Speech-Act Based Communication: Progress in the Formal Semantics and in the Implementation of Multi-agent Oriented Programming Languages

Álvaro F. Moreira[1], Renata Vieira[2], and Rafael H. Bordini[1]

[1] Institute of Informatics
Federal University of Rio Grande do Sul
CP 15064, CEP 91501-970, Porto Alegre – RS, Brazil
{alvaro.moreira,r.bodini}@inf.ufrgs.br
[2] Faculdade de Informática
Pontifícia Universidade Católica do Rio Grande do Sul
CP 275, CEP 93022-000, Porto Alegre – RS, Brazil
renata@pucrs.br

Abstract. In this paper we revisit the motivations and the initial developments that led to our DALT 2003 paper *Extending the Operational Semantics of a BDI Agent-Oriented Programming Language for Introducing Speech-Act Based Communication*. We then discuss our own follow-up work which consisted in formally defining a larger set of speech-act based performatives and deploying them in *Jason*, a fully-fledged implementation of AgentSpeak. Subsequent research referring to the computationally grounded semantics of speech-act based agent communication that we introduced in that paper is also discussed.

1 Introduction

In [13], we introduced an operational semantics of speech-act based communication for AgentSpeak(L) [16], defining semantic rules for handling some of the performatives defined by Searle [19]. We were motivated mainly by two facts: first we realised that, at that time, the semantics for agent-oriented programming languages was given only at a very abstract level and important social and pragmatical aspects, such as inter-agent communication, were completely neglected. Second, previous attempts at giving semantics for agent communication were based on the approach in [11], asserting pre and post conditions on mental states of agents expressed in the modal logic introduced in [7]. Although that was a well-established way of defining the meaning of speech-act based communication, it lacked a computational interpretation and could not, therefore, be used for guiding the implementation of programming languages.

Given that state of affairs, and also our interest in developing AgentSpeak(L) into a core language for investigating agent-oriented languages (both on their formal and practical aspects), we endeavoured to define a computationally grounded

C. Sakama et al. (Eds.): DALT 2011, LNAI 7169, pp. 111–116, 2012.

semantics for speech-act communication in a way that could be used as a direct guide for the implementation of multi-agent programming languages.

The operational semantics given in our DALT-2003 paper was an extension of the formal semantics of AgentSpeak(L) which was first presented in [14] and further developed in [3]. We started by considering the performatives *Tell*, *Untell*, *Achieve*, *Unachieve*, *TellHow*, *UntellHow* and by formalising only the effects of receiving these illocutionary forces on the computational interpretation — formalised in [3] — of beliefs, desires, and intentions of AgentSpeak agents.

When working in that direction, we realised it was important to keep track of the source of the messages being exchanged along an agent's execution. For that purpose, all the atomic predicates in the belief base were annotated with their *source of information*. Those annotations can be of 3 different types: *self* when it comes from the internal plan execution of the agent, *percept* when it derives from the agent perception of its environment, and it can also be an agent's ID when the message has been sent by another agent.

In our semantics, the performatives *Tell* and *Untell* affect the belief base of the agent by respectively adding and removing beliefs from it. The performatives *Achieve* and *Unachieve* add new events in the agent's set of events. These events, later when handled in an agent reasoning cycle, might have effects on the intentions of the receiving agent (i.e., pursuing a new intention with *Achieve* and dropping an intention with *Unachieve*). Plans can also be communicated with *TellHow* and *UntellHow*, for respectively adding and removing a plan from the agent's plans library.

At around the same time, some of us were working on a fully-fledged implementation of an extension of AgentSpeak (that eventually culminated in platform called *Jason* [4]). It became clear that the work on the semantics of communication was crucial and had to be pursued further.

2 Improvements to the Original Proposal

Following the DALT paper, in [22] we extended the formal treatment to a larger set of performatives. Besides those already considered, we also defined the operational semantics for sending and receiving *AskIf*, *AskAll*, and *AskHow* messages.

The sender of a message with performative *AskIf* gets blocked (i.e., the intention that originated the message sending has its execution suspended) until it gets a reply saying whether the content of the message is true for the receiving agent. Similarly, the intention that originated an *AskAll* gets suspended until it gets the set of answers that make the content of the message true for the receiver. The intention that gave rise to an *AskHow* message is suspended until it obtains a set of plans that match, for the receiver, the triggering event in the message's content.

We also generalised the content of the messages allowing agents to send and receive sets of predicates and sets of plans instead of only a single predicate and a single plan (as it was the case in the DALT original paper). As before, our