application "Always use the most specific piece of knowledge applicable"

3.1.1. USE-NORMAL-PLAN applied to resolving goal conflicts

The most specific re-planning strategy is likewise analogous to the planning strategy for ordinary goals, namely, find a normal plan. A normal plan in the case of goal conflict is to find a stored plan specifically designed for use in a goal conflict between the kinds of goals found in the current situation. For example, consider the following situation:

(1) Mary was very hungry, but she was trying to lose some weight. She decided to take a diet pill.

In (1), there is a conflict between the goal of losing weight and satisfying hunger, as the normal plan for the latter goal involves eating. The RE-PLAN meta-plan is used, and the USE-NORMAL-PLAN strategy applied. The normal plan found that is applicable to both goals is to take a diet pill.

Just as many objects are functionally defined by the role they play in ordinary plans, so some objects are functionally defined by the role they play in plans aimed at resolving specific goal conflicts. Thus a diet pill is an object functionally defined by its ability to resolve the conflict between hunger and weight loss; a raincoat is defined by the role it plays in preventing wetness when one must go outside. In fact, a great deal of mundane planning knowledge appears to consist of plans for resolving specific types of goal conflicts.

3.1.2. Intelligent use of TRY-ALTERNATE-PLAN to find non-conflicting set

A general planning strategy that is applicable when a plan cannot be made to work is to try another plan for that goal. In the case of resolving goal conflicts, this means that alternative plans for each conflicting goal can be proposed until a set is found that are not in conflict. As noted above, this may be costly, but it will only be tried when no canned conflict resolution plan has been found. Moreover, the plan can provide some intelligent ways of proposing new alternatives that may help keep costs down.

For example, consider the following situation:

(2) John was going outside to pick up the paper when he noticed it was raining. He looked for his raincoat, but he couldn't find it. He decided to get Fido to fetch the paper for him.

Here, John first thought to walk outside, but then found that this would cause a conflict. As his normal plan for resolving this conflict failed, John tried proposing other plans, looking for ones that wouldn't entail his getting wet. Since getting the dog to fetch the paper is such a plan, and since John presumably doesn't care if Fido gets wet, this plan is adopted.

The meta-planning strategy used here is called TRY-ALTERNATIVE-PLAN. The difference between using this meta-plan and blind generate and test strategies is that some control can be exerted here over exactly what is undone and what is looked for as a replacement. For example, the backtracking here need not be chronological or dependency-directed, but can be *knowledge-directed*. That is, rather than undo the last planning decision, a planning decision related to either goal can be undone, possibly based on an informed guess.

In addition, when fetching a new plan, it may be possible to specify in the fetch some conditions that the fetched plan may have to meet without actually testing that plan for a conflict. For example, in the case of getting the newspaper when it is raining, we can ask for a plan for getting something that doesn't involve going outside. That is, we can look for a plan for one goal that does not contain an action that led to the original conflict. If our memory mechanism can handle such requests, then we can retrieve only those plans that do not conflict in the same way that the original plan does.

In order for this to work, TRY-ALTERNATIVE-PLAN needs to know what part of a plan contributed to the goal conflict so it can look for a plan without this action. This generally depends upon the kind of conflict. We can formulate this within the meta-planning framework by defining a meta-plan called MAKE-ATTRIBUTION. Here, MAKE-ATTRIBUTION is used as a subplan of the TRY-ALTERNATIVE-PLAN meta-plan, although we shall make other uses of it below. TRY-ALTERNATIVE-PLAN first asks MAKE-ATTRIBUTION to specify a cause of the problem, and then fetches a new plan without the objectional element in it.

TRY-ALTERNATIVE-PLAN can also control how far up the proposed goal-subgoal structure it should go to undo a decision. Thus, if no alternative plan for a goal can be found, the goal itself can be questioned if it is a subgoal of some other plan. For example, consider the following scenario:

(3) John was going to get the newspaper when he noticed it was raining. He decided to listen to the radio instead.

Here the entire subgoal of getting the newspaper was eliminated. Since this was apparently a subgoal of finding out the news, the alternative plan of listening to the radio can be substituted. Once again, MAKE-ATTRIBUTION is used to propose a plan that doesn't involve an unwanted step. The difference between this and the last case is that here a plan lying above the conflicting goal is re-planned.

3.2. CHANGE-CIRCUMSTANCE

In addition to the RE-PLAN meta-plan, the other general goal conflict resolution strategy is to change the circumstance that contributes to the conflict. This is actually more general than RE-PLAN, because it may be applicable to conflicts where the goals themselves exclude one another, whereas RE-PLAN requires the conflict to be plan-based.

CHANGE-CIRCUMSTANCE can resolve a goal conflict by altering a state of the world that is responsible for the goals conflicting with one another. Once this has been achieved, the original set of plans may be used without encountering the original problem.

For example, consider the following situations:

(4) John had a meeting with his boss in the morning, but he was feeling ill and wanted to stay in bed. He decided to call his boss and try to postpone the meeting until he felt better.

(5) John wanted to live in San Francisco, but he also wanted to live near Mary, and she lived in New York. John tried to persuade Mary to move to San Francisco with him.

In (4), John's conflict is caused by his plan to attend the meeting and his plan to stay home and rest. These plans conflict because of the time constraints on John's meeting, which force the plans to overlap; the plans require John to be in two places at once, so they cannot be executed simultaneously. If the time constraint on attending the meeting were relaxed, however, then the conflict would cease to exist. Thus rather than alter his plans, John can seek to change the circumstances that cause his plans to conflict by attempting to remove the time constraint that are a cause of the difficulty.

In (5), the conflict is between living in San Francisco and being near Mary, who is in New York. The basis for this exclusion involves the location of San Francisco and of Mary. However, if one of these locations were changed so that the distance between them were reduced, them the state would no longer exclude one another. Thus John can attempt to change Mary's location, while still maintaining his original goals.

To decide what circumstance to change, a planner once again needs to analyze the cause of the conflict. Thus CHANGE-CIRCUMSTANCE first requires the use of MAKE-ATTRIBUTION to propose a candidate for alteration. As was the case for RE-PLAN, MAKE-ATTRIBUTION requires access to detailed knowledge about the nature of negative goal interactions in order to find a particular circumstance with which to meddle. An analysis of such interactions appears in Wilensky (1978).

4. Goal Abandonment

When attempts to resolve a goal conflict are unsuccessful, a planner must make a decision about what should be salvaged. In terms of meta-planning, we can describe these "goal abandonment" situations as follows. The inability to achieve a RESOLVE-GOAL-CONFLICT meta-goal results in the planner having this failed meta-goal. Having a failed RESOLVE-GOAL-CONFLICT meta-goal is a condition that triggers the meta-theme MAXIMIZE THE VALUE OF THE GOALS ACHIEVED. This triggering condition causes this meta-theme to invoke a new meta-goal, called CHOOSE-MOST-VALUABLE-SCENARIO. This goal is satisfied when the relative worth of various achievable subsets of the conflicting goals is assessed, and the subset offering the greatest potential yield determined.

To achieve this meta-goal, we postulate a SIMULATE-AND-SELECT meta-plan. This plan proposes various combinations of goals to try, and computes the worth of each combination. The most valuable set of goals is returned as the scenario most worth pursuing.

4.1. The SIMULATE-AND-SELECT meta-plan

The SIMULATE-AND-SELECT meta-plan has a rich structure. To begin with, it makes a number of presumptions about evaluating the cost and worth of goals and of comparing them to one another. We presume that values can be attributed to individual states in isolation *ceteris paribus*, and that the value of a set can be computed from its parts. This does not presume that the computation is simple; indeed, it may involve the consultation of large amounts of world knowledge. However, we do assume that all values can be made commeasurable. The general issues involving attributing values to goals are discussed (although by no means resolved) in Wilensky (1980).

The SIMULATE-AND-SELECT meta-plan has in effect two distinct options. The first is quite straightforward. It simply involves constructing maximal achievable (i. e., non-conflicting) subsets from among the conflicting goals, and evaluating the net worth of each one. Since we are generally dealing with two goals in a conflict, this means just evaluating the worth of one goal and comparing it to the value of the other. Thus if having the newspaper is deemed more valuable than getting wet, then the planner walks outside to get the newspaper and allows himself to get soaked. Alternatively, a reader trying to understand someone else's behavior would use knowledge about this meta-plan to make inferences about their value system. If we observe John risking getting wet into to get his morning paper, then we conclude that having his paper is worth more to him than getting wet.

However, there is another set of alternatives that need to be considered. Consider once again the example of fetching the newspaper in from the rain, in which the original goals are to get the newspaper and to remain dry. Rather than abandon either goal completely, a reasonable alternative is to try to reduce the degree to which one gets wet as much as possible. A plan for remaining as dry as possible while moving through the rain is to run as fast as one can. This plan satisfies one goal entirely, and another to a degree. The total value of this scenario is likely to be greater than the value of staying dry but not getting the paper, and since the other abandonment possibility (getting the paper but getting soaked) is clearly worse than this (i. e., getting the paper but getting less soaked), the scenario involving partial fulfillment is likely to be adopted.

Partial goal fulfillment is a general principle that is applicable to all goals that involve scalar values. It allows the SIMULATE-AND-SELECT meta-plan to propose options in which the partial fulfillment of one goal enables the (possibly partial) fulfillment of the other. This process is illustrated by the "newspaper in the rain" example: MAKE-ATTRIBUTION determines that the problem with the "stay dry" goal above is that it requires not going outside. Thus a partial version of this goal is sought that doesn't involve this condition. In the case of not getting wet above, the "stay as dry as possible" alternative is selected because this doesn't require not going outside. This scenario is therefore hypothesized and evaluated along with the strict abandonment options, and the one with the highest value chosen.

5. Summary and Projections

We have proposed a theory of planning based on four tenets: (1) Commonsense planning is essentially the consideration of interactions of otherwise simple plans, (2) knowledge about planning should be usable both by a planner and an understander, (3) planning problems should be formulated as meta-goals, and solved by the same planning mechanism responsible for the fulfillment of ordinary goals, and (4) to accomplish much of its mandate, the planner makes projections of the future based on its current knowledge of the world and its own tentative plans.

These tenets form the basis for a model of planning whose most salient features are a goal detector and a projector. The goal detector is used to infer goals, including meta-goals, based on the situations in which the planner finds itself; the projector is used to guess what the future will bring based on the planner's current beliefs and plans. As the goal detector has access to the hypothetical situations simulated by the projector, it can detect problems with currently intended plans by noticing their consequences in hypothetical realities. These problems are dealt with by setting up meta-goals to try to assure a more desirable future state of affairs.

We examined this model of planning in the particular domain of goal conflict resolution. Here we found use for the meta-plans RE-PLAN (consisting of USE-NORMAL-PLAN and USE-ALTERNATE-PLAN) and CHANGE-CIRCUMSTANCE for the meta-goal RESOLVE-GOAL-CONFLICT. Both meta-plans make use of the powerful sub-plan MAKE-ATTRIBUTION. For the related goal of CHOOSE-MOST-VALUABLE-SCENARIO, the SIMULATE-AND-SELECT meta-plan is used to create alternatives involving goal abandonment and partial goal fulfillment. MAKE-ATTRIBUTION was found to be useful here as well.

We are currently attempting to test these ideas in two programs. PAM, a story understanding system, uses knowledge about goal interactions to understand stories involving multiple goals. That is, PAM can detect situations like goal conflict and goal competition, and, realizing that these threaten certain meta-goals, PAM will interpret a character's subsequent behavior as a meta-plan to address the negative consequences of these interactions. Thus PAM makes use of the knowledge structures described above, but of course, it does not test the model of planning per se.

Both the model of planning knowledge and of planning is being used in the development of PANDORA (Plan ANalyzer with Dynamic Organization, Revision and Application). PANDORA is given a description of a situation and determines if it has any goals it should act upon. It then creates plans for these goals, using projection to test them. New goals, including meta-goals, may be inferred in the process, possibly causing PANDORA to revise its previous plans

The following is an example of the kind of planning situation that PANDORA can handle. PANDORA is presented with a task that requires it to get some nails and to get a hammer. PANDORA proposes the normal plans for these goals, which require it to go to the store, buy the desired item, and return. As the plans involve some common preconditions, the meta-theme "DON'T WASTE RESOURCES" causes PANDORA to have the meta-goal COMBINE-PLANS. A meta-plan associated with this goal synthesizes a new plan that involves going to the store, buying both objects, and returning.

PANDORA can also detect and resolve a number of goal conflict-base situations. In addition, PANDORA is being used to model the planning processes of a human who needs to cook dinner during a power failure, in which most of the normal plans for one's goals will not be effective.

References

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

2 Hayes-Roth, B. and Hayes-Roth, F. (1978). Cognitive Processes in Planning. RAND Report R-2366-ONR.

3 McDermott, Drew (1978). Planning and Acting. In *Cognitive Science* vol. 2, no. 2.

4 McCarthy, J. and Hayes, P. J. (1969) Some philosophical problems from the standpoint of artificial intelligence. In Meltzer and D. Michie (eds.) *Machine intelligence*, vol. 4. New York: American Elsevier, pp. 463-502.

5 Newell, A., and Simon, H. A. (1972). *Human Problem Solving*. Englewood Cliffs, N. J.: Prentice Hall.

6 Rieger, C. (1975). The Commonsense Algorithm as a Basis for Computer Models of Human Memory, Inference, Belief, and Contextual Language Comprehension. In *Theoretical Issues in Natural Language Processing*, R. Schank and B. L. Nash-Webber, (eds.). Cambridge, Mass.

7 Sacerdoti, E. D. (1977). *A Structure for Plans and Behavior*. Elsevier North-Holland, Amsterdam.

8 Schank, R. C. and Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding*. Lawrence Erlbaum Press, Hillsdale, N.J.

9 Shortliffe, E. H. (1976). *MYCIN: Computer-based Medical Consultations*. American Elsevier.

10 Stefik, Mark J. (1980). Planning and Meta-Planning MOLGEN: Part 2. Stanford Heuristic Programming Project HPP-80-13 (working paper), Computer Science Department, Stanford University.

11 Sussman, G. J. (1975). *A Computer Model of Skill Acquisition*. American Elsevier, New York.

12 Wilensky, R. (1978). Understanding goal-based stories. Yale University Research Report No. 140.

13 Wilensky, R. (1980). Meta-planning: Representing and Using Knowledge about Planning in Problem Solving and Natural Language Understanding. Berkeley Electronic Research Laboratory Memorandum No. UCB/ERL/M80/33.