Appears in the 1998 AAAI Fall Symposium on Cognitive Robotics

Plan Management Issues for Cognitive Robotics: Project Overview

John F. Horty
Philosophy Department and
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
horty@umiacs.umd.edu

Martha E. Pollack
Department of Computer Science
and Intelligent Systems Program
University of Pittsburgh
Pittsburgh, PA 15260
pollack@cs.pitt.edu

Abstract

Progress in cognitive robotics can come either from efforts to provide physical robots with more autonomy, or from efforts to make AI techniques more generally applicable to the needs of robotic agents. We are conducting research on dynamic plan management, which is an instance of the latter type of effort. In this paper, we provide an overview of our research project. Plan management is the process (or set of related processes) by which a goal-directed agent coordinates, updates, and monitors its plans in response to ongoing changes in its environment. We discuss the main challenges in automating plan management, and then illustrate our approach by focusing on one plan management task: evaluating options for action in the context of existing plans. We also briefly describe the Plan Management Agent (PMA), a system we are building as a testbed for our plan management theories. Although PMA is not a robotic system, we are interested in also applying dynamic plan management techniques to mobile robots, which is why we are participating in this symposium.

Introduction

Cognitive robotics is aimed at bridging the gap between traditional AI research and traditional research in robotics. The former has studied a wide range of reasoning processes, including processes for forming plans to achieve goals. However, it has not paid adequate attention to the issues involved in grounding those processes in real, changing, uncertain environments. In contrast, robotics research has concentrated on the interaction between reasoning processes and the environments in which they are performed, but the reasoning processes studied have generally been fairly "low-level", aimed at tasks like path planning or object recognition. Clearly, there is a need to develop and analyze techniques for high-level reasoning by agents situated in real environments. Progress towards this goal can come from either of two directions: from efforts to provide physical robots with more autonomy, or from efforts to make AI techniques more generally applicable.

We are conducting research on dynamic plan management, which is an instance of the latter type of effort. In this paper, we provide an overview of our research project. Plan management is the process (or set of related processes) by which a goal-directed agent coordinates, updates, and monitors its plans in response to ongoing changes in its environment. We discuss the main challenges in automating plan management, and then illustrate our approach by focusing on one plan management task: evaluating options for action in the context of existing plans. We also briefly describe the Plan Management Agent (PMA), a system we are building as a testbed for our plan management theories. Although PMA is not a robotic system, we are interested in also applying plan dynamic management techniques to mobile robots, which is why we are participating in this symposium.

Background

A significant amount of prior research effort in the field of Artificial Intelligence has gone into the design and analysis of planning algorithms. For the most part, this work has been guided by several strong, simplifying assumptions, most notably, that the plans will be performed in static, deterministic environments. Although these assumptions have made rigorous formal analysis possible, and indeed, may have been required for initial progress in the field, they make sense only for a limited number of applications, in which planning is done more or less in isolation of other reasoning tasks, and also in isolation of plan execution. Once we turn our attention to agents that perform autonomously in dynamic, uncertain environments, the assumptions made by traditional planners are violated, and it becomes necessary to rethink the traditional AI approaches to planning.

In fact, we argue that the rethinking required is quite radical. Traditional AI planning systems do one thing: they form plans, that is, sequences of actions that lead to a specified goal state when executed in a specified initial state. To give up the assumptions of traditional planning, one needs to change the representations and algorithms used to form plans. A body of recent work has been aimed at doing just that, e.g., by developing algorithms for conditional

(Etzioni *et al.* 1992; Peot & Smith 1992; Collins & Pryor 1995) and probabilistic planning (Blythe 1998; Draper, Hanks, & Weld 1994; Kushmerick, Hanks, & Weld 1995; Goldman & Boddy 1994; Onder & Pollack 1997), and for generating plans in the context of time-dependent utility functions (Williamson & Hanks 1994; Haddawy, Doan, & Goodwin 1995).

But changing the way in which plans are formed is only part of what is required. The demands of dynamic, uncertain environments mean that in addition to being able to form plans—even probabilistic, uncertain plans—agents must be able to effectively *manage* their plans. That is, they need to be able to decide which planning problems to consider in the first place. They need to be able to form incomplete plans now, adding detail later, and they thus need to be able to decide how much detail to include now and when to add more detail. They need to be able to weigh alternative incomplete plans, and to decide among competing alternatives. They need to be able to integrate plans with one another, and to decide when to treat an existing plan as an inflexible commitment and when, instead, to consider modifications of it.

Some of the issues that arise in plan management have been studied in the design of plan execution systems (sometimes also called "reactive planners"), in which the process of deciding which actions to perform is interleaved with the process of executing those actions (Firby 1996; Gat 1992; Georgeff & Ingrand 1989). However, most of the work on plan execution focuses on controlling a set of processes that are either immediately executable or have been suspended but will become executable in response to a particular, known trigger. Plan execution technology has been less concerned with the issues that arise when an agent must not only control its current activities, but also manage future commitments.

An Illustration of Plan Management

We can illustrate the reasoning tasks involved in plan management by means of a simple example, based on the kind of reasoning that humans perform daily. Suppose you have an lecture to present on Monday morning. The week before, you decide to prepare the lecture over the weekend. You don't decide exactly when you will do the preparation, because your other weekend tasks are not fixed yet, and you are confident that you will have sufficient time during the weekend for the preparation. On Friday, before you leave the office for the weekend, you decide on the general outline of the lecture, so that you can bring home the books you will need for the preparation. On Saturday morning, you decide that you will do errands on Saturday, go to a movie Saturday night, and prepare your lecture on Sunday. However, Saturday afternoon, you receive a phone call from a friend who has an extra ticket to the football game on Sunday afternoon. You therefore revise your plans, and decide not to go to the movie, but instead to begin preparing your lecture on Saturday evening. You're not sure whether you'll be able to complete it Saturday evening, but you know that if you can't, you will still have time to do so Sunday evening, after the football game.

This scenario illustrates the type of reasoning tasks we need to model. Plan management consists in making the following sorts of decisions:

- How much detail should now be included in the plans being formed, and how much can be deferred? In the current example, it is important to decide on the general outline of the lecture on Friday, so that you can bring home the right books, but it is not necessary to determine all the details yet.
- When should the missing detail be filled in? You can wait until the weekend to decide exactly when to prepare the lecture. However, once you receive the phone call about the football ticket, you need to decide when you will prepare the lecture, so that you can determine whether it is possible to attend the game and still successfully carry out your existing plan.
- When should existing plans be modified or replanned? This decision is relatively easy when something occurs that causes a plan to break: in such cases, replanning is obviously required. The question is more difficult in cases in which an existing plan is still sound, but where the possibility of modification is suggested by a new opportunity. This kind of situation occurs in our example, where it is still possible for you to adhere to your original plan of preparing the lecture Sunday afternoon, but where replanning is necessary if you would rather go to the football game.
- How can a partial plan be evaluated in the context of existing commitments? In the current example, one "cost" of attending the football game is that you can no longer attend the movies.

Our current research project focuses on developing automated mechanisms for making these kinds of decisions. In the next section, we describe our approach to one of these questions: evaluating a a plan in context.

Evaluating Options in a Context

The theory of rational choice, as formulated in the economic and philosophical literature, assumes that agents evaluate alternative actions by reference to a probability distribution over their possible outcomes together with a utility function defined on those outcomes: in the simplest case, the agent combines probability and utility into a notion of expected utility defined over actions, and then chooses some action whose expected utility is maximal. Our approach to this problem differs in two important ways from that of classical decision theory. First, while decision theory assumes that the utility of an outcome is given as part of the background setting, we note that the overall desirability of an option presented to an agent is often not immediately apparent; and we are explicitly concerned with the mechanism through which it might be discovered. We focus, in particular, on the case in which the option presented to an agent has a known benefit, but requires some effort—the execution of a plan—for its achievement. In order to evaluate the overall desirability of the option, the agent thus has to arrive at some assessment of the cost involved in achieving it.

Second, we insist that the task of evaluating an option should be computationally realizable; and in particular, our work here is developed within the theoretical framework first articulated in (Bratman 1987), and then further elaborated in (Bratman, Israel, & Pollack 1988; Pollack 1992), according to which it is best to view a resource-bounded agent as operating always against the background of some current set of intentions, or plans. In contrast to standard decision theory, where actions are evaluated in isolation, we develop a model in which the options presented to an agent are evaluated against a background context provided by the agent's current plans—commitments to future activities, which, at any given point, may themselves be only partially specified. The interactions between the new option and the background context can complicate the task of evaluating the option, rendering it either more or less desirable in context than it would have been in isolation.

Here, we sketch our theory for a restricted setting, in which all plans are primitive (not hierarchical) and complete, and all actions have deterministic outcomes. In this simple setting, the only ways in which one plan can influence the cost of another is by allowing or blocking the possibility that separate steps might be merged into one. Although our restriction to this special case prevents us from considering many of the more interesting ways in which plans might interact, even this very simple setting is sufficiently rich to allow us to illustrate the shape of our theory, and we defer a detailed treatment of more complicated plan interactions to subsequent work.

Primitive plans

Basic concepts We represent primitive plans using the standard formalism in which a plan consist of a set of steps, temporal constraints on those steps, and causal links, which record dependency relations among steps. As usual, we assume a set of action types, defined in terms of preconditions and effects (for clarity, we limit our attention only to propositional preconditions and effects). The plan steps are instances of the action types. We allow for both qualitative and quantitative temporal constraints; the latter associate steps with actual time points. To this end, we model time as a totally ordered set of moments $\{m_0, m_1, \ldots\}$, where $m_i < m_j$ if and only if $i < j$, and we assume here that each step occupies a single moment of time.

Definition 1 (Primitive plan) A *primitive plan* $\mathcal{P}$ is a triple of the form $\langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$, with these components defined as follows: $\mathcal{S}$ is a set of steps of the form S_i, each associated with a time indicator t_i; $\mathcal{O}$ is a set of ordering constraints, of the form $t_i = t_j$, $t_i < t_j$, $t_i = m_k$, or $t_i < m_k$, where t_i and t_j are time indicators associated with steps belonging to $\mathcal{S}$ and m_k is a moment; $\mathcal{L}$ is a set of causal links of the form $\langle S_i, Q, S_j \rangle$, where Q is an effect of the step S_i and a precondition of the step S_j.

We assume a function *Type* associating each step S_i with $\mathit{Type}(S_i)$, its action type. We require $\mathcal{O}$ to contain a temporal constraint of the form $t_i < t_j$ whenever there is a link $\langle S_i, Q, S_j \rangle$ in $\mathcal{L}$. And we suppose that an entailment relation $\vdash$ is defined on the temporal constraint language, allowing us to draw out implicit consequences (for example, $\{t_j = m, t_i < t_j\} \vdash t_i < m$), and providing us, also, with a notion of consistency for a set of temporal constraints.

Below, we will use as illustration a plan to buy a shirt at the mall, represented as $\mathcal{P}_1 = \langle \mathcal{S}_1, \mathcal{O}_1, \mathcal{L}_1 \rangle$, where $\mathcal{S}_1 = \{S_1, S_2, S_3, S_4\}$, $\mathcal{O}_1 = \{t_1 < t_3, t_2 < t_3, t_3 < t_4, t_3 = m_6\}$, and $\mathcal{L}_1 = \{\langle S_1, A, S_3 \rangle, \langle S_2, B, S_3 \rangle, \langle S_3, C, S_4 \rangle\}$. Step S_1 represents the action of going to the mall; S_2 is the action of bringing one's wallet (which, we suppose, includes a credit card); S_3 represents the action of actually buying the shirt; and S_4 is a dummy step representing the achievement of the goal. The first three temporal constraints are qualitative ones, inherited from the causal links, while the fourth is a quantitative constraint specifying that S_3 must be performed precisely at m_6 (the moment, perhaps, that the shirt is on sale).

We will say that a plan is scheduled when each of its steps has been assigned a specific moment of execution. In this paper, we prohibit schedules with concurrent actions, although, importantly, two steps of the same action type can be *merged*—assigned to the same moment of execution.

Definition 2 (Schedules) A *schedule* for a plan $\mathcal{P} = \langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ is a set of constraints $\overline{\mathcal{O}}$ such that: (1) there is a constraint of the form $t_i = m$ in $\overline{\mathcal{O}}$ for each S_i in $\mathcal{S}$; (2) $\mathcal{O} \cup \overline{\mathcal{O}}$ is consistent; and (3) $\mathit{Type}(S_i) = \mathit{Type}(S_j)$ whenever $\mathcal{O} \cup \overline{\mathcal{O}} \vdash t_i = t_j$. The plan $\mathcal{P}$ is said to be *scheduled* whenever there exists a set of constraints $\overline{\mathcal{O}} \subseteq \mathcal{O}$ such that $\overline{\mathcal{O}}$ is a schedule for the plan; $\mathcal{P}$ is said to be *schedulable* whenever there exists a schedule for it.

As an example, the constraint set $\overline{\mathcal{O}} = \{t_1 = m_3, t_2 = m_4, t_3 = m_6, t_4 = m_7\}$ is a schedule for the plan $\mathcal{P}_1$ above, showing that this plan is schedulable. Of course, a plan whose ordering constraints are themselves inconsistent cannot be scheduled, but even a plan whose ordering constraints are consistent may nevertheless fail to be schedulable, since its only consistent linearizations may be those in which type distinct steps are assigned to the same moment. Schedulability is thus a stronger requirement than mere consistency of temporal constraints.

We focus here on plans that are complete, in the sense that no further planning is needed in order to guarantee the preconditions of their various steps, although additional scheduling may still be required.

Definition 3 (Complete plans) Let $\mathcal{P} = \langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ be a plan. A precondition A of a step S_i from $\mathcal{S}$ is *established* whenever there is some link $\langle S_j, A, S_i \rangle$ in $\mathcal{L}$. A link $\langle S_j, A, S_i \rangle$ from $\mathcal{L}$ is *threatened* whenever there is both an action S_k in $\mathcal{S}$ with effect $\neg A$ and a schedule $\overline{\mathcal{O}}$ for $\mathcal{P}$ such that $\mathcal{O} \cup \overline{\mathcal{O}} \vdash t_j < t_k < t_i$. The plan $\mathcal{P}$ is *complete* just in case each precondition of each step from $\mathcal{S}$ is established and no link from $\mathcal{L}$ is threatened.

This definition of plan completeness is equivalent to the standard notion from the literature, except that it replaces the idea of temporal consistency with the stronger notion of schedulability.

In order to assess the desirability of a new option against a background context, we need to be able to reason about the plans that are formed when two others are combined, as follows.

Definition 4 (Union of plans) Given plans $\mathcal{P} = \langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ and $\mathcal{P}' = \langle \mathcal{S}', \mathcal{O}', \mathcal{L}' \rangle$, the *union* of the two plans is $\mathcal{P} \cup \mathcal{P}' = \langle \mathcal{S} \cup \mathcal{S}', \mathcal{O} \cup \mathcal{O}', \mathcal{L} \cup \mathcal{L}' \rangle$.

Note that the union of two independently schedulable plans might not be schedulable, since their temporal constraint sets may not even be jointly consistent; also, the union of two complete plans might not be complete, since steps in one may threaten links in the other. If the union of two complete plans can be made complete and schedulable simply through the addition of ordering constraints, we say that the plans are strongly compatible.

Definition 5 (Strong compatibility) Let $\mathcal{P} = \langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ and $\mathcal{P}' = \langle \mathcal{S}', \mathcal{O}', \mathcal{L}' \rangle$ be complete plans. Then $\mathcal{P}$ and $\mathcal{P}'$ are *strongly compatible* just in case there is a temporal constraint set $\mathcal{O}''$ such that $\langle \mathcal{S} \cup \mathcal{S}', \mathcal{O} \cup \mathcal{O}' \cup \mathcal{O}'', \mathcal{L} \cup \mathcal{L}' \rangle$ is complete and schedulable.

As an example, consider the plan $\mathcal{P}_2 = \langle \mathcal{S}_2, \mathcal{O}_2, \mathcal{L}_2 \rangle$, where $\mathcal{S}_2 = \{S_5, S_6\}$, $\mathcal{O}_2 = \{t_5 < t_6\}$, and $\mathcal{L}_2 = \{\langle S_5, D, S_6 \rangle\}$, and where the step S_5 has D and $\neg A$ as effects. (Intuitively, $\mathcal{P}_2$ might represent the plan of going home, with D representing the proposition that the agent is at home, and $\neg A$, of course, the proposition that the agent is no longer at the mall.) Then $\mathcal{P}_2$ and the previous $\mathcal{P}_1$ are strongly compatible, as shown by the constraint set $\mathcal{O}'' = \{t_3 < t_5\}$, since $\langle \mathcal{S}_1 \cup \mathcal{S}_2, \mathcal{O}_1 \cup \mathcal{O}_2 \cup \mathcal{O}'', \mathcal{L}_1 \cup \mathcal{L}_2 \rangle$ is complete and schedulable.

The notion of strong compatibility is just that: very strong. It does not allow either of two compatible plans to be modified in any way, but only supplemented with additional scheduling information, in order for their joint execution to be guaranteed. This notion is not, however, the strongest available. A stronger notion is that of perfect compatibility, where two complete plans $\mathcal{P}$ and $\mathcal{P}'$ are defined as *perfectly compatible* just in case their union $\mathcal{P} \cup \mathcal{P}'$ is itself complete and schedulable. It is easy to see that $\mathcal{P}_1$ and $\mathcal{P}_2$, though strongly compatible, are not perfectly compatible, since the joint plan $\mathcal{P}_1 \cup \mathcal{P}_2$ allows for schedules in which S_5 occurs between S_1 and S_3, threatening the link $\langle S_1, A, S_3 \rangle$.

Semantics Eventually, we will want to interpret a plan as specifying a set of allowed futures—intuitively, those futures consistent with an execution of the plan. For reasons of space, however, we restrict ourselves in this paper to a simpler account, in which complete and scheduled plans, rather than futures, are taken as the points in the semantic space, and more abstract plans are associated with sets of these.

We begin by adapting the notion of refinement (Kambhampati, Knoblock, & Yang 1995) from the plan generation literature.

Definition 6 (Refinement; $\sqsubseteq$) Let $\mathcal{P} = \langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ and $\mathcal{P}' = \langle \mathcal{S}', \mathcal{O}', \mathcal{L}' \rangle$ be plans. Then $\mathcal{P}'$ is a *refinement* of $\mathcal{P}$ ($\mathcal{P} \sqsubseteq \mathcal{P}'$) just in case $\mathcal{S} \subseteq \mathcal{S}'$ and $\mathcal{O} \subseteq \mathcal{O}'$ and $\mathcal{L} \subseteq \mathcal{L}'$.

Letting Π represent the set of complete and scheduled plans, we define the semantic interpretation of a plan as follows.

Definition 7 (Interpretation; $v[\mathcal{P}]$) The *interpretation* of a plan $\mathcal{P}$ is the set of its complete and scheduled refinements: $v[\mathcal{P}] = \{\mathcal{P}' : \mathcal{P} \sqsubseteq \mathcal{P}'\} \cap \Pi$.

The idea, of course, is that a plan is to be interpreted as the set of ways in which it might be carried out, and so it is natural to define a plan as consistent whenever there is some way in which it can be carried out.

Definition 8 (Plan consistency) A plan $\mathcal{P}$ is *consistent* just in case $v[\mathcal{P}] \neq \emptyset$.

Note that a complete plan is consistent just in case it is schedulable, and that an incomplete plan is consistent just in case it has a complete and schedulable refinement.

Evaluation of options

For the purposes of this paper, we define an *option* as a complete plan that is presented to an agent for acceptance or rejection. We suppose that an agent evaluates each new option $\mathcal{P}$ against the background of a context $\mathcal{C}$, some plan to which it is already committed, and that the process of evaluation proceeds as follows. First, the agent determines whether $\mathcal{P}$ is compatible with $\mathcal{C}$—where, for the purposes of this paper, we will assume that the concept of compatibility can be usefully approximated through our notion of strong compatibility—and if not, $\mathcal{P}$ is rejected. Of course, this policy of immediately rejecting incompatible options is a considerable simplification. More realistically, an agent faced with an incompatible option $\mathcal{P}$ could explore either local revisions to the plan that might guarantee compatibility, or else alternative plans for achieving the goal that $\mathcal{P}$ aims at; and if the goal is valuable, the agent might also consider modifications of his background context. However, we cannot examine these more sophisticated alternatives in the present paper.

Assuming compatibility, then, the agent should accept the new option just in case its benefit outweighs its cost in the context. Again, we simplify by supposing that the benefit of the option $\mathcal{P}$—represented here as $\beta(\mathcal{P})$—is both apparent and independent of context (in the most natural case, this benefit will derive from the goal state at which the plan is directed). All that remains to be specified, then, is the cost of the new option $\mathcal{P}$ in the context $\mathcal{C}$.

Cost in isolation

We begin by defining the cost of a plan in isolation. We take as given a function *Cost* mapping action types into real numbers representing their costs, and assume that the function is extended to the steps of a plan in the natural way: $Cost(S_i) = Cost(Type(S_i))$.

Next, we introduce an auxiliary notion of point cost, defined only for complete, scheduled plans—the points in the semantic space. Where $\mathcal{P} = \langle \mathcal{S}, \mathcal{O}, \mathcal{L} \rangle$ is such a plan, we partition the plan steps into sets of actions forced (by the temporal constraints) to occur at the same moment, taking $[S_i] = \{S_j : \mathcal{O} \vdash t_i = t_j\}$ for each $S_i \in \mathcal{S}$. We then let $[\mathcal{P}]$ represent the set of these equivalence classes: $[\mathcal{P}] = \{[S_i] : S_i \in \mathcal{S}\}$. It follows from our definition of a schedule that steps in the same equivalence class will necessarily represent actions of the same type; these type-identical steps performed at the same moment are to be thought of as collapsing into a single merged step. We therefore define the point cost of the plan itself as the sum of the costs assigned to the merged steps it contains:

$$Point\text{-}cost(\mathcal{P}) = \sum_{[S_i] \in [\mathcal{P}]} Cost(S_i).$$

Given this auxiliary notion, it is now natural to define the cost of an arbitrary consistent plan as the point cost of the least expensive way in which it might be carried out, that is, the least expensive point in its semantic interpretation.

Definition 9 (Cost of a plan; $\kappa(\mathcal{P})$) Where $\mathcal{P}$ is a consistent plan, the *cost* of $\mathcal{P}$ is the point cost of its least expensive complete and scheduled refinement: $\kappa(\mathcal{P}) = \min\{Point\text{-}cost(\mathcal{P}') : \mathcal{P}' \in v[\mathcal{P}]\}$.

It is easy to see that $\kappa(\mathcal{P}) = Point\text{-}cost(\mathcal{P})$ whenever $\mathcal{P}$ is itself a complete and scheduled plan, and that $\kappa(\mathcal{P}_\emptyset) = 0$ for the null plan $\mathcal{P}_\emptyset = \langle \emptyset, \emptyset, \emptyset \rangle$.

Cost in context Having defined the cost of a plan in isolation, we now turn to our central task of defining the cost of a new option $\mathcal{P}$ in the context of a background plan $\mathcal{C}$. Our treatment of this concept is simple: we take the cost of the new option in context to be its *marginal* cost—the cost of carrying out $\mathcal{P}$ along with $\mathcal{C}$, less the cost of carrying out $\mathcal{C}$ alone.

Definition 10 (Cost of a plan in a context; $\kappa(\mathcal{P}/\mathcal{C})$) Where the plans $\mathcal{C}$ and $\mathcal{P}$ are strongly compatible, the *cost of $\mathcal{P}$ in the context $\mathcal{C}$* is $\kappa(\mathcal{P}/\mathcal{C}) = \kappa(\mathcal{P} \cup \mathcal{C}) - \kappa(\mathcal{C})$.

It follows immediately from this definition that the cost of a plan in the null context is identical to its cost in isolation: $\kappa(\mathcal{P}/\mathcal{P}_\emptyset) = \kappa(\mathcal{P})$. It is also worth noting that the cost of a plan in any context that already includes that plan as a component is zero: $\kappa(\mathcal{P}/\mathcal{P} \cup \mathcal{C}) = 0$.

This definition can be illustrated with a case in which the cost of a new option is actually affected by the background context. Suppose the agent's background context is simply the plan to buy a shirt at the mall, represented by our earlier $\mathcal{P}_1$, and imagine that the agent is presented with the new option of going to the mall for some swim goggles. More exactly, we can take the new option as the plan $\mathcal{P}_3 = \langle \mathcal{S}_3, \mathcal{O}_3, \mathcal{L}_3 \rangle$, where $\mathcal{S}_3 = \{S_7, S_8, S_9, S_{10}\}$, $\mathcal{O}_3 = \{t_7 < t_9, t_8 < t_9, t_9 < t_{10}\}$, and $\mathcal{L}_3 = \{\langle S_7, A, S_9 \rangle, \langle S_8, B, S_9 \rangle, \langle S_9, E, S_{10} \rangle\}$. Here, the steps S_7 and S_8 again represent actions of going to the mall and bringing one's wallet, steps sharing the respective types of S_1 and S_2 from the background plan $\mathcal{P}_1$; the step S_9 represents the action of purchasing the goggles; S_{10} is again a dummy step representing goal achievement; and the statement E represents the proposition that the agent has swim goggles. Let us suppose that these various steps carry the following costs: each of S_2, S_3, S_8, and S_9 carries a cost of 1, since both carrying a wallet and making a purchase are easy to do; each of S_1 and S_7 carries a cost of 10, since any trip to the mall is abhorrent; and S_4 and S_{10}, as dummy steps, both carry a cost of 0.

Given this information, it is clear that $\kappa(\mathcal{P}_1) = 12$—the cost of the agent's background plan is 12. Presumably, then, the benefit of this background plan must be at least 12—we must have $\beta(\mathcal{P}_1) \geq 12$—or the agent would not have adopted it. Suppose, however, that $\beta(\mathcal{P}_3) = 2$. It is clear also that $\kappa(\mathcal{P}_3) = 12$, so that, considered in isolation, the new option would not be worth pursuing. On the other hand, it is easy to see that $\kappa(\mathcal{P}_3 \cup \mathcal{P}_1) = 13$, since the least expensive execution of the joint plan, in which both the steps S_1 and S_7 as well as the steps S_2 and S_8 are merged, carries a cost of 13. Therefore, we have $\kappa(\mathcal{P}_3/\mathcal{P}_1) = \kappa(\mathcal{P}_3 \cup \mathcal{P}_1) - \kappa(\mathcal{P}_1) = 1$. Even though the new option would not be worth pursuing in isolation, it is worth pursuing in context, since its benefit is greater than its cost in context.

As this example shows, the cost of a plan in context may be less than its cost in isolation, but it is also possible for the cost is context to be greater.

Cost estimates Although the notion of cost as the least expensive method of execution is defined for any consistent plan, we do not necessarily assume that the agent knows the true cost either of his background plan or of any new options under consideration. Instead, the agent may only estimate the cost of its plans.

Definition 11 (Cost estimate for a plan) Where $\mathcal{P}$ is a consistent plan, a *cost estimate* for $\mathcal{P}$ is an interval of the form $\epsilon = [\epsilon^-, \epsilon^+]$, where ϵ^- and ϵ^+ are nonnegative real numbers such that $\epsilon^- \leq \kappa(\mathcal{P}) \leq \epsilon^+$.

Cost estimates, so defined, accurately bound the actual cost of a plan, and are thus related to the interval measures of plan cost used in the decision-theoretic plan generation literature (Williamson & Hanks 1994; Haddawy, Doan, & Goodwin 1995; Goodwin & Simmons 1998).

We now show that, under certain coherence conditions, a cost estimate for a plan in context can be derived from a cost estimate for the context together with a cost estimate for the plan and context combined. Assume that $\mathcal{P}$ and $\mathcal{C}$ are strongly compatible plans, and that $\epsilon_{\mathcal{C}} = [\epsilon_{\mathcal{C}}^-, \epsilon_{\mathcal{C}}^+]$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}} = [\epsilon_{\mathcal{P}\cup\mathcal{C}}^-, \epsilon_{\mathcal{P}\cup\mathcal{C}}^+]$ are cost estimates for the plans $\mathcal{C}$ and $\mathcal{P}\cup\mathcal{C}$ respectively. We know from the definition of a cost estimate that $\epsilon_{\mathcal{C}}^- \leq \epsilon_{\mathcal{C}}^+$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}^- \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^+$, but the definition tells us nothing about the relations among the intervals $\epsilon_{\mathcal{C}}$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}$ themselves. Nevertheless, it is reasonable to conclude that $\epsilon_{\mathcal{C}}^- \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^-$, since the least expensive execution of the compound plan $\mathcal{P}\cup\mathcal{C}$ cannot be less costly than the least expensive execution of $\mathcal{C}$, one of its components; and similarly, $\epsilon_{\mathcal{C}}^+ \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^+$. We characterize the pair of estimates $\epsilon_{\mathcal{C}}$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}$ as *jointly coherent* just in case these two conditions hold: $\epsilon_{\mathcal{C}}^- \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^-$ and $\epsilon_{\mathcal{C}}^+ \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^+$.

As long as $\epsilon_{\mathcal{C}}$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}$ are jointly coherent we can derive a cost estimate $\epsilon_{\mathcal{P}/\mathcal{C}} = [\epsilon_{\mathcal{P}/\mathcal{C}}^-, \epsilon_{\mathcal{P}/\mathcal{C}}^+]$ for the plan $\mathcal{P}$ in the context $\mathcal{C}$ in the following way. Given joint coherence, the end points of the intervals $\epsilon_{\mathcal{C}}$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}$ can stand in only two possible ordering relations:

$$(1) \quad \epsilon_{\mathcal{C}}^- \leq \epsilon_{\mathcal{C}}^+ \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^- \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^+,$$
$$(2) \quad \epsilon_{\mathcal{C}}^- \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^- \leq \epsilon_{\mathcal{C}}^+ \leq \epsilon_{\mathcal{P}\cup\mathcal{C}}^+.$$

In either case, it is clear that $\epsilon_{\mathcal{P}/\mathcal{C}}^+$ should be defined as $\epsilon_{\mathcal{P}\cup\mathcal{C}}^+ - \epsilon_{\mathcal{C}}^-$, the maximum possible distance between points in $\epsilon_{\mathcal{P}\cup\mathcal{C}}$ and $\epsilon_{\mathcal{C}}$. In case (1), we know that $\epsilon_{\mathcal{P}/\mathcal{C}}^-$ should likewise defined as $\epsilon_{\mathcal{P}\cup\mathcal{C}}^- - \epsilon_{\mathcal{C}}^+$, the minimum possible distance. In case (2), it is reasonable to take $\epsilon_{\mathcal{P}/\mathcal{C}}^-$ as 0, since we know, even when the low estimate for executing $\mathcal{P}\cup\mathcal{C}$ is less than the high estimate for executing $\mathcal{C}$, that the true cost of executing $\mathcal{P}\cup\mathcal{C}$ can be no less than the true cost of executing $\mathcal{C}$. Combining cases (1) and (2), we can therefore take $\epsilon_{\mathcal{P}/\mathcal{C}}^-$ as $\max[0, \epsilon_{\mathcal{P}\cup\mathcal{C}}^- - \epsilon_{\mathcal{C}}^+]$, leading to the following general definition.

Definition 12 (Cost estimate for a plan in context) Where the plans $\mathcal{P}$ and $\mathcal{C}$ are strongly compatible, let $\epsilon_{\mathcal{C}} = [\epsilon_{\mathcal{C}}^-, \epsilon_{\mathcal{C}}^+]$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}} = [\epsilon_{\mathcal{P}\cup\mathcal{C}}^-, \epsilon_{\mathcal{P}\cup\mathcal{C}}^+]$ be a pair of jointly coherent cost estimates for the plans $\mathcal{C}$ and $\mathcal{P}\cup\mathcal{C}$. Then the *cost estimate for the plan $\mathcal{P}$ in the context* $\mathcal{C}$ is the interval $\epsilon_{\mathcal{P}/\mathcal{C}} = [\epsilon_{\mathcal{P}/\mathcal{C}}^-, \epsilon_{\mathcal{P}/\mathcal{C}}^+]$, where $\epsilon_{\mathcal{P}/\mathcal{C}}^- = \max[0, \epsilon_{\mathcal{P}\cup\mathcal{C}}^- - \epsilon_{\mathcal{C}}^+]$ and $\epsilon_{\mathcal{P}/\mathcal{C}}^+ = \epsilon_{\mathcal{P}\cup\mathcal{C}}^+ - \epsilon_{\mathcal{C}}^-$.

It follows immediately from this definition that $\kappa(\mathcal{P}/\mathcal{C})$, the true cost of $\mathcal{P}$ in the context $\mathcal{C}$, lies within the derived interval $\epsilon_{\mathcal{P}/\mathcal{C}}$; and it is also easy to see that the derived interval $\epsilon_{\mathcal{P}/\mathcal{C}}$ narrows monotonically as the intervals $\epsilon_{\mathcal{C}}$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}$ are narrowed.

The derived interval estimate of cost in context is useful because, in many cases, it allows an agent to accept or reject an option without calculating its true cost. Suppose, for example, that an agent with background plan $\mathcal{C}$ is considering the new option $\mathcal{P}$ with benefit $\beta(\mathcal{P})$; and imagine that the agent has assigned estimated costs $\epsilon_{\mathcal{C}}$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}$ to the plans $\mathcal{C}$ and $\mathcal{P}\cup\mathcal{C}$, from which it derives the estimate $\epsilon_{\mathcal{P}/\mathcal{C}} = [\epsilon_{\mathcal{P}/\mathcal{C}}^-, \epsilon_{\mathcal{P}/\mathcal{C}}^+]$ for the cost of $\mathcal{P}$ in the context $\mathcal{C}$. Then if $\beta(\mathcal{P}) > \epsilon_{\mathcal{P}/\mathcal{C}}^+$, the agent is justified in adopting the new option, since the cost in context of the option is necessarily less than its benefit; and likewise, the agent is justified in rejecting the option if $\beta(\mathcal{P}) < \epsilon_{\mathcal{P}/\mathcal{C}}^-$, since its cost in context is necessarily greater than its benefit. If $\epsilon_{\mathcal{P}/\mathcal{C}}^- \leq \beta(\mathcal{P}) \leq \epsilon_{\mathcal{P}/\mathcal{C}}^+$, there are two subcases to consider. First, if it happens that $\epsilon_{\mathcal{P}/\mathcal{C}}^- = \epsilon_{\mathcal{P}/\mathcal{C}}^+$, then, since we know that $\kappa(\mathcal{P}/\mathcal{C})$ lies within the interval $\epsilon_{\mathcal{P}/\mathcal{C}}$, it follows that $\beta(\mathcal{P}) = \kappa(\mathcal{P}/\mathcal{C})$, and so the agent is justified either in accepting or rejecting the option. If $\epsilon_{\mathcal{P}/\mathcal{C}}^- < \epsilon_{\mathcal{P}/\mathcal{C}}^+$, on the other hand, the agent's interval estimates do not provide enough information to determine whether the option should be adopted or rejected. In this last case, and only this case, the agent is forced to refine his estimates further before making a rational decision, narrowing his cost estimates for $\mathcal{C}$ and $\mathcal{P}\cup\mathcal{C}$, and thereby also narrowing his derived estimate for $\mathcal{P}$ in the context of $\mathcal{C}$.

Reasoning procedures

We have developed and implemented algorithms that enable an artificial agent to perform the reasoning processes described above. The first stage of the processing involves determining whether a new option $\mathcal{P}$ is strongly compatible with a given context $\mathcal{C}$. Algorithms for checking plan compatibility have been studied in earlier work (Yang 1997); we have extended the algorithms developed there to handle plans with quantitative temporal constraints. The second stage of processing computes the estimates $\epsilon_{\mathcal{P}\cup\mathcal{C}}^-$ and $\epsilon_{\mathcal{P}\cup\mathcal{C}}^+$, comparing these to $\beta(\mathcal{P})$, and then iterating to refine the estimates if needed. The computation of the estimates is done using two lattices, representing possible schedules for $\mathcal{P}$ and $\mathcal{P}\cup\mathcal{C}$, where the ordering in the lattice is induced by the step merging decisions in the schedule. In a fashion somewhat reminiscent of the candidate-elimination algorithm (Mitchell 1997), our algorithm maintains, for each lattice, frontiers that bound the space of possible legal schedules for the plans. We provide details of our algorithms elsewhere (Horty & Pollack 1998).

PMA: The Plan Management Agent

To further investigate our theories of plan management, including the theory of option evaluation, we have been building a testbed system, the Plan Management Agent (PMA). This system is intended to be a "smart assistant", that helps a user manage a potentially large and complex set of plans in a dynamic setting. To date, we have been developing a PMA for an academic user, although the core of the system is domain-independent, and could be re-used for other types of user by modifying the knowledge base. The system currently consists of four main components:

1. a reasoning module, which performs the main plan management tasks;
2. a knowledge base, which stores both static information, about the types of procedures performed in the domain, and dynamic information, about the commitments the user currently has.
3. a GUI, which has knowledge of display information, and serves as the interface between the user and the reasoning module.
4. a message-processing module, which controls information flow between the systems component.

Examples of procedures that might be stored in the knowledge base of an academic user include knowledge of the procedures associated with teaching a class, submitting a paper to a conference, overseeing the review of a paper, and so on. Each of these procedures may decompose into other procedures: for instance, submitting a paper involves submitting an electronic abstract, copying the paper, and sending it. There may also be events that need monitoring: after submission, you should expect a notification of receipt within a week. Some procedures may be performed in alternative ways: you may send a paper by mail, by overnight courier, or, if you live in the same city as the program chair, by hand-delivering it. And of course, activities can conflict with one another. For instance, a new option to oversee the review of a might paper conflict with one's plans to be away on vacation at the time the reviews were due.

The reasoning module so far has these capabilities: it can record commitments to structured procedures, determine whether a new option conflicts with existing commitments, determine some (though not all) of the refinements to a new option that are required to prevent such conflicts, and compute the cost of a new option in the context of existing commitments. We are now working to add more plan management reasoning capabilities. Examples include having the PMA alert the user when an expected activity has not occurred—for instance, if you have not received a notice that your submission was received—so that the user can take an appropriate action. In fact, eventually the PMA might take some such actions itself, on behalf of the user.

PMA is not a robotic system, but to be useful it must be capable of effective plan management in dynamic systems. We therefore expect that much of what we learn in building PMA can have impact for robot agents, who clearly need the same capabilities.

Acknowledgements This research has been sponsored in part by National Science Foundation grant number IRI-9619562 to the first author, and by National Science Foundation grant number IRI-9619579 and Air Force Office of Scientific Research Contract F49620-98-1-0436 to the second author.

References

Blythe, J. 1998. *Planning under uncertainty in dynamic domains.* Ph.D. Dissertation, Carnegie Mellon University.

Bratman, M. E.; Israel, D. J.; and Pollack, M. E. 1988. Plans and resource-bounded practical reasoning. *Computational Intelligence* 4:349–355.

Bratman, M. E. 1987. *Intention, Plans and Practical Reason.* Cambridge, MA: Harvard University Press.

Collins, G., and Pryor, L. 1995. Planning under uncertainty: some key issues. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence.*

Draper, D.; Hanks, S.; and Weld, D. 1994. A probabilistic model of action for least-commitment planning with information gathering. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence.*

Etzioni, O.; Hanks, S.; Weld, D.; Draper, D.; Lesh, N.; and Williamson, M. 1992. An approach to planning with incomplete information. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 115–125.

Firby, R. J. 1996. Modularity issues in reactive planning. In *Proceedings of the Third International Conference on AI Planning Systems*, 78–85.

Gat, E. 1992. Integrating planning and reacting in a heterogenous asynchronous architecture for controlling real-world mobile robots. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, 802–815.

Georgeff, M. P., and Ingrand, F. F. 1989. Decision-making in an embedded reasoning system. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 972–978.

Goldman, R., and Boddy, M. 1994. Epsion-safe planning. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence.*

Goodwin, R., and Simmons, R. 1998. Search control of plan generation in decision-theoretic planners. In *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems (AIPS-98)*, 94–101.

Haddawy, P.; Doan, A.; and Goodwin, R. 1995. Efficient decision-theoretic planning: techniques and empirical analysis. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence.*

Horty, J. F., and Pollack, M. E. 1998. Option evaluation in context. In *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-98).*

Kambhampati, S.; Knoblock, C. A.; and Yang, Q. 1995. Planning as refinement search: A unified framework for evaluating design tradeoffs in partial-order planning. *Artificial Intelligence* 76(1-2):167–238.

Kushmerick, N.; Hanks, S.; and Weld, D. 1995. An algorithm for probabilistic planning. *Artificial Intelligence* 76:239–286.

Mitchell, T. M. 1997. *Machine Learning*. New York: McGraw-Hill Co., Inc.

Onder, N., and Pollack, M. E. 1997. Contingency selection in plan generation. In *Proceedings of the 4th European Conference on Planning*, 364–376.

Peot, M., and Smith, D. E. 1992. Conditional nonlinear planning. In *Proceedings of the First International Conference on AI Planning Systems (AIPS-92)*, 189–197.

Pollack, M. E. 1992. The uses of plans. *Artificial Intelligence* 57:43–68.

Williamson, M., and Hanks, S. 1994. Optimal planning with a goal-directed utility model. In *Proceedings of the Second International Conference on Artificial Intelligence Planning Systems*, 176–181.

Yang, Q. 1997. *Intelligent Planning: A Decomposition and Abstraction Based Approach*. New York: Springer.