A Replanning Algorithm for a Reactive Agent Architecture

Guido Boella and Rossana Damiano

Dipartimento di Informatica
Cso Svizzera 185 Torino ITALY
guido@di.unito.it

Abstract. We present an algorithm for replanning in a reactive agent architecture which incorporates decision-theoretic notions to drive the planning and meta-deliberation process. The deliberation component relies on a refinement planner which produces plans with optimal expected utility. The replanning algorithm we propose exploits the planner's ability to provide an approximate evaluation of partial plans: it starts from a fully refined plan and makes it more partial until it finds a more partial plan which subsumes more promising refinements; at that point, the planning process is restarted from the current partial plan.

1 Introduction

In this paper we present a replanning algorithm developed for a reactive agent architecture which incorporates decision-theoretic notions to determine the agent's commitment. The agent architecture is based on the planning paradigm proposed by [3], which combines decision-theoretic refinement planning with a sound notion of action abstraction ([2]): given a goal and a state of the world, the planner is invoked on a partial plan (i.e. a plan in which some actions are abstract) and iteratively refines it by returning one or more plans which maximize the expected utility according to the agent's preferences, modelled by a multi-attribute utility function.

The decision-theoretic planning paradigm extends the classical goal satisfaction paradigm by allowing partial goal satisfaction and the trade-off of goal satisfaction against resource consumption. Moreover, it accounts for uncertainty and non determinism, which provide the conceptual instruments for dealing with uncertain world knowledge and actions having non-deterministic effects. These features make decision-theoretic planning especially suitable for modelling agents who are situated in dynamically changing, non deterministic environments, and have incomplete knowledge about the environment.

However, decision-theoretic planning frameworks based on plan refinement ([3]) do not lend themselves to reactive agent architectures, as they do not include any support for reactive replanning. In this paper, we try to overcome this gap, by proposing an algorithm for replanning for a reactive agent architecture based on decision-theoretic notions.

Since optimal plans are computed with reference to a certain world state, if the world state changes, the selected plan may not be appropriate anymore. Instead of planning an alternative solution from the scratch, by re-starting the planning process from the goal, the agent tries to perform replanning on its current plan.

D. Scott (Ed.): AIMSA 2002, LNAI 2443, pp. 183–192, 2002.

The replanning algorithm is based on a *partialization process*: it proceeds by making the current solution more partial and then starting the refinement process again. This process is repeated until a new feasible plan is found or the partialization process reaches the topmost action in the plan library (in this case, it coincides with the standard planning process).

We take advantage of the decision-theoretic approach on which the planner is based not only for improving the quality of the replanned solution, but also for guiding the replanning process. In particular, the planner ability to evaluate the expected utility of partial plans provides a way to decide whether to continue the partialization process or to re-start refinement: for each partial plan produced in the partialization step, it is possible to make an approximate estimate of whether and with what utility the primitive plans it subsumes achieve the agent's goal.
Then, the pruning heuristic used during the standard planning process to discard sub-optimal plans can be used in the same way during the replanning process to reduce its complexity.

2 The Agent Architecture

The architecture is composed of a *deliberation module*, an *execution module*, and a *sensing module*, and relies on a *meta-deliberation* module to evaluate the need for re-deliberation, following [9]. The agent is a BDI agent ([7]), i.e. its internal state is defined by its beliefs about the current world, its goals, and the intentions (plans) it has formed in order to achieve a subset of these goals . The agent's deliberation and redeliberation are based on decision-theoretic notions: the agent is driven by the overall goal of maximizing its utility based on a set of preferences which are encoded in a utility function.

The agent is situated in a dynamic environment, i.e. the world can change independently from the agent's actions, and actions can have non-deterministic effects, i.e., an action can result in a set of alternative effects. Moreover, there is no perfect correspondence between the environment actual state and the agent's representation of it.

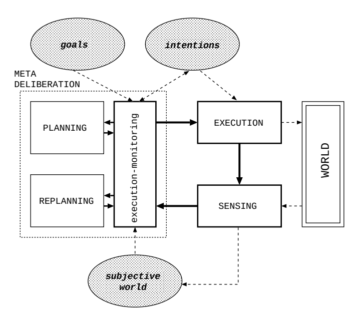

Fig. 1. The structure of the agent architecture. Dashed lines represent data flow, solid lines represent control flow.

In this architecture, intentions are not static, and can be modified as a result of re-deliberation: if the agent detects a significant mismatch between the initially expected and the currently expected utility brought about by a plan, the agent revises its intentions by performing re-deliberation. As a result, the agent is likely to become committed to different plans along time, each constituted of a different sequence of actions. However, while the intention to execute a certain plan remains the same until it is dropped or satisfied, the commitment to execute single actions evolves continuously as a consequence of both execution and re-deliberation.