Advisable Planning Systems

Karen L. Myers
Artificial Intelligence Center
SRI International
333 Ravenswood Ave.
Menlo Park, CA 94025
myers@ai.sri.com

Abstract

The Advisable Planner project seeks to make AI planning technology more accessible and easier to use through the metaphor of advice-taking. In particular, we are developing an advisable planning system that marries an advice-taking interface to state-of-the-art planning technology. The advice-taking module will accept user specifications of characteristics for both the desired solution and the problem-solving process to be employed for a given task, while the planning component will employ those directives to guide plan construction. This design will enable users to interact with planning systems at high levels of abstraction to influence the plan generation process in terms that are meaningful to them. This paper describes the overall objectives of the project and outlines the technical approach to be employed.

Introduction

Artificial Intelligence (AI) planning technology provides powerful tools for solving problems that involve the coordination and scheduling of actions in the pursuit of specified goals. The AI community has produced several planning systems whose demonstration on realistic problems attests to the value of automated planning techniques. Nevertheless, there has been little success in transitioning this technology to user communities. A major reason for the lack of technology transfer lies with the difficulty of using the planning systems. AI planners have traditionally been designed to operate as 'black-boxes': they take a description of a domain and a set of goals and automatically synthesize a plan for achieving the goals. This design has several drawbacks. First, it explicitly limits the amount of influence that a user can have on the generated plans. Second, it requires complete and accurate formalizations of the domain, since the system is expected to operate without user intervention. Providing such comprehensive domain information is time-consuming and expensive, and represents a significant investment for each new application.

Recent trends toward *mixed-initiative* styles of planning have led to support for certain low-level interactions on the part of humans, such as ordering goals for expansion, selecting operators to apply, and choosing instantiations for planning variables (Tate, Drabble, & Kirby 1994; Drabble & Tate 1995; Wilkins 1993). While a step in the right direction, these interactions are too fine-grained to suit users, who generally want to be involved with the planning process at a higher, more strategic level.

We have recently embarked on a three-year project whose objective is to make AI planning technology more accessible and controllable through the marriage of an *advice-taking* interface with an automated planning system. The resultant *advisable planner* (AP) will accept a variety of instructions and advice from a user and employ those directives to guide plan construction. Advice will encompass a broad range of constructs, including partial sketches of plans or subplans for achieving a set of goals, specific subgoals to be used in pursuit of the overall objectives, restrictions and prescriptions on the use of specific objects and actions. and desired attributes of the final solution.

The advisability paradigm encourages a more active role for users in the plan generation process. In particular, it enables users to interact with a planning system at high levels of abstraction to guide and influence the planning process, with the underlying planner performing the time-consuming work of filling in necessary low-level details and detecting potential problems. As such, the AP model of planning embodies a shift in perspective on the role of planning technology: an Advisable Planner is a tool for enhancing the skills of domain users, not a replacement for them.

AP Architecture

Our model for the Advisable Planner consists of an advice-taking interface layered on top of a core planning system. Advice-taking augments the capabilities of the underlying planning system in the sense that the system does not require advice for its operation. Rather, advice simply influences the set of solutions that the system will provide for a given task. Overall, the AP contains two distinct phases: the *advice translation* phase. and the *problem-solving* phase.

User Advice
||
Parsing
⇓
Normalized Advice
||
Compilation
⇓
Plan Constraints

Figure 1: Advice Representation and Translation

Advice translation involves mapping from user-supplied advice into appropriate internal representations for the planner. The translation process involves several stages, as illustrated in Figure 1. *User advice*, specified in some natural (or pseudo-natural) language is parsed into an intermediate normalized representation. The normal form provides a planner-independent representation of the advice, thus enabling a clean semantic definition and portability amongst different planners. *Advice compilation* is the planner-dependent translation from normalized advice to internal constraints defined in terms of planner-specific operators, goals, and objects. These constraints will be used to direct the plan construction process.

The problem-solving phase takes the compiled advice representations, and generates only solutions that satisfy the advice (referred to as *advice enforcement*). Planning proceeds in a mixed-initiative style whereby the user can make planning-time requests to modify current plans or previously stated advice. The AP may request additional domain information from the user during planning to aid in resolving detected trade-offs in the plan, to recover from planning failures, or to clarify user-supplied advice.

The main thrusts of the AP project are (1) to identify advice idioms for planning that are natural and expressive yet amenable to computationally efficient processing, and (2) to develop advice-sensitive planning algorithms that enforce compiled advice. The many interesting and important issues related to managing interactions with users (such as natural language understanding, dialog management, *etc.*) are beyond the scope of the project.

Basic Advice Idioms

We model advice as task-specific filters on the set of solutions to a given planning problem. Advice differs from the forms of knowledge traditionally used to represent a planning domain (operator schemas, world models, *etc.*). Rather than expressing basic properties of the domain, advice captures session-specific recommendations on the application of traditional forms of planning knowledge in pursuit of a specific objective.

We have identified three basic *advice idioms* that provide broad coverage of the kinds of influence required by users of planning systems.

Task Advice designates specific goals to be achieved and actions to be performed. As such, task advice amounts to partial specification of a solution to a planning task. Examples include Defend the Northeast Sector and Secure Air Superiority in Sector A before Air Superiority in Sector B. Plan sketches constitute another form of task advice.

Strategic Advice consists of recommendations on how goals and actions are to be accomplished; for instance, Use an air-based strategy to defend the Northeast sector and Don't send troops through cities with populations greater than 100,000.

Evaluational Advice encompasses constraints on some metric defined for the overall plan. Common metrics include resource usage, execution time, and solution quality. For example, the directives Deployment time should not exceed 10 days and Don't employ more than 5 sorties in Region H constitute examples of evaluational advice.

The different advice idioms require individualized methods of enforcement. For instance, strategic and task advice are *local* in nature in that they relate to a bounded portion of a plan. In contrast, evaluational advice involves a metric defined over the entire plan (*i.e.*, every action in the plan must be considered to determine whether it contributes to the value being measured). Furthermore, task advice involves the addition and removal of specific goals, while strategic advice influences the choice of methods for achieving both current and future goals.

Our model of advice does not include *control* of problem-solving, such as ordering goals for expansion, or choosing instantiations for planning variables (Currie & Tate 1991; Stefik 1981). While control is an important issue for automated planning systems, it is our view that effective control of the planning process requires deep insights into the mechanics of the planning system itself. While control is an important issue for automated planning systems, it is our view that effective control of the planning process requires deep insights into the mechanics of the planning system itself. As such, it is not a responsibility that should be borne by the user (who generally will not be an expert in AI planning) and thus is not appropriate as a topic for user advice.

Structured Advice

The basic idioms are inadequate to model the full range of influence over the planning process required by users. Greater influence can be attained by enriching advice with additional structuring mechanisms. We describe several such constructs that the AP prototype will support.

One important structuring mechanism is the situationalization of advice to restricted *contexts*. Consider the statement

> Use a feint with strong air support if there are no other feints in progress.

This directive expresses a piece of strategic advice to be applied only in certain situations. Such contextualization of advice is critical to providing adequate expressivity for users. We model contexts in terms of the *problem-solving state* of the planner, using a *Belief-Desire-Intention* approach. *Beliefs* correspond to the expected world state at a given point in a plan, *Desires* correspond to the goals within a plan, and *Intentions* correspond to commitments to actions. This model supports surface-level constructs such as

> "if X is true" *(Belief context)*
> "to achieve X" *(Desire context)*
> "when doing X" *(Intention context)*

Preferences constitute a second important structuring mechanism for advice. Users frequently have multiple, conflicting desires, possibly with situation-specific preferences among them. For instance, a user may want to minimize both the cost and execution time of a plan under construction; these two objectives generally conflict. A user may express conditions under which one objective is more important than the other; for instance, execution time should take precedence when it exceeds some threshold. The AP will enable users to express conditional preferences among advice. The system will identify sources of conflict when all advice cannot be satisfied and inform the user of the space of trade-offs available. This approach will enable the users to make informed choices about alternative solutions to pursue.

A third advice structuring mechanism to be supported in the AP is *advice rationale*, which captures a user's objective in asserting a piece of advice. For example, the advice statements

> Use F117As in order to minimize collateral damage.
> Complete deployment in 10 days in order to protect Route 5.

both include the motivation for the basic advice that they express. Rationales will enable treatment of advice that is more likely to match the user's intent. For example, in the case where collateral damage is already low, the directive to use F117As could be discarded should it conflict with other advice. Similarly, the second piece of advice could be ignored if some other aspect of the plan ensures adequate protection for Route 5.

Technical Approach

In operational terms, advice is compiled into constraints that influence decisions made by the core planning technology. In addition to satisfying standard planning notions of goal/operator applicability, the associated advice constraints must also be satisfied. While traditional planning constraints are grounded in projections about the *world* state, advice constraints further include conditions on the the current plan and open goals (*i.e.*, the *problem-solving* state). This model provides great flexibility while supporting a broad range of advice constructs.

A plan and each of its nodes has an accompanying set of compiled advice constraints. We are developing advice-sensitive planning algorithms in which advice constraints accumulate during processing, being passed downward through the plan refinement structure. We refer to this style of planning as *constraint planning*, given the similarity to Constraint Logic Programming (CLP) (Jaffar & Lassez 1987): both consist of a problem-reduction search augmented with constraints on the overall structure being defined. For CLP, the constraints restrict instantiations of variables; for advisable planning, the constraints further restrict the choice of problem-reduction rules (*i.e.*, the operators to be applied).

AP Prototype

We are developing a prototype Advisable Planner system based on the SIPE-2 generative planner (Wilkins 1988; Wilkins 1993). SIPE-2 is a good choice as the basis of the AP for several reasons. It is a state-of-the-art partial-order AI planning system that supports planning at multiple levels of abstraction. It is capable of rapidly producing complex plans that include parallel actions, conditional actions, and resource assignments. Its replanning capabilities will be useful for handling runtime revisions to goals, domain information and advice. SIPE-2 has been applied successfully to a broad range of applications. In particular, it served as the core reasoning engine in SRI's SOCAP system (System for Operations Crisis Action Planning) (Wilkins & Desimone 1994; Wilkins *et al.* 1995), which was used for the second Integrated Feasibility Demonstration of ARPI (Bienkowski 1995). SOCAP successfully generated employment plans for dealing with specific enemy courses of action, and expanded deployment plans for getting relevant combat forces, supporting forces, equipment, and supplies to their destinations in time for the successful completion of their mission. In sum, the level of sophistication of SIPE-2 will provide a powerful framework in which to develop a rich set of advice-processing methods.

The AP prototype will provide a menu and template interface that enables users to present advice to the AP in a restricted natural language format. In addition, an interactive graphical display will enable users to view, sketch, and modify plans; it will be based on SRI's Act Editor system (Myers 1993; Wilkins & Myers 1995), another ARPI-sponsored technology.

Conclusions

The notion of problem-solving systems that can take advice from humans has been around since the beginning of AI itself (McCarthy 1985). Despite the conceptual appeal, there has been little success to date in building automated advice-taking systems because of the intractability of the task in its most general form. We believe that the paradigm can be made tractable for specific classes of applications and tasks by grounding advice in a focused set of problem-solving activities. The Advisable Planner project presents a step toward this goal for the paradigm of generative planning.

The advice-taking paradigm enables users to provide significant, high-level input to the problem-solving process. This approach has several advantages over fully automated methods. Of greatest significance is the potential for increased user control and involvement in plan generation. Users in many application areas are reluctant or unwilling to cede responsibility of problem-solving to an automated system. A solution generated by a black-box module is unacceptable for several reasons. First, users often resist a solution when they do not understand the process by which it was reached. Second, there is often no single 'correct' solution to a problem. Rather, users want to customize the solution that is produced to match their individual needs and desires. For example, military campaign plans generally reflect the biases and preferences of the commanders in charge, despite the fixed doctrine that underlies much of the planning process (Thaler & Shlapak 1995).

The AP will also lead to a simplification of the domain-modeling problem: because the system is designed to interact with the user during the formulation of plans, the assumption of complete and error-free knowledge required by current-generation planners is eliminated. This interactive approach is imperative for most applications, as it is impossible to replicate fully the domain and commonsense knowledge of human experts.

Overall, we expect that the Advisable Planner paradigm will make automated planning systems more accessible and appealing to users by making it easier to generate high-quality plans that are well-suited to their needs.

Acknowledgements

This research has been supported by ARPA Contract F30602-95-C-0259.

References

Bienkowski, M. 1995. Demonstrating the operational feasibility of new technologies: the ARPI IFDs. *IEEE Expert* 10(1).

Currie, K., and Tate, A. 1991. O-Plan: the open planning architecture. *Artificial Intelligence* 32(1).

Drabble, B., and Tate, A. 1995. O-Plan mixed initiative planning capabilities and protocols. Technical report, University of Edinburgh.

Jaffar, J., and Lassez, J.-L. 1987. Constraint logic programming. In *ACM Symposium on Principles of Programming Languages.*

McCarthy, J. 1985. Programs with common sense. In Brachman, R. J., and Levesque, H. J., eds., *Readings in Knowledge Representation.* Morgan Kaufmann.

Myers, K. L. 1993. *The ACT Editor User's Guide.* Artificial Intelligence Center, SRI International, Menlo Park, CA.

Stefik, M. 1981. Planning and meta-planning. *Artificial Intelligence* 16(2).

Tate, A.; Drabble, B.; and Kirby, R. 1994. O-Plan2: An open architecture for command, planning and control. In Fox, M., and Zweben, M., eds., *Knowledge Based Scheduling.* Morgan Kaufmann.

Thaler, D. E., and Shlapak, D. A. 1995. Perspectives on theater air campaign planning. Technical report, Rand Corporation.

Wilkins, D. E., and Desimone, R. V. 1994. Applying an AI planner to military operations planning. In Fox, M., and Zweben, M., eds., *Intelligent Scheduling.* Morgan Kaufmann. 685–709.

Wilkins, D. E., and Myers, K. L. 1995. A common knowledge representation for plan generation and reactive execution. *Journal of Logic and Computation* 5(6).

Wilkins, D. E.; Myers, K. L.; Lowrance, J. D.; and Wesley, L. P. 1995. Planning and reacting in uncertain and dynamic environments. *Journal of Experimental and Theoretical AI* 7(1):197–227.

Wilkins, D. E. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm.* Morgan Kaufmann.

Wilkins, D. E. 1993. *Using the SIPE-2 Planning System: A Manual for Version 4.3.* Artificial Intelligence Center, Menlo Park, CA.