Robot-TINO Advancements

(Position Statement)

Giuseppe De Giacomo, Luca Iocchi, Daniele Nardi, Riccardo Rosati
Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, 00198 Roma, Italy
{degiacomo,iocchi,nardi,rosati}@dis.uniroma1.it

1 Introduction

The goal of our work is to devise a principled and practically feasible realization of a KR&R approach to reasoning about actions in the realm of mobile robots. In [1, 2] we presented a formal setting for reasoning about actions and its implementation on the mobile robot Tino of the *Erratic* family, capable of integrating reacting behavior with action planning and execution.

Specifically, the basis of our proposal for reasoning about actions is provided by Propositional Dynamic Logics (PDLs). In this setting PDLs formulae denote properties of states, and actions (also called programs) denote state transitions from one state to another. The dynamic system itself is described by means of axioms. Two kinds of axioms are introduced, "static axioms", that describe background knowledge, and "dynamic axioms", that describe how the situation changes when an action is performed. A plan can be generated by finding a constructive existence proof for the state where the desired goal is satisfied. In a PDL setting a plan consists of a sequence of transitions, which leads to a state satisfying the goal.

The implementation has been obtained by relying on the tight correspondence that exists between PDLs and Description Logics (DLs). By exploiting this correspondence we have been able both to develop an interesting theoretical framework for reasoning about actions and to obtain an implementation that uses a knowledge representation system based on DLs. In particular, we have reinterpreted dynamic axioms by means of the so-called procedural rules. By relying on the epistemic interpretation of these rules, we have defined a setting which provides both an epistemic representation of dynamic axioms and a weak form of reasoning. In this way, we obtain a computationally feasible and semantically justified approach to deductive planning.

Our initial work has been mainly concerned with the position of the robot and its movement capabilities. We are currently investigating several extensions, that will enable Tino to address more complex scenarios. Such extensions include the implementation of new primitive actions, the definition of a special form of frame axioms, and the introduction of sensing actions.

2 New primitive actions

Tino is based on a two-level architecture, which combines a reactive control mechanism with a planning system. The control layer is based on a fuzzy controller [3] which is responsible for reactively executing primitive actions coming from the planner.

Inside the fuzzy controller, primitive actions are mapped into *behaviors*, which are basic control activities. Behaviors are distinguished in reactive ones, like avoiding obstacles, and (low-level) goal-oriented ones, like following a corridor. A blending mechanism is used to integrate reactive and goal-oriented behaviors, so that the robot can follow a corridor while avoiding obstacles.

In practice, each behavior is implemented by a set of fuzzy rules specifying control actions to be performed in determined situations. For example, the Avoid-Collision behavior includes a rule like

```
IF collision-right AND NOT collision-left THEN turn-left
```

where `collision-right` and `collision-left` are fuzzy variables and `turn-left` is the control action. An activity level computed from evaluation of rule antecedent is associated with the control action. The activity level measures how much the control action is desirable in the current situation.

Introducing new primitive actions for Tino amounts to designing and implementing new behaviors, i.e. writing fuzzy rules that approximate a desirability function characterizing the behavior. We have extended the set of primitive actions for Tino and the set of objects that can be described in the planning system. Now the robot can reach specific positions on the map, that are described to have significant properties.

Since the planning system can reason about new primitive actions and new objects, we can express more refined high level goals for the robot. For example, by describing state properties about a museum environment, we can give to the robot the goal "*guide me in a tour visiting pictures whose author is Raffaello*".

3 Frame problem

We have introduced in our framework a special form of frame axioms, namely:

$$KC \sqsubseteq \forall R.C$$

Although simpler than a standard dynamic axiom in our framework, such axiom is able to capture the persistence of the property C while executing the action R. The algorithm presented in [2] for computing the *first order extension* of a knowledge base, has been slightly modified to deal with frame axioms of this form.

From a practical viewpoint, we need an extension of the CLASSIC language to implement this algorithm. In particular, using the `FILLS` operator to express the successor state is a severe limitation in addressing the frame problem. In order to avoid to explicitly name the successor state, we use CLASSIC rules of the form

$$C \mapsto \exists R.\top \sqcap \forall R.D$$

to describe effects of actions, while rules like

$$C \mapsto \forall R.C$$

are used to represent persistency of properties.

The $\exists R.\top$ construct is treated in a special way: a new individual is created and added to the knowledge base, provided that no other individuals with the same properties are already present. Moreover, this extension allows for representing succinctly domains with an exponential number of states.

We also exploit the possibility we have in CLASSIC of defining a role hierarchy (that is, an action hierarchy) to reduce the number of frame axioms. For example, we can say that *all painting actions do not affect the position of the robot* by writing

$$Corridor \mapsto \forall paint.Corridor$$

where *paint* is the role representing a generalization of all painting actions.

4 Sensing and conditional plans

The possibility of representing the epistemic state of the agent allows for a very simple formalization of sensing (or knowledge-producing) actions. Roughly speaking, sensing actions are actions that only change the epistemic state of the agent without changing the actual world. An example of this kind of actions is the action *sense-door*, which requires the agent to check whether the door is open or closed.

Indeed, the epistemic operator K induces a straightforward representation of actions whose execution only changes the knowledge of the agent without affecting the state of the world. Most importantly, the minimal knowledge semantics of K in the epistemic framework embodies the principle of minimal learning, or default persistence of ignorance, which is a necessary property in order to correctly formalize sensing actions.

Suppose R is a generic sensing action whose effect is to let the agent know the truth value of the fluent D. Also, suppose P is the precondition for the execution of R. The effect of such a sensing action is represented in our framework by the following dynamic axiom:

$$KP \sqsubseteq \exists KR.\top \sqcap \forall KR.(KD \sqcup K\neg D)$$

We are currently addressing the problem of planning in the presence of sensing actions. In particular, we have shown that the computational complexity of the problem of plan existence does not increase when sensing actions are allowed, i.e. plan existence can be computed in polynomial space. We are currently extending our plan construction procedure for taking into account sensing actions. Notice that, since the formalism allows for reasoning in the presence of sensing, plans will include conditional actions.

References

[1] Giuseppe De Giacomo, Luca Iocchi, Daniele Nardi, and Riccardo Rosati. Moving a robot starting from a theory of actions. In *Proceedings of the AAAI-96 Workshop on Theory of Actions, Planning and Control*, pages 60–69, 1996.

[2] Giuseppe De Giacomo, Luca Iocchi, Daniele Nardi, and Riccardo Rosati. Moving a robot: the KR&R approach at work. In *Proceedings of the Fifth International Conference on the Principles of Knowledge Representation and Reasoning (KR-96)*, pages 198–209, 1996.

[3] A. Saffiotti, K. Konolige, and E. Ruspini. A multivalued logic approach to integrating planning and control. *Artificial Intelligence*, 76:481–526, 1995.