

Provided by the author(s) and University College Dublin Library in accordance with publisher policies. Please cite the published version when available.

Title	Planning to replan in a multi-agent environment
Author(s)	McEleney, Bryan; O'Hare, G. M. P. (Greg M. P.)
Publication Date	2004-12-20
This item's record/more information	<a href="http://hdl.handle.net/10197/4480">http://hdl.handle.net/10197/4480</a>

Downloaded 2013-09-16T11:03:32Z

Some rights reserved. For more information, please see the item record link above.



# Planning to Replan in a Multi-agent Environment

Bryan McEleney and Gregory O'Hare

Department of Computer Science  
University College Dublin, Dublin 4, Ireland

bryan.mceleney@ucd.ie, gregory.o'hare@ucd.ie

## Abstract

A multi-agent planner is described that accounts for the replanning occurring when one agent's action is observed by another. A nested belief model is used to generate an expectation of the other agent's response. Using the planner's output, a dialogue system is being developed which decides whether uncertainties in the belief model should be resolved through dialogue before execution of the domain level plan.

## 1. Introduction

In a changeable and uncertain environment, reactive planning is used to repair and refine plans over time as observations are acquired. Similarly, when one agent must share the environment with others, reactive planning comes into play as each agent observes unpredictable actions by others. Through plan recognition over these observations, the agent may take the other agent's plan and cooperatively add to it and execute another step. The process can be summed up as a repeating cycle of execution, observation, plan recognition, re-planning, and then back to execution. In each iteration, a different agent takes a turn. Examples can be seen everywhere from football to natural language dialogue.

We have constructed a planner that models the action cycle, allowing the decision of one agent to depend on the expected recognition and re-planning process of the next agent, and for that to depend on the third move and so on. A nested belief model contains the agent's own beliefs about the domain state and his own plan rules, his beliefs about the other agent, and so on, continuing to a depth dependent on the number of plan steps. This is needed because agents typically have differing beliefs about states, plan rules and about others' intentions. The agent uses the nested belief model to build a *decision tree* of alternative outcomes, with choice nodes wherever an agent replans. In addition, chance nodes are used since each re-planning step is only probabilistically known. Thereafter, branches are selected by assuming each agent maximises expected utility at each choice point, values are passed back to the root node, and a root decision is made. The result is like a conditional plan since only the chance nodes remain. Modelling of the replanning process sets

this planner apart from cooperative distributed planning approaches (eg [1]) which decompose plans and then perform negotiations between agents to coordinate the sub-plans.

## 2. Constructing Plans

The plans are constructed solely by hierarchical decomposition, and so are represented as one tree. Each agent is assumed to be focussed, in that recognised plans are only extended by adding children to existing nodes in the tree if possible, and in order. This is achieved by a left to right postorder tree traversal. If adding a child is not possible, a parent is added to the tree's root, and a child is subsequently added to the parent. As a result, the probabilistic element of the plan recognition algorithm only concerns the uncertainty relating to the intended parent. This is easily determined via a stored list of parent probability values for each node. Plan recognition is challenging since the action history includes a mix of actions by different agents, with changing intentions. We have found an easy solution in a generate-and-test approach, to run the planner forwards, from the next level of nesting in the belief model, and filter out the plan hypotheses whose sequence of leaf nodes matches the action history. This is recursive on the plan recognition algorithm, since a sub step of planning is to perform plan recognition. To continue the recognised plan, a choice node is generated by applying a plan rule in a focussed fashion to each of the hypotheses. If a parent node needs to be added to accomplish this, a chance node is introduced to represent the possible intentions of the previous actor.

We have implemented and applied this planner to problems in natural language dialogue. For example, figure 1 illustrates one path in a decision tree for a car repair task. An agent is deciding whether to use a short but ambiguous phrase to obtain the large spanner. The nested belief model for this problem includes at the first level, the first agent's intention use\_large\_spanner, for which there is no uncertainty since it is the agent's own belief. The first agent derives a plan from this intention, whose leaf node is to ask for a spanner. Plan rules within the first level are used for this. To continue, the second agent performs plan recognition. The minimal covering plan for

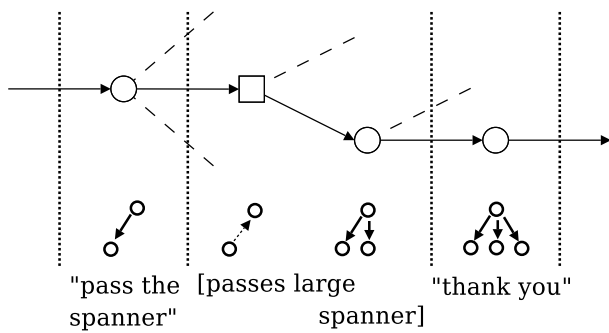


Figure 1: A plan on a decision tree path.

the singleton action history is the action itself, so this is the hypothesis. Since a child cannot be attached to this plan, a parent is added. There are two hypotheses for the parent: `use_small_spanner` and `use_large_spanner`. These come about by using beliefs at the third level of nesting (since the second agent is reasoning about the first agent), which contains the uncertain parent probability list for `ask_spanner`. A chance node is needed to represent the uncertainty, and two corresponding continuations are produced at the choice node, by using beliefs about plan rules at the second level of nesting. In the third step, a parent is available from the previous agent's minimal plan, there is no uncertainty since a chance node has already been introduced for that parent, and so a child can be added immediately, using beliefs about plan rules at the third level of nesting.

### 3. Value of Information and Plan Construction Dialogue

Decision trees can be much larger than that of figure 1. For example, in cooking a meal or building a house there might be long sequences of re-planning steps. In particular, there can be numerous chance nodes caused by uncertain nested beliefs. These nodes can be removed easily by communication between the agents. For example, if one agent is unsure about whether the other will use eggs in his recipe, he can ask. The expected utility gain in doing so can be computed using a value-of-information calculation [2]. To do so, the decision trees for positive and negative answers for the question are examined and their weighted sum gives the expected change in utility. Using this principle, we are developing a natural language dialogue planner that decides whether to communicate about chance nodes. Since it is based on utility, it can decide which plan options to focus on under time-pressure, or decide whether it is better to hold on to, or to give away the floor.

The planner is based on a set of four dialogue *moves* which are inspired by the units of dialogue seen in examples of human cooperative planning, eg. [3]. For each,

we have worked out a utility gain function:

- **query-reply** this is the basic question and answer move
- **give-information** this complements query-reply since it passes information without being asked. It has the same utility function but it is calculated with respect to the beliefs of the next agent. Notice that because of belief differences, one agent may value an information-lend whereas the other does not value the complementary query-reply
- **propose** this is used to promote an attacking debate by declaring the current most valued plan
- **attack** this is an agent's response to a propose. Suppose that the other agent proposed  $p$ , that the attacker's value for  $p$  is  $x$  but the attacker's favoured plan  $q$  is valued  $y$ . Then the attacker expects to gain  $y - x$  in utility by passing information that changes the other agent's decision. An attack is a move sequence composed of any other move type, and so the utility gain is apportioned among the elements of the sequence. An agent considering a propose move uses the expected utility of the resulting attack to evaluate his propose. This is done by running the attack utility function at the next level of nesting, to estimate the other agent's point of view.

## 4. Conclusions

We have developed a multi-agent planner that accounts for the replanning process of the other agents by building decision trees. We are now developing a dialogue planner that analyses the decision trees to decide upon exchanging information that improves their expected utility.

## 5. References

- [1] E. Durfee and V. Lesser. Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation, IEEE Transactions on Systems, Man, and Cybernetics, 21(5), 1167 - 1183, 1991.
- [2] D. Lindley, Making Decisions, 2nd Edition, Wiley 1985.
- [3] D. Gross, J. Allen and D. Traum, The TRAINS 91 Dialogues, TRAINS Technical Note 92-1, Computer Science Dept., University of Rochester, June 1993.