1

Time and Space Coordination of Mobile Agents

Gabriel Ciobanu

“A.I.Cuza” University, Faculty of Computer Science
Blvd. Carol I no.11, 700506 Iași, Romania
Romanian, Academy, Institute of Computer Science
gabriel@info.uaic.ro

Summary. We study the evolution of agents able to move by using local knowledge and various migration means. We present two models of distributed systems with an explicit notion of location and some quantitative notions including time and capacity. We use these models for controlling mobility in open distributed systems by means of timers, bounded capacities and a simple migration primitive.

1.1 Introduction

The technology of agent systems, both hardware and software, is rather advanced. However design principles and techniques to define and verify their correct behaviour are at a more primitive stage. One approach in modelling the system behaviour has been the design of formal calculi in which the fundamental concepts underlying agent systems can be described and studied. In this paper we propose two formalisms for describing the behaviour of mobile agents in a distributed world. They are based on two existing formalisms called distributed \(\pi\)-calculus and ambient calculus, to which we add a network layer, a migration action, and other quantitative notions useful for coordination in time and space of the mobile agents.

A first approach could be given by defining a simple formal language for describing the systems in terms of their structure, namely how they are constructed from individual interacting agents. A semantic theory is defined in order to understand the behaviour of systems described in such a language, in terms of their ability to interact. Here a system consists of a finite number of agents which intercommunicate using a fixed set of named communication channels. This set of channels constitutes a connection topology through which all communication takes place; it includes both communication between agents, and between the system and its users.

The current agent systems are highly dynamic. Moreover, it is possible to create new communication links with other entities, and perhaps relinquish existing links. The \(\pi\)-calculus [10] and ambient calculus [4] are two formalisms seeking to address at least some dynamic aspects of such agents. The \(\pi\)-calculus includes the dynamic generation of communication channels and thus allows the underlying connection
topology to vary as systems evolve; it also allows private communication links to be established and maintained between agents. Ambient calculus is more oriented to dynamic aspects, working with ambients which represents units of movement.

Many concepts of the distributed systems are at most implicit (if not existing) in the \( \pi \)-calculus. Perhaps the most useful missing concept is that of domain (location), understood generally as a locus for computational activity. Thus one could view a distributed system as consisting of a collection of domains, each capable of hosting computational processes (agents), which in turn can migrate between domains. We use an extension of the \( \pi \)-calculus in which these domains have an explicit representation as locations. Distributed \( \pi \)-calculus is a formalism for agent systems in which dynamically created domains are hosts to resources which may be used by agents, and agents reside in domains, migrating eventually between domains for the purpose of using locally defined resources. Moreover, we use types to manage access control to resources in distributed systems. A domain may wish to restrict to certain agents the access to certain resources; we can think of resources having capabilities associated with them. Then domains may wish to distribute selectively to agents such capabilities on its local resources, and have agents manipulating these capabilities. Therefore in distributed \( \pi \)-calculus a system consists of a collection of domains, hosting agents, which can autonomously migrate between domains. These domains also host local channels on which agents communicate; but more generally these may be used to model local resources, accessible to agents currently located at the host domain. We describe the mobile agents by using a new primitive \( \text{go } l \). \ P \) which enables migration between domains. If an agent executing \( \text{go } l \). \ P \) is currently residing at \( k \), it migrates to the domain \( l \) and continue there with the execution of \( P \).

We define some quantitative ingredients over the systems described by the distributed \( \pi \)-calculus in order to coordinate the mobile agents in time and space. We add timers to distributed \( \pi \)-calculus. We assume a global notion of time, but we rather use a relative time of interactions given by timers. The global clock advances the time, and the interactions happen whenever the involved resources are available according to an interval of time given by timers of the interacting components (timers define timeouts for various resources, making them available only for a determined period of time). Using timers, we can control the concurrency of the components, and we can select between different choices in the system evolution. This provides a natural and flexible synchronization technique able to integrate and regulate dynamically the possible evolutions of the components. In a second formalism given by an extension of the mobile ambients, interaction in space relates to the ability to identify a specific domain in which a required and available resource exists. We use timers and timeout recovery processes over mobile ambients such that the resulting formalism provides a flexible coordination even in an open distributed environment where a number of possibly unknown entities are interacting. Several challenges related to these systems include finding the appropriate mechanisms to deliver adequate responses to time-critical demands, appropriate interaction methods, a decentralized control taking into account the time and space resources.