

## Meta-planning

Robert Wilensky

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

### 1.0 INTRODUCTION

This paper is concerned with the problems of planning and understanding. These problems are related because a natural language understander must apply knowledge about people's goals and plans in order to make the inferences necessary to explain the behavior of a character in a story (Wilensky 1978a). Thus while a story understander is not a planner, it must embody a theory of planning knowledge.

I have developed such a theory in the construction of PAM (Plan Applier Mechanism), a story understanding program. This paper is concerned not with the understanding mechanism itself, but that part of its planning knowledge which is independent of whether that knowledge is used to explain someone's behavior or to generate a plan for one's own use.

One part of this theory of planning knowledge is essentially world knowledge. This includes a classification of intentional structures into elements like plans, goals, and themes (Schank and Abelson, 1975), a description of the structure of these elements (e.g., plans have preconditions and actions that instantiate them, plans are used to achieve goals, etc.), and an actual body of knowledge about particular elements (e.g., asking for something is a way of getting something from someone).

When one attempts to use this world knowledge to understand the intentions of a story's characters, a number of problems soon become apparent. In particular, what is difficult in understanding a person's behavior is not so much understanding the goal and plan he is operating under, but the fact that there are usually numerous goals and plans present in a situation. It is the interactions between these intentional elements that cause much of complexity in both understanding and planning.

For example, consider the following stories:

- (1) John was in a hurry to get to Las Vegas, but he noticed that there were a lot of cops around so he stuck to the speed limit.
- (2) John was eating dinner when he noticed that a thief was trying to break in to his house. After he finished his dessert, John called the police.

In (1), a plausible plan to achieve John's goal is to speed, but John chose to abandon this goal instead. What's needed to understand this story is not just knowledge about cops and speeding tickets, but knowledge that a person might abandon one goal if it conflicts with another goal he considers to be more significant.

Likewise, (2) strikes most people as strange since John should have reacted to the intruder more strongly. The unusualness of this story is due not to knowledge about the plans and goals involved, but the apparent unproductive scheduling of these plans. A more intelligent planner would have dealt with the threat immediately, and then perhaps returned to his meal when that situation had been disposed of.

Thus to understand the behavior of a character, or to generate an intelligent plan, it is necessary to take into account the interactions between goals. Various planning programs (e.g., Sussman 1975, Sacerdoti 1977) deal with some of these interactions by providing specific programmed mechanisms to deal with particular situations. For example, Sussman's HACKER has a celebrated critic that knows about goals clobbering "brother goals", and detects this bug in plans suggested by the plan synthesizer.

The difficulty with this type of solution is that burying this knowledge about how to plan in a procedure assures that such knowledge could not be shared by a program that wished to use this knowledge to understand someone else's behavior in a complicated situation. In addition, as I hope to show, there is a lot of structure to this knowledge that is missed in this fashion, and which is extremely useful both to the tasks of planning as well as plan understanding.

### 2.0 META-PLANNING

One solution to this problem is to create a second body of planning knowledge that is called meta-planning. By this I mean that knowledge about how to plan should itself be expressed in terms of a set of goals for the planning process (called meta-goals), and a set of plans to achieve them (meta-plans). Meta-goals and meta-plans are rodder to the same planning mechanism (or plan understander) that is used to produce a plan of action (or explanation) from ordinary plans.

For example, consider the following situation, either from the point of view of plan understanding or plan generation:

- (3) John's wife called him and told him they were all out of milk. He decided to pick some up on his way home from work.

Most intelligent planners would come up with John's plan, assuming they knew that they pass by a grocery store on the route home. In order to produce this plan, it is necessary to go through the following processes:

1. Realizing that the goal of getting home and getting some milk are overlapping, i.e., that they should be pursued together rather than independently.

2. Adjusting one's plans accordingly. In this case, the plan is modified so as to
  1. Produce a route that takes the planner near the grocery store.
  2. The "go home" plan is suspended at the point at which the grocery store is reached.
  3. The "get milk" plan is executed.
  4. The "go home" plan is resumed.

In terms of meta-planning, this situation has the following structure: There is an important meta-goal "Don't Waste Resources". This goal organizes a number of situations, including those involving goal overlap. That is, the existence of goals that are more profitably pursued together than independently means that there is a potential for wasting a resource. Thus the "Don't Waste Resources" meta-goal is activated (i.e., put on a meta-goal queue). Now a plan for this goal must be chosen. One such meta-plan that is applicable here is Plan Integration, that is, merging two existing plans to take advantage of their common subcomponents. The application of this meta-plan fulfills the "Don't Waste Resources" meta-goal.

The advantage of the meta-planning approach is that the problem of how to deal with complex goal interactions can be stated as a problem to be solved by the same planning mechanism one applies to "ordinary" goals. For example, one may first try out a number of canned solutions, then some standard planning procedures, and if all else fails, try to construct a novel solution.

Note that there are at least three important differences between meta-planning and planning using constraints or critics: Constraints and plan generators are asymmetric in that constraints reject plans, but they don't themselves propose new ones. In contrast, meta-goals not only pick up violations, but suggest new plans to fix the problem. Meta-goals are declarative structures, and thus may be used in the explanation process as well as in planning. In addition, meta-goals are domain independent, encoding only knowledge about planning in general.

McDermott's notion of a policy, or a secondary task comes closest to the notion of meta-planning I propose here. A policy is essentially an explicitly represented constraint. The primary differences between a policy and a meta-goal are that meta-goals include goals that are not necessarily constraints per se; meta-goals refer only to facts about planning as their domain, whereas policies may include domain specific information; policies often entail the creation of pseudo-tasks, whereas meta-goals have meta-plans that deviate less from the structure of normal plans.

Hayes-Roth and Hayes-Roth (1978) uses the term meta-planning to refer to decisions about the planning process. While my use of the term is similar to theirs, they include all types of planning decisions under this name, and their meta-planning is not formulated in terms of explicit meta-goals and meta-plans. I use the term to refer only to a subset of this knowledge, and only when that knowledge is conveniently expressible in terms of explicit meta-goals and meta-plans.

## 2.1 Kinds Of Meta-goals

The following is a brief description of the more important meta-goals so far encountered, along with the situations in which they arise and some standard plans applicable to them. This list is not meant to be complete. It merely reflects the current state of our analysis.

### META-GOALS, SITUATIONS, AND META-PLANS

#### 1. Don't Waste Resources

##### Situations to Detect

##### 1. Goal Overlap

##### Associated meta-plans:

1. Schedule Common Subgoals First
2. Plan Integration
3. Plan Piggybacking (find a new plan that simultaneously fulfills both goals)

##### 2. Multiple Planning Options (more than one plan is applicable to a known goal)

##### Associated meta-plans:

##### 1. Select Less Costly Plan

##### 3. Plan Non-integrability (situations in which the execution of two plans will adversely affect one another, e.g., one undoes a subgoal established by the other)

##### Associated meta-plans:

##### 1. Schedule Sequentially

##### 4. Recurring Goals (A goal arises repeatedly)

##### Associated meta-plans:

1. Subsume Recurring Goal (Establish a state that fulfills a precondition for a plan for the goal and which endures over a period of time (see Wilensky, 1978b))

##### 5. Recursive subgoals (a subgoal is identical to a higher level goal, causing a potential infinite loop)

##### Associated meta-plans:

##### 1. Select Alternate Plan

#### 2. Achieve As Many Goals As Possible

##### Situations to detect

##### 1. Goal Conflict

##### Associated meta-plans: various conflict resolution plans (see Wilensky 1978a)

##### 2. Asymmetric Goal Conflict (Both goals cannot be accomplished if A is performed before B, but B being performed first poses no difficulty)

Associated meta-plans:

1. Schedule Innocuous Plan First
2. Plan Splicing (If one plan has already been started, suspend it, divert to the other plan, and resume original plan when new plan has been executed)
3. Goal Competition (Goal interference with the goal of another planner)

Associated meta-plans:

1. various anti-plans (plans to deal specifically with opposition)
2. various plans for resolving the competition
3. Maximize the Value of the Goals Achieved

Situations to detect

1. Unresolvable Goal Conflict

Associated meta-plans:

1. Abandon Less Important Goal

4. Don't Violate Desirable States

Situations to detect

1. Danger

Associated meta-plans:

1. Create a preservation goal

2. Maintenance Time

Associated meta-plans:

1. Perform Maintenance

3. Anticipated Preservation Goal (The performance of another plan will cause the planner to have a preservation goal)

Associated meta-plans:

1. Select Alternate Plan
2. Protective Modification (Modify original plan so as not to provoke preservation goal)

The following example will illustrate how meta-goals are used in the planning process: Suppose a planner were given the task of fetching a newspaper from outside. It's raining outside, however. After generate an appropriate plan, a detection rule under "Don't Violate Undesirable States" would notice that the execution of this plan will result in the planner getting wet. The meta-plan associated with this meta-goal fires up a new goal of "preserve dryness". Now a plan for this goal is sought.

If a stored plan for this goal is to wear protective clothing, it would be scheduled before the initial plan. If not, then we could establish a subgoal of getting a raincoat. This might spawn a plan that involves going outside, which would violate the Recursive Subgoals condition. The meta-plan here is to choose another plan. If we fail to find one, the "Achieve As Many Subgoals As Possible" meta-goal is activated, as a goal conflict is now seen to exist. The meta-plans for goal conflict resolution are attempted. If they fail, then an unresolvable goal conflict situation exists, and "Maximize the Goals Achieved" is activated. The meta-plan here is to abandon the less important goal. The planner selects whichever goal he values more and then abandons the other.

### 3.0 APPLICATIONS

We are currently attempting to use meta-planning in two programs. PAM, a story understanding system, uses knowledge about goal interactions to understand stories involving multiple goals. As PAM has been discussed at length elsewhere, we will forego a discussion of its use of meta-planning here.

Meta-planning is also being used in the development of a planning program called PANDORA (Plan Analyzer with Dynamic Organization, Revision and Application). PANDORA is given a description of a situation and creates a plan for the goals it may have in that situation. PANDORA is dynamically told about new developments, and changes its plans accordingly.

The following is typical of the kind of situational planning PANDORA is capable of working on. PANDORA is presented with a situation in which it believes it is cooking dinner for itself. PANDORA then receives a call from an old friend, who's only in town for a short while. PANDORA now determines that it has the goal of meeting with this old friend, and that this goal is in conflict with the original goal of preparing dinner as they both occupy the same time slot. Realizing the "Achieve As Many Goals As Possible" meta-goal will be violated, PANDORA looks for a meta-plan for this goal. One such plan currently available to PANDORA is Plan Integration. Thus PANDORA generates the new plan of inviting its friend to join it for dinner.

### References

- 1] Hayes-Roth, B. and Hayes-Roth, F. (1978). Cognitive Processes in Planning. RAND Report R-2366-ONR.
- 2] McDermott, Drew (1978). Planning and Acting. In Cognitive Science vol. 2, no. 2.
- 3] Sacerdoti, E. D. (1977). A Structure for Plans and Behavior. Elsevier North-Holland, Amsterdam.
- 4] Schank, R. C. and Abelson, R. P. (1977). Scripts, Plans, Goals, and Understanding. Lawrence Erlbaum Press, Hillsdale, N.J.
- 5] Sussman, G. J. (1975). A Computer Model of Skill Acquisition. American Elsevier, New York.
- 6] Wilensky, R. (1978). Understanding goal-based stories. Yale University. Research Report #140.