Logical Implementation of Uncertain Agents

Nivea de C. Ferreira, Michael Fisher, and Wiebe van der Hoek

Department of Computer Science, University of Liverpool, UK
{niveacf, michael, wiebe}@csc.liv.ac.uk

Abstract. We consider the representation and execution of agents specified using *temporal logics*. Previous work in this area has provided a basis for the direct execution of agent specifications, and has been extended to allow the handling of agent beliefs, deliberation and multi-agent groups. However, the key problem of *uncertainty* has not been tackled. Given that agents work in unknown environments, and interact with other agents that may, in turn, be unpredictable, then it is essential for any formal agent description to incorporate some mechanism for capturing this aspect. Within the framework of executable specifications, formal descriptions involving uncertainty must also be executable. The contribution of this paper is to extend executable temporal logic in order to allow the representation and execution of uncertain statements within agents. In particular, we extend the basis of the METATEM temporal framework with a *probabilistic belief* dimension captured by the recently introduced $P_{\mathcal{F}}KD45$ logic. We provide a description of the extended logic, the translation procedure for formulae in this extended logic to an executable normal form, and the execution algorithm for such formulae. We also outline technical results concerning the correctness of the translation to the normal form and the completeness of the execution mechanism.

1 Introduction

The logical characterisation of agent-based systems is now a well established area [6,19,21]. Such a characterisation can not only provide an unambiguous semantics for agents but can also allow key techniques such as formal specification to be used in the analysis of agent-based systems. An important aspect is the direct execution of these formal agent specifications. Here, a model satisfying the agent specification is extracted, with the process of extracting such a model corresponding to an execution. This is analogous to the use of Prolog in classical logic whereby the system searches for a model satisfying the specification. Such direct execution also provides a strong link between the semantics of an agent and its implementation, something that is often lacking in contemporary agent programming frameworks [5].

Given that agents essentially work in unpredictable environments, and interact with other agents that may, in turn, be unpredictable, it is vital for any formal description of an agent to incorporate some mechanism for capturing uncertainty [14] on a level that is more fine-grained than assuming the agent has just knowledge, or even beliefs. In dealing with uncertainty one can choose among many existing approaches, often categorised as either numerical or symbolic. We here opt for a framework incorporating *Probability Theory* as, at least semantically, this allows us to remain close to the *possible worlds paradigm*. This possible-worlds view is by far the most popular way (see also [20]) to model the varieties of agent we require (cf. [19]).

C. Bento, A. Cardoso, and G. Dias (Eds.): EPIA 2005, LNAI 3808, pp. 536–547, 2005.

In our work, we choose to extend a standard temporal logic, to which execution has previously been applied, with an added element of uncertainty. Thus, we produce a new logic from the combination of standard temporal logic with a novel logic of *probabilistic belief*, $P_F KD45$ [8]. This provides a simple, and intuitive, basis for specifying agents uncertain about their environment and their choices.

Once we have the temporal specification of an uncertain agent, an implementation can be developed in a number of ways, for example by refinement to a standard programming language or the automatic synthesis of an automaton [17,16]. Such a synthesis approach is necessarily complex, generating an implementation that is guaranteed to satisfy the specification in all environments. However, the route we choose is to directly execute the temporal specification in order to provide an implementation. Note that this just involves searching for *one* acceptable execution, which is generally less complex (and often much quicker) than attempting full synthesis [2]. Our approach to direct execution extends the METATEM programming language [3], which executes purely temporal statements, and can be utilised to animate agent specifications.

The contribution of this paper is to devise a logical framework for uncertainty in agents that (1) is conceptually clear and simple, (2) can be used in modal (intensional) logical specifications of agents, and (3) can be added on top of METATEM to give a rich but still executable temporal/doxastic logic. The executable framework we develop is called PROTEM.

The rest of this paper is organised in the following way. Section 2 presents a brief overview of the METATEM Framework and the $P_F KD45$ logic. We then describe how these systems can be combined forming the basis for uncertain agent implementation, with the associated normal form and execution mechanism being subsequently described in Sections 3 and 4, respectively. Finally, in Section 5, comments on related work, potential applications, and future research, together with concluding remarks, are provided.

2 A Temporal Doxastic Logic

2.1 Temporal Basis of METATEM

In previous work on the METATEM framework, the representation of simple dynamic agents using temporal logic [12], the representation of deliberation within these agents, and an extension to agents that have beliefs [10] were considered. In the work presented in this paper we provide an extension of this framework, using an appropriate logical formalism to incorporate uncertainty.

In METATEM, logical formulae represent an agent's specification. The framework allows the animation of an agent's specification by direct execution of these formulae, essentially providing an implementation of the agent's behaviour. This approach follows the *imperative future* paradigm [4], and applies an iterative *forward-chaining* process to a set of temporal formulae in a specific normal form in order to (attempt to) construct a model for the specification.

Temporal logic is an extension of classical logic in which temporal order is important. Thus, statements are not just true or false, but are true or false dependent upon the moment in time at which they are evaluated. Typical operators of the temporal logic