Available online at www.sciencedirect.com
SCIENCE DIRECT®
Electronic Notes in Theoretical Computer Science
ELSEVIER Electronic Notes in Theoretical Computer Science 134 (2005) 55–75
www.elsevier.com/locate/entcs

Propositional Statecharts for Agent Interaction Protocols

H. R. Dunn-Davies and R. J. Cunningham[1]

Department of Computing
Imperial College
London, UK
SW7 2BZ

S. Paurobally[2]

Department of Electronics and Computer Science
University of Southampton
Southampton, UK
SO17 1BJ

Abstract

Current representations of interaction protocols either concentrate on ease of understanding and neglect the need to specify protocols unambiguously (e.g. [2]), or concentrate on formal rigour and neglect usability issues (e.g. [21,10]). The on-going development of propositional statecharts has been motivated by the need to provide a notation which satisfies both of these requirements. Here we give a brief description of propositional statecharts, a restricted form of David Harel's statechart formalism [13] designed to represent agent interaction protocols. Statecharts are a popular extension of Euler Diagrams and are widely used as part of the UML framework. Propositional statecharts take advantage of the intuitive nature of the statechart formalism, and include labelling conventions to ensure that interaction protocols are defined unambiguously while remaining easy to understand.

Keywords: Agents, Agent Communication, Euler Diagrams, Higraphs, Propositional Statecharts, Statecharts, UML

[1] Email: hdd2@doc.ic.ac.uk, rjc@doc.ic.ac.uk
[2] Email: sp@ecs.soton.ac.uk

1571-0661/$ – see front matter
doi:10.1016/j.entcs.2005.02.020

1 Introduction

At the turn of the millennium the field of software agent research had become one of the most active areas of R&D activity in computing [30]. Since then agents have been widely tipped as the logical next step in the evolution of computer science (e.g. [8]). Much of the intuitive appeal of the agent paradigm stems from the idea that agents could solve problems through cooperation and negotiation, as people do, therefore providing the potential for automating many of the tasks which currently still have to be performed by humans. The potential ability of agents to communicate with each other in a goal directed manner forms an integral part of this idea.

Researchers such as Greaves et al. [12] have identified three characteristics which must be shared by different agents to enable them to communicate effectively: a shared infrastructure for message passing, a common ontology, and a common agent communication language. These researchers have also noted that in all but the very simplest interactions it is necessary for each agent to have a common understanding of the set (or collection of sets) of rules underlying the interaction. A set of rules governing an interaction is commonly known as an Interaction Protocol.

The problems of ensuring that the agents involved in an interaction share a common infrastructure for message passing and a common agent communication language have been addressed by the design of general purpose frameworks for message passing (such as the ICM, described in [7]) and languages (such as KQML [9], or FIPA ACL, described at `http://www.fipa.org`[3]). However the design of a single generic protocol suitable for every interaction is less realistic. Agents may be expected to interact in a wide range of different circumstances, each of which may call for the use of different protocols. For example an agent may be required to communicate with a single agent, or with a group of agents at the same time, and could be required to participate in a tightly defined interaction such as an auction, which could take any one of a variety of forms. For this reason it would be extremely difficult to define a single protocol that would be suitable for every situation, making it necessary to find another way of ensuring that within any particular transaction, each agent is using exactly the same protocol.

Current agent interaction protocols are designed independently of agents that implement them. A library of such protocols can be found on FIPA's website (`http://www.fipa.org`). It is intended that agent designers implement these standard protocols when constructing software agents to enable

[3] FIPA is a non-profit organisation aimed at producing standards for the interoperation of heterogeneous software agents.