Agents in Proactive Environments

Dov Gabbay[1] and Rolf Nossum[2] and Michael Thielscher[3]

[1] Department of Computing
Imperial College
London (UK)
[2] Department of Mathematics
Agder College
Kristiansand (Norway)
[3] Intellectics, Department of Computer Science
University of Technology
Darmstadt (Germany)

Abstract. Agents situated in proactive environments are acting autonomously while the environment is evolving alongside, whether or not the agents carry out any particular actions. A formal framework for simulating and reasoning about this generalized kind of dynamic systems is proposed. The capabilities of the agents are modeled by a set of conditional rules in a temporal-logical format. The environment itself is modeled by an independent transition relation on the state space. The temporal language is given a declarative semantics.

1 Introduction

To motivate what follows, we discuss some aspects of scenarios involving agents and their environment, both capable of changing the state of affairs of the world. The environment in these scenarios is evolving freely, whether or not there is any action on the part of the agents. This seems realistic, many real-world scenarios are like that. This contrasts with the scenario classes that have mainly been studied, where the environment is perceived as a purely reactive mechanism; e.g. in [San94] pp. 16-17 we find: "The definitions in this book will be made in such a way that if no action is invoked by the ego, then the world will advance by one single timestep while keeping all feature-values unchanged." Scenarios like the following go beyond this idealistic view of the world pausing whenever an agent decides not to act:

> An interest-bearing bank deposit accrues interest from time to time, increasing the balance of the account. The owner of the account can meanwhile make deposits and withdrawals, not necessarily synchronized with the times at which interest is added. The rate of interest on such accounts varies with the bank, with the size of the deposit, with inflation, trading balance and other economic parameters in that particular country, and ultimately on the current state of the world economy.

We propose a formal framework for modeling dynamic environments with situated agents whose actions influence the development in the course of time. We start with a brief discussion on inert vs. transient state components in Section 2. In Section 3, we introduce a temporal execution language named TEAL. It allows for the specification of both proactive environments and the effects on it caused by the performance of actions. Section 4 contains a brief discussion on the operational semantics, followed by the sketch of a declarative semantics in Section 5 on the basis of an abstract, general model structure for specifications of dynamic systems developed in [Thi95].

2 Inertia vs. transience

We take the state of affairs in the world to consist of a countable set F of atoms (called *fluents*), some of which are designated as *inert*. These are the fluents that are thought of as stable if not explicitely changed. The agent, as well as the environment, can change them. For example, the lights in a room are inert; when the switch is flicked, the status of the light changes, and remains inert until the bulb burns out, or the switch is flicked again. The lights in the stairways of some buildings are fitted with a time-delay circuit, which turns them off when they have been on for a while; this can be viewed as the environment acting on the fluent, independently of the agent that turned it on.

The non-inert fluents are *transient*, i.e. lasting for a single time unit. This is an approximation to the transient nature of events like the sound of a doorbell, or a flash of lightning.

Some transient fluents are designated as *actions*, and are carried out by agents or their environment. The flicking of a switch is an action, and therefore transient, but the status of the light, which is an effect of the flicking, is inert. Ringing a doorbell is an action, and so is setting off a lightning flash in a thunderstorm, but the latter is an action available only to the environment.

A flash of lightning can have drastic effects, which can greatly influence the further evolution of agent scenarios, for instance by rendering some courses of action impossible for some agents. Flashes of lightning are in general not provoked by any action on the part of an agent, so it would be unnatural to have a model where the environment was restricted to responding directly to actions of an agent. On the other hand, the agent can influence the evolution of the environment, for instance by erecting a lightning rod.

3 TEAL—a Temporal Executable Action Language

We proceed to give a formula language similar to that in [GN95], with an informal interpretation of some formulas as describing the actions of an agent. A precise formal semantical interpretation in terms of the logic of dynamical systems (in the sense of [Thi95]) is sketched in a later section.

The language elements are the following. Propositional fluents, denoted p, q, r, etc; classical connectives with their usual semantics, as well as temporal modalities $\bigcirc$ ("tomorrow"), $\bullet$ ("yesterday"); and actions, which are denoted by atomic terms like $flick\_switch$, and ground terms like $make\_deposit(250)$. There is a distinguished predicate $exec(...)$ on actions, generating atomic propositions from action terms.

Fluents and negated fluents are called literals. Literals are combined with connectives and modalities into formulas. Literals with 1 or more $\bullet$'s on them are called past literals, and literals with 1 or more $\bigcirc$'s on them are called future literals. Clauses of the form

$$past\ literals \wedge exec(a) \rightarrow future\ literals$$

are called action rules. Action rules are interpreted by evaluating the past literals, and if they are true, recording the action as having been carried out at the present time, and performing the future literals at the appropriate points in time. More precisely, a rule

$$\bigwedge_i \bullet^i c_i \wedge exec(a) \rightarrow \bigwedge_j \bigcirc^j e_j$$

is eligible at time t if $t \models \bigwedge_i \bullet^i c_i$, cfr. the operational semantics below. Whether it is actually carried out depends on constraints, and on possible conflicts with other rules.

4 Operational semantics

For simplicity we take integer time with finite past. Action rules can be executed at points of time, resulting in a change in the future state.

For the moment, we disregard possible conflicts between different rules, and concentrate on the operational semantics of executing a single rule at a certain timepoint t. Issues of competing clauses, and in the case of concurrency, conflicting ones, are dealt with elsewhere.

We rely on an underlying notion of truth of fluents at timepoints. Intuitively, we may visualize the development of the state through time as a two-dimensional matrix, indexed in one direction by fluents and in the other by time. For finite-past integer time, the matrix will be infinite in one direction, that of advancing time.

Thus, operationally speaking, an execution model is a set of fluents together with a mapping $\models$ of timepoints and fluents to truth values. For a fluent f and a timepoint t, $t \models f$ is either true or false. This matrix can be maintained as a temporal database, e.g. using the techniques of [McB93]. Formulas composed by propositional connectives or temporal modalities are interpreted by the usual inductive definitions:

- $t \models \neg\varphi$ iff not $t \models \varphi$
- $t \models \psi \rightarrow \varphi$ iff $t \models \psi$ implies $t \models \varphi$
- $t \models \bullet\varphi$ iff $t-1 \models \varphi$
- $t \models \bigcirc\varphi$ iff $t+1 \models \varphi$

To execute a rule at time t, the preconditions are computed against the temporal database. If they are met, the action fluent $exec(a)$ is recorded as true at time t, and the effect fluents are entered into the database at the appropriate times as indicated by their temporal prefixes.

There is no guarantee that these updates of the temporal database can be made consistently. Conflict resolution between rules that have inconsistent effects, can be dealt with along the lines of [BT97].

5 Declarative Semantics

As the basis for a declarative semantics we adopt a general abstract framework proposed in [Thi95], which provides formal means for declarative characterization of a variety of action languages. This framework is commonly referred to as "Logic of Dynamic Systems," abbreviated LoDS.

Like TEAL, LoDS grounds on the paradigm that state transitions in a dynamic system naturally occur while time passes by, and one or more agents have the possibility to direct the development of the system by executing actions. LoDS supports concurrency of actions and events, and allows to formalize nondeterministic actions and delayed effects. The semantics includes the conflict resolution strategy of [BT97].

Domain and scenario specifications in terms of LoDS are given formal notions of interpretations, models, and entailment. The mapping of TEAL programs to LoDS domains thus enables us to precisely characterize what conclusions a specification in TEAL allows. The details on this mapping can be found in [GNT97].

References

[BT97] S.-E. Bornscheuer and M. Thielscher. Explicit and implicit indeterminism: Reasoning about uncertain and contradictory specifications of dynamic systems. *Journal of Logic Programming*, 31(1–3):119–155, 1997.

[GN95] D. Gabbay and R. Nossum. A temporal executable agent language. *Workshop on Executable Temporal Logic at IJCAI'95*, Montreal, 1995.

[GNT97] D. Gabbay, R. Nossum, and M. Thielscher. Agents in Proactive Environments. Imperial College Research Report, 1997.

[McB93] P. J. McBrien. Principles of implementing historical databases in RDBMS. In M. Worboys and A. F. Grundy, editors, *Advances in Databases*, pp. 220–237, Keele, UK, 1993. Springer LNCS 696.

[San94] E. Sandewall. *Features and Fluents. Volume 1: The representation of knowledge about dynamical systems.* Oxford University Press, 1994.

[Thi95] M. Thielscher. The logic of dynamic systems. In C. S. Mellish, editor, *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 1965–1962, Montreal, 1995. Morgan Kaufmann.