LOGIC TUPLE SPACES FOR THE COORDINATION
OF HETEROGENEOUS AGENTS

Abstract. This work presents $\text{ACLT}$, a coordination model aimed to combine and coordinate heterogeneous agents by means of a communication abstraction inspired to the Linda model, but rooted in a logic framework. The twofold interpretation of a logic tuple space both as a message and as a knowledge repository naturally induces a categorisation of agents as logic and non-logic. While non-logic agents adopt the basic Linda kernel, logic agents exploit the full power of the $\text{ACLT}$ model, which supports deduction and reasoning over the content of the tuple space. By providing a conceptual framework where logic inference and temporal tuple space evolution coexist, $\text{ACLT}$ provides a suitable environment to build heterogeneous multi-agent systems, where hybrid agent architectures can be designed, integrating reasoning capabilities together with reactive behaviours.

1. Introduction

When combining independent and possibly heterogeneous agents to build a multi-agent system, one of the most critical issues concerns the definition of a proper coordination and synchronisation model which adequately supports agent interaction. In fact, the notion of agent is usually intended to be a higher-level concept than just a system process, particularly emphasising problem solving and coordination aspects, as well as autonomous behaviour, which ask for high-level communication metaphors. A well-known approach to this issue consists of defining some form of shared memory abstraction to be used as a message repository, and a communication protocol allowing agent competition and cooperation in a multi-agent framework. Several such coordination models have been proposed in literature, based on abstractions like blackboards (Englemore et al., 1988) and tuple spaces, like Linda (Gelernter et al., 1985; Gelernter et al., 1992; Gelernter et al., 1989), Shared Prolog (Brogi et al., 1991), and Polis (Ciancarini, 1994;
However, the design and the implementation of multi-agent systems raise case-specific issues, including the development of suitable coordination strategies for effective problem solving, protocols to support reasoning activities (also concerning inter-agent communication), negotiation and conflict detection, resolution mechanisms, etc. These requirements call for agents with an enhanced capability of symbolic computing, which be able to manipulate knowledge and perform inference operations.

As a result, there is a growing need for enriched communication abstractions, which allow coordination and deduction aspects to coexist, supporting higher-level forms of interaction than raw information exchange (like message-passing or signals). This is the case, for instance, when agents are used to (incrementally) build and store in a knowledge repository a (possibly partial) representation of a given application domain, where the ability of deducing new facts which have not been explicitly generated, exchanged, or stored, becomes relevant.

The $\textit{ACCT}$ model (Agents Communicating through Logic Theories, first proposed in (Omicini et al., 1995)) is founded on a first-order logic-based approach, whose key idea is to assume the logic theory as a communication abstraction, supporting (and combining) Linda-like communication operations as well as logic operations based on don't know non-determinism. As a consequence, reactive behaviours may be naturally integrated with high-level symbolic activities based on reasoning and planning, leading to those hybrid architectures which currently represent one of the main goals in fields like robotics (Lyons et al., 1992; Zanichelli et al., 1994).

In fact, while low-level agents handling basic tasks and/or real-time jobs such as sensor data acquisition will just exploit the $\textit{ACCT}$ communication abstraction as a synchronisation device, high-level agents, encapsulating higher-level coordination policies and system strategies, will be able to interpret the tuple space also as a logic theory, performing logic inferences and deductions based on the available knowledge.

The main contribution of this work is to show how the interpretation of the communication abstraction both as a simple communication device, and as a logic theory containing knowledge over the application domain, suggests a natural criterion to conceptually categorise two kinds of agents, based on the way such abstraction is used. As a result, each kind of agent may interact with the others at its own abstraction level, achieving all advantages of coordination and synchronisation while maintaining its autonomy from the viewpoint both of its implementation technology and of its computation model.

This work is structured as follows. Section 2 introduces the basics of the $\textit{ACCT}$ coordination model, by discussing the assumptions, the language