From: AAAI Technical Report SS-93-03. Compilation copyright © 1993, AAAI (www.aaai.org). All rights reserved.

Exploiting Predictions to Organize Goals in Real-world Domains

Elise H. Turner

Department of Computer Science
University of New Hampshire
Kingsbury Hall
Durham, NH 03824
eht@cs.unh.edu

Abstract

In many situations in the real-world, actions can be grouped together in ways that are beneficial. If an agent could predict all of its actions, it would be easy for the agent to organize them. However, not all actions can be predicted in advance in the real-world.

In this paper, we suggest a method for organizing actions in such a way that the agent can be responsive to new goals. Instead of predicting all actions, the agent forms only an abstract plan. The steps of the abstract plan must be chosen so that they form reliable predictions. They also must group goals together in meaningful ways. As goals arise, the agent can add them to the appropriate step of the abstract plan. The abstract plan gives the agent access to predictions that are important to organize its actions. However, because details of the plan are only added when needed, the agent remains reactive to the unpredictable aspects of the world. We present two systems which use this method: JUDIS which organizes dialogues for distributed systems, and NBA-PLANNER which sequences locations to be visited by an underwater robot.

1 Introduction

Classical planning allows agents to set out an entire course of action. By anticipating all of its actions, the agent is able to organize those actions into an efficient plan. For example, while planning to plant a garden, an agent may first recognize that it needs to buy a rake, and, later, recognize that it must buy a hoe. If the agent forms a traditional plan to plant the garden, it will be able to group the actions of buying a rake and a hoe together in one trip to the hardware store. The predictions made through planning allow agents to determine which actions should be grouped together.

However, predictions in real-world domains may not be reliable. As proponents of reactive planning suggest, unexpected events may make it impossible for the agent to carry out its plan. In this case, the decisions based on the plan's predictions are worthless and the effort spent to create the plan has been wasted.

In this paper, we suggest a method for allowing an agent to take advantage of global predictions to organize goals while remaining responsive to new goals which may enter the system. Predictions are stored in an abstract plan called a *template*. Details are added only as needed — when a new goal is added to an abstract step or when an abstract step must be executed. This approach differs from reactive planners which create hierarchical plans [2; 3] because information about future steps in the plan helps to determine when a goal will be achieved.

We discuss two systems, JUDIS and NBA-PLANNER, which embody this approach to planning. JUDIS [5; 7; 8] works in the domain of natural language processing to organize dialogues for distributed problem solving systems. NBA-PLANNER [1; 6] sequences locations for an autonomous underwater vehicle. After brief overviews of these systems, we will discuss our insights from implementing the approach for two very different domains.

2 JUDIS

JUDIS organizes communication goals from a distributed problem solver into a single, coherent conversation. This task is difficult because different parts of the problem solver may address different parts of the problem solving task. The conversation would flip-flop between these subtasks if goals were simply satisfied in the dialogue in the order in which they were sent from the problem solvers. JUDIS uses predictions about the topics that are likely to be discussed in the dialogue to group incoming goals by topic. Figure A shows an example of a brief dialogue that has been organized by JUDIS. The arrows indicate new goals that arrive after the conversation has begun. The tail of the arrow shows

when the goal arrived at the interface; the head shows where it is satisfied in the dialogue.

JUDIS forms a template from schemas which capture the conventions of conversation. Steps in these schemas are either utterances or other schemas. The steps which can fill a schema are determined by topic. For example, if JUDIS is to help a user plan a meal, the schema to predict the conversation would have the following steps: appetizer, main course, dessert. A schema for relating a story would have individual events in the story as its steps.

JUDIS's schemas form an abstraction hierarchy. The most abstract schemas form the initial set of predictions. They are replaced with more specific schemas as the incoming goals and previous conversation hone the converser's predictions. For example, JUDIS's initial template may only make the predictions that there will be an opening, middle and closing to the conversation. The highly conventional opening and closing may contain very specific predictions, but the middle portion may contain no predictions at all. If JUDIS has a goal to discuss a meal, the schema for discussing meal will replace the empty middle portion of the conversation.

These schemas help JUDIS to make useful predictions for two reasons. First, they make predictions that are reliable. Since the conventions are shared by the community of language users, they constrain the behavior of the other converser. Consequently, these conventions form a set of predictions that are likely to be carried out. The value of such predictions for organizing conversations is further supported by work which suggests people delay utterances because they will fit into the conversation more naturally at a later time [4].

Second, the abstraction into topics that is provided by these schemas divide the world in an important way. Changing topics can be seen as a difficult plan step since it requires the agents to re-focus their attention on a different aspect of the problem. Consequently, to make the conversation coherent we would like to minimize topic switches. Since conversation goals will be grouped together by topic, these switches will be minimized.

To form the template, JUDIS finds the schema which is most closely associated with the known features of the conversation. When a new goal enters the system, JUDIS must find a place for it within the current predictions for the dialogue. To add a new goal to the predictions, JUDIS replaces a current plan step with a new step that will achieve the new goal along with the goals previously associated with that step which is a legal step in the predicted plan. Because the predicted conversation is always made up of conventional schemas, it remains acceptable to the user. However, the predicted conversation retains the ability to handle new goals because some step in the schema can always be expanded to include the new goal.[1]

3 NBA-Planner

The task of NBA-PLANNER is to sequence the locations that will be visited by an autonomous underwater vehicle (AUV). Since AUVs have limited resources to sustain them during a mission, NBA-PLANNER must find the lowest cost path possible. However, the AUV must carry out NBA-PLANNER's plan in an unpredictable environment. Not only must the AUV deal with unanticipated obstacles, but it also must handle new mission goals which may cause it to visit additional goal locations.

To lower costs, NBA-PLANNER forms an abstract plan that can be globally optimized. The world is abstracted into the *naturally-bounded areas (NBAs)* from which NBA-PLANNER takes its name. These areas are surrounded by *natural boundaries*, permanent features of the environment that are difficult or dangerous for the AUV to cross. Natural boundaries include geological and biological features such as mountains, boulder fields, high rocks, coral reefs, and kelp beds. Figure B shows an AUVs environment divided into NBA's. A "G" signifies a goal location.

This abstraction creates predictions which are likely to be realized and divides the world in an important way. The natural boundaries help make accurate predictions because they are permanent features. They are likely to appear in an agent's model of the world and are not likely to change between the time that the model is created and the time that the agent executes its plan. Also, travel between NBAs is assumed to be much more costly than travel within them. This means that minimizing travel between NBAs will be the most important step in minimizing the total cost of the agent's travel. In general, all travel within an NBA should be completed before moving to the next NBA, so local planning can be ignored until the AUV reaches the NBA.

NBA-PLANNER forms a global plan which visits all NBAs which contain goal locations. A globally optimal plan is formed using a variation of an algorithm for the traveling salesman problem. This plan includes all NBAs which contain goal locations (e.g., A,C,E in Figure B), as well as some others which link those NBAs (e.g., D in Figure B). This plan can only be created and optimized if the agent sees visiting the goal locations as a series of actions that will be achieved.

Adding a new goal to the plan is easy for NBA-PLANNER. First, it simply associates the new location

[1] If the goal is not related to any currently predicted topic, it can be added as a new topic at the most abstract level of the plan.

J: Hello.
U: Hello.
J: May I help you?
U: I'd like to plan a party for a friend. He likes Itlaian food.
J: Would you like antepasto for an appetuzer?
U: That sounds good.
J: Would you like spaghetti for the main dish? We could also have salad and garlic bread as side-dishes.
U: Could we have Italian dressing on the salad?
J: Sure
Would you like cannoli for dessrt?
U; Yes.
J: The last time I served cannoli someone could not eat it because he was lactose intolerant. Are any of your guests lactose intolerant?
U: No.
J: I think we have planned a good meal. Please tell me how it turns out.
U: Good-bye.

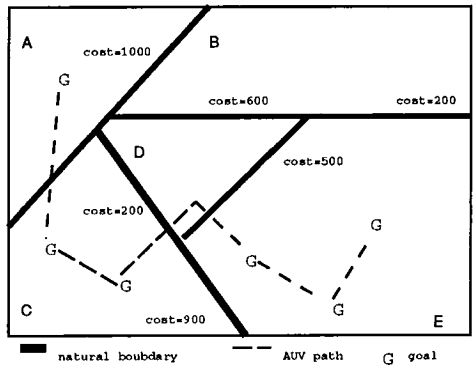

Figure 1: A: A dialogue with JUDIS **B: An AUV Mission**

with the NBA which contains it. If the NBA is currently in the global plan, no more needs to be done. If the new location is in the NBA that the AUV is currently traveling in, it must add the location to its plan for that NBA. Only when the new location is in an NBA not in the global plan does NBA-PLANNER need to replan at the global level. However, the global plan is formed only at the level of NBAs, the time taken to re-plan is proportional to the number of NBA's, not the number of goal locations.

NBA-PLANNER exploits its predictions about the plan when goals can be satisfied in several locations. In these cases, the goal is associated with an NBA which contains active goals, if possible. These NBAs will remain in the template even if it is re-planned. If such an NBA cannot be found, the location is associated with an NBA that is in the plan but does not contain an active goal. This choice highlights the conflict between predictive and reactive planning. Although this is the best choice if the plan is carried out, the agent is now committed to travel to an NBA that otherwise might not be included in the template if it is re-planned. If no location is in an NBA already in the plan, the NBA that is closest to an active NBA already in the plan is chosen and the template is replanned.

4 Using the Approach in Other Domains

From our work with JUDIS and NBA-PLANNER, we are able to make several generalizations for applying our approach in other domains:

- **Which domains are appropriate?** Our approach to planning should be adopted for any domain where the benefits of the template outweigh the risk of replanning. Several characteristics of the domain favor the template:
  1. Local decisions lead to less than optimal plans and resources to execute plans are scarce.
  2. Some outside force, such as societal conventions or institutional standards, dictate how the task must be carried out, so the necessary organization may not be achieved unless the agent follows a template.
  3. The agent has a great deal of control over the world or knowledge about the world at the level of detail reflected by the template.
  4. The agent is committed to achieving the goals in the template and, therefore, will not abandon the plan unless absolutely necessary.

  Other characteristics increase the risk of relying on a template:
  1. Planning and replanning are very costly compared to the cost of not following the optimal plan.
  2. Other agents are working against the agent, and these agents' behavior is not constrained by any rules.
  3. The domain or task is so dissimilar to others that the agent has encountered that it is impossible for it to make reliable predictions at any level of detail.
  4. Goals arise in such a way that they cannot be grouped (e.g., goals must be achieved in a particular sequence or so much time elapses between similar goals that the planner has gone to the next step in the template).

- **What level of abstraction provides the best predictions?** Several factors indicate how predictive a template might be. If the agent is committed to the plan and has control over it at this level of detail, it is at a reasonable level of abstraction. The template is more likely to be predictive if there are several options available for achieving its steps. This favors very abstract steps in the template. However, more detail is warranted when the agent has more information. If the domain is governed by rules which the agent knows, or if the agent has had a great deal of experience with the domain and that experience supports further detail, more detailed predictions can be made. Although in many ways more detailed predictions can be considered better, additional detail may make it more complicated to add goals to the template.
- **How is the template formed?** In a conventional domain or one in which the agent has a great deal of experience, schemas can be used to provide the template. Schemas are advantageous because they can be easily instantiated to form abstract plans. They also are likely to be predictive – they have worked in the past, and they may include important information that is not explicitly in the agent's world knowledge base. When schemas are not available, or when it is more important to optimize some aspect of the plan, an abstract template can be formed using any standard planning technique. In choosing the technique, it is important to evaluate the trade-off between planning time and execution time as well as the cost of stopping execution to replan.
- **How many details should be added when expanding the template to add a new goal?** Only the details required to organize the goals should be added to the template when a new goal arrives. In some cases, as with NBA-PLANNER, the only detail that will be added is the goal. In other cases, as with JUDIS, goals can only be added to steps in certain ways. Here, when a goal is added, the agent also commits to the details needed to add the goal. Details should be added only when necessary so that the agent can remain as reactive as possible.

References

[1] Terri Louise Briggs. Exploiting natural boundaries in the autonomous underwater vehicle domain to limit resource utilization. Technical Report 91-15, University of New Hampshire, December 1991. M.S. thesis.

[2] R. J. Firby. An investigation into reactive planning in complex domains. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 202–206, Los Altos, California, 1987. Morgan Kaufmann.

[3] M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 677–682, Los Altos, California, 1987. Morgan Kaufmann.

[4] E. A. Schegloff. Sequencing in conversational openings. *American Anthropologist*, 70:1075–1095, 1968.

[5] Elise H. Turner. Integrating intention and convention to organize problem solving dialogues. Technical Report GIT–ICS–90/02, School of Information and Computer Science, Georgia Institute of Technology, 1990. Ph.D. thesis.

[6] Elise H. Turner and Terri L. Briggs. Exploiting natural boundaries in planning robot travel. in preparation.

[7] Elise H. Turner and Richard E. Cullingford. Exploiting knowledge of conversation to organize communication goals. Submitted to *Computational Linguistics*.

[8] Elise H. Turner and Richard E. Cullingford. Using conversation MOPs in natural language interfaces. *Discourse Processes*, 12(1):63–90, 1989.