Plan Representation for Robotic Agents

Michael Beetz
Munich University of Technology,
Department of Computer Science IX,
Orleanstr. 34,
D-81667 Munich, Germany

Abstract

Most robotic agents cannot fully exploit plans as resources for better problem-solving performance because of imminent limitations of their plan representations. In this paper we propose plan representations that are, for a given job, representationally and inferentially adequate and inferentially and acquisitionally efficient. We state what these properties mean in the context of robotic agents and describe how plan representations can be designed to satisfy them. The proposed plan representations have been successfully employed in several longterm experiments on autonomous robots.

Introduction

In recent years, robotic agents, including XAVIER (Simmons *et al.* 1997), RHINO (Beetz *et al.* 2001; Burgard *et al.* 2000), MINERVA (Thrun *et al.* 2000), and NMRA (Muscettola *et al.* 1998), have shown impressive performance in longterm demonstrations. These robotic agents have in common that they use plans as resources for improving their competence. Substantial portions of their controllers are implemented as plans, symbolic specifications of the robots' intended activity that cannot only be executed but also reasoned about and revised. Plan-based control enables these robots to flexibly interleave complex and interacting tasks, exploit opportunities, quickly plan their courses of action, and, if necessary, revise their intended activities.

For the design of such control systems it is useful to consider plan languages as a form of knowledge representation. Rich and Knight (1991) propose representational and inferential adequacy and inferential and acquisitional efficiency as key criteria for designing domain knowledge representations. Transferring these notions to plan representation, we consider the representational adequacy of plan representations to be their ability to specify the necessary control patterns and the intentions of the robots. Inferential adequacy is the ability to infer information necessary for dynamically managing, adjusting, and adapting the intended plan during its execution. Inferential efficiency is concerned with the time resources that are required for plan management operations and adaptation. Finally, acquisitional efficiency systems is the degree to which they support the acquisition of new plan schemata and planning knowledge.

How well plan languages satisfy these features often determines the degree to which an autonomous robot can

- automatically structure continuous behavior and represent the learned structure as plans that support plan management and opportunistic task execution.
- use and learn routine plans that exhibit coping and situated behavior and have therefore a high expected utility.
- anticipate and forestall execution failures based on realistic predictions of the behavior generated by modern robot control systems.

While these capabilities are necessary for realizing many adaptive robotic agents that are to exhibit competent problem-solving behavior in real environments, surprisingly little effort has been spent on developing plan representations that aim at satisfying combinations of these requirements. This paper reports our experiences in designing special purpose plan representations for robotic agents that aim at being representationally and inferentially adequate as well as inferentially and acquisitionally efficient. Such plan languages provide substantial improvements over existing plan representations in that they support the proper integration of different mechanisms for perception, deliberation, action, learning, and communication and that they provide fast built-in causal and teleological inference and manipulation mechanisms for such expressive plan languages.

We will illustrate the use of such plan languages in the context of designing plans for a robot office courier — in particular navigation plans — and discuss their advantages: First, the high-level plans can achieve better performance than achieved by state-of-the-art navigation systems. Second, the plans can be used in very flexible plans, plans that contain opportunistic plan steps and sensor-driven high-priority tasks. Third, the plans support very fast prediction-based online rescheduling of complex tasks.

In the remainder of this paper we proceed as follows. Section 2 discusses the state-of-the-art in plan-based control of robotic agents. In section 3 we state basic design principles underlying our plan representation. Then we illustrate the principles using example representational constructs and explain our design choices. We conclude with a description of longterm demonstrations and a final discussion.

State of the Art

Current robotic agents typically employ reactive plan languages, problem space languages, and policies for Markov decision problems as their means of behavior specification.

Reactive plan languages are mainly concerned with the effective and competent achievement of goals (Firby 1987; Myers 1996) and not with plan generation and revision. Therefore, the control structures provided by these languages are designed to produce flexible and reliable behavior. They specify coping behavior, situation-specific plan expansion, control of continuous processes, and synchronized concurrent behavior. Coping behavior enables robots to deal with interruptions, small problems, and local failures at execution time. Situated plan execution enables the robots to execute sketchy plans by expanding situation-specific subplans to achieve goals and check whether goals are satisfied upon completion. Situations are used to index the appropriate methods to achieve the goals.

Reactive plans take into account that robot behavior is the result of concurrent, interacting control processes. They change the robot's behavior by changing the set of active control processes. Reactive plans can monitor for important transient situations and handle asynchronous events and signals such as successful goal achievement. Key design issues are synchronization of concurrent processes and the modular design of subplans so that these subplans can be used in multiple constellations of active plans (Firby 1994).

MDP policies model the controlled process as a finite state automaton in which the robot's actions cause stochastic state transitions. The robot is rewarded for achieving its goals quickly and reliably. A solution for such problems is a policy, a mapping from discretized robot states into fine-grained actions. MDPs form an attractive framework for navigation because they use a uniform mechanism for action selection and a parsimonious problem encoding. The policies computed by MDPs aim at robustness and optimizing the average performance. Problems in the application of MDP planning techniques are the size of realistic state spaces and the handling of concurrent actions and dynamically changing and interacting goals.

Problem space plans Most robot action planners make the problem-space hypothesis (Newell 1990), under which problems are stated by a state space and a set of operators that transform states to successor states. A solution is an operator sequence that transforms a given initial state into a state that satisfies the given goal. Problem space plans are designed to simplify plan generation from first principles.

Problem space plans rest on several assumptions. One is the assumption that only temporal orderings cause the interactions between plan steps. Another one is that the robot is always able to complete plan steps. Contrary to these assumptions, many other control patterns for constraining the interactions between plan steps are necessary for flexible and reliable robot control (see above). The problem space plans also do not take into account the interactive character of plan steps, where the plan steps can be initiated by the robot but need outside feedback for their completion.

Problem space plans typically provide only guidelines for execution. This has the disadvantage that planning processes use models that are too abstract for predicting all consequences of their decisions and that planning processes cannot exploit the control structures provided by the lower layer for specifying flexible and reliable behavior.

The lack of sophisticated control structures brought researchers to side step plan representations for the integration of mechanisms others than robot actions. They have specified complex behavior specifications in other formalisms, even when they performed plan management operations. As a consequence, the mechanisms became black boxes for the plan-based control mechanisms. Examples are the situation-specific configuration of image processing routines (Firby *et al.* 1996) and the use of compiled grammars for dialogue control that prevent plan management.

Layered Software Architectures Problem space plans are typically used in layered robot control architectures (Bonasso *et al.* 1997). These architectures run planning and execution at different software layers and different time scales where a sequencing layer synchronizes between both layers. Each layer uses a different form of plan or behavior specification language. The planning layer typically uses a problem space plan, the execution layer employs feedback control routines that can be activated and deactivated. The intermediate layer typically uses a reactive plan language.

While layered architectures mitigate some of the limitations of plan languages they introduce another problem. Instead of a single plan representation the robot has three different ones with incompatible expressiveness. This implies that the representations have to be transformed into each other, which yields considerable information loss.

Requirements

In this paper, we propose the use of a single and application specific plan representations for robotic agents that support (1) a large spectrum of plan management operations, (2) flexible, reliable, and efficient task execution, and (3) the automatic acquisition of robot plans.

Applications of plan-based control. When designing such plan representations it is crucial to look at different applications and their requirements for plan representations.

In our research we design plan-based controllers for robot couriers and tour-guide robots in human environments, autonomous robot soccer, automatic factory control, and distributed supply chain management.

A key characteristic feature of courier and tour-guide applications is that robots have to perform certain tasks at ***specified locations***. The robot courier, for example, has to pick up a letter at one location and deposit it at another one. The robot manages its plan in that it groups its actions according to the locations where they have to be performed. It then sorts the tasks with respect to the locations such that the order satisfies the robots' criterion of utility, for example, to minimize the expected time resources. Thus plan management operations can be supported by plans being modular and transparent with respect to the locations where the actions have to be performed.

In contrast, in autonomous robot soccer the task of plan-based control is the coordination and synchronizations of the ***movements*** of the different robots. Thus, an appropriate plan representation should represent the movements explicitly, transparently, and modularly. The situation is yet different for the factory application. In factory automation the issue is how to assign jobs to machines, how to ***schedule*** the jobs, and where to place the parts in between the production steps. Finally, in plan-based control of distributed supply chain management the key is to dynamically revise ***negotiation*** tactics.

Properties of plan representation languages. From these considerations we draw the conclusions that a single general plan representation language cannot match all those needs at the same time. Therefore we propose the use of special purpose plan representations which are tailored for the respective application. Let us now flesh out what we mean by representational and inferential adequacy and inferential and acquisitional efficiency in the context of plan-based control of robotic agents.

1. Representational Adequacy: The plans that are reasoned about and manipulated must have the expressiveness of reactive plan languages. In addition to being capable of producing flexible and reliable behavior, the syntactic structure of plans should mirror the control patterns that cause the robot's behavior — they should be realistic models of how the robot achieves its intentions. Plans cannot abstract away from the fact that they generate concurrent, event-driven control processes without the robot losing the capability to predict and forestall many kinds of plan execution failures. A representationally adequate plan representation for robotic agents must also support the control and proper use of the robot's different mechanisms for perception, deliberation, action, and communication. The full exploitation of the robot's different mechanisms requires mechanism-specific control patterns. Control patterns that allow for effective image processing differ from those needed for flexible communication, which in turn differ from those that enable reliable and fast navigation. To fully exploit the robot's different mechanisms, their control must be transparently and explicitly represented as part of the robot's plans. The explicit representation of mechanism control enables the robot to apply the same kinds of planning and learning techniques to all mechanisms and their interaction.

2. Inferential Adequacy: The plan management mechanisms must be equipped with inference techniques that infer the information necessary for plan management. The computational processes for competent plan management must infer the purpose of subplans, find subplans with a particular purpose, automatically generate a plan that can achieve some goal, determine flaws in the behavior that is caused by subplans, and estimate how good the behavior caused by a subplan is with respect to the robot's utility model. Pollack and Horty (1999) stress the point that maintaining an appropriate and working plan requires the robot to perform various kinds of plan management operations including plan generation, plan elaboration, commitment management, environment monitoring, model- and diagnosis-based plan repair, and plan failure prediction.

3. Inferential Efficiency: Plans must support economic inference and plan management. The generation of effective goal-directed behavior in settings where the robots lack perfect knowledge about the environment and the outcomes of actions and environments are complex and dynamic, requires robots to maintain appropriate plans during their activity. They cannot afford to entirely replan their intended course of action every time their beliefs change.

3. Acquisitional Efficiency: Plan representations should support the learning of plans for subsymbolic control processes and efficient and reliable plans for routine activities.

Robot Courier and Navigation Plans

Navigation actions are representative for a large subset of the robots' physical actions: they are movements controlled by the motors of the robot. Physical movements have a number of typical characteristics. First, they are often inaccurate and unreliable. Second, they cause continuous (and sometimes discontinuous) change of the respective part of the robot's state. Third, the interference of concurrent movements can be represented as the superposition of the individual effects. In addition, sometimes movements are planned before they are executed and therefore the causal models need simple models of the plans that are computed. So by designing a plan representation for navigation we hope to get a language that is applicable to a broad class of physical behaviors. The second advantage of choosing navigation as our domain is that navigation is the best understood and developed capability of autonomous mobile robots that can be used for plan-based control.

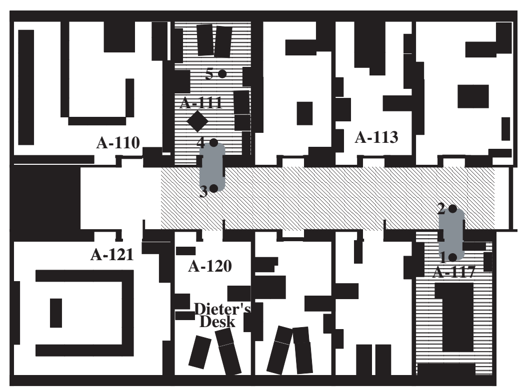

Figure 1: Topological navigation plan for navigating from room A to B with regions indicating different travel modes.

Let us now look at the representational issues for plan-based control of navigation behavior. We will do so by stepwise developing a plan representation for an autonomous robot office courier. We will first represent navigation plans as ordinary reactive plans (3.1). We will then introduce subplan patterns that allow for structuring of continuous control processes and thereby support the acquisition of high-performance symbolic navigation plans (3.2). In the next step, we will take these navigation plans and extend them