Hybrid approach for mobile agent specification

Mâamoun Bernichi, Fabrice Mourlin
Paris 12 University
LACL
Créteil, France
bernichi@gmail.com, fabrice.mourlin@wanadoo.fr

***Abstract* — Agent based systems are rapidly gaining prominence in the industry. These agent systems are all heterogeneous in nature and usually not interoperable with each other. The agents have their own interface and new project means new approach with an ad hoc solution. Specification languages already exist for distributed systems even for mobile description such that pi calculus, join calculus or kell calculus. All these calculus are dedicated to software description. But architecture plays also a strategic role. Architecture constraints involve specific deployment of agent over network. Also, we defined a specification based on a two level approach: agent level and architecture level.**

The agent level is based on a famous process algebra called higher order pi calculus language defined by Robin Milner. Mobile agents are described as recursive terms where each term is a part of agent's behavior. As another layer, architecture level described location where agents can be deployed or where mobile agents can migrates. We use another process algebra called Safe Distributed pi calculus language defined by Matthew Hennessy.

Both levels are useful to check properties about initial deployment or about resource accessibility. Also, we defined tools to extract information from formal specifications and to build synthetic view such as deployment steps or navigation views.

***Index Terms* — mobile agent, formal specification, architecture specification.**

I. Introduction

IN the context of applications exploiting code mobility, the mobile agent paradigm is one of the most promising technologies to build distributed applications in heterogeneous environments. Mobile agents are autonomous software entities with the capability of dynamically changing their execution environments in a network-aware fashion and roaming through network nodes to carry out tasks on behalf of users. In this way, mobile agent systems constitute a middleware capable of supporting distributed and dynamic applications based on mobile agents. The main advantage of mobile agents is that they can significantly save bandwidth, by moving locally to the resources they need and by carrying the code to manage them. Moreover, mobile agents can deal with non-continuous network connection, and as a consequence they intrinsically suit mobile computing systems [1].

Mobile agent applications are often built with a pragmatic approach. There are a lot of mobile agent frameworks such as JADE [2], MAFEH or MEIP [3] which help developers to create an application based on basic modules. Also, validation step of the application is not really structured and no property can be set or proved about run time.

Formal models are needed for careful description and reasoning about agents, just as for other kinds of distributed systems. Important issues that arise in modeling agent computation are: agent communication, dynamic creation and destruction of agents, mobility, and naming. Our past experience about formal specification, has led us to choose a process algebra called higher order pi calculus [4] for description of mobile agent. Robin Milner has written a complete operational semantics which allows us to compute semantics states from a given specification.

But this approach is not sufficient; some features can be described easily with this process algebra. For instance, when an agent moves from one node to another one, it is not easy to specify a negotiation protocol between incoming agent and agent host. We can want to express that a mobile agent is accepted if and only if it does not access to a specific local resource. Local resource access and more generally security control are always complex rules which depend on location. It means the place where an agent is precisely on a network. We decided to use process algebra for that kind of description. We use safe distributed pi calculus language [5] for the definition of architecture and resource access.

So, we present a work about formal specification of mobile agent system. Our approach is based on two levels: agent level and architecture level. For a system, we have two sets of specifications based on two different extensions of pi calculus. This common origin allows us to use closely these specifications and to build observation about agent navigation, resource availability. We developed tools for parsing specification and for transformation of input description into more synthetic observations. Several observations are built based on semantics steps such as navigation path, mobile agent acceptance or communication achievement. These properties are first metrics which lead software designer

This work was realized at the L.A.C.L. (Laboratory of Algorithm, Complexity and Logics of Paris 12 university, France).

Mâamoun Bernichi is with the L.A.C.L., Paris 12 university, France (e-mail: bernichi@gmail.com).

Fabrice Mourlin., is with the L.A.C.L., Paris 12 university, France corresponding author to provide phone: +33-615051551; fax: : +33-145170350; e-mail: fabrice.mourlin@wanadoo.fr)

towards a more adequate prototype.

REFERENCES

[1] Karnik, N. M. and Tripathi, A. R. (1998), "Design Issues in Mobile-Agent Programming Systems", IEEE Concurrency, Vol. 6, No. 3, pp. 52-61, July-September.

[2] Fabio Bellifemine, Agostino Poggi, and Giovanni Rimassa. "JADE - A FIPA-compliant agent framework", http://sharon.cselt.it/projects/jade, 2000.

[3] R.-S. Chen, C.C. Chen, C.-C. Chang, "A Java-based mobile agent framework for enterprise applications", International Journal of Computer Applications in Technology 2002 - Vol. 15, No.6 pp. 276-286.

[4] Milner, R. "Communicating and Mobile Systems: the pi-Calculus", Cambridge University Press. (1999).

[5] M.Hennessy and J. Rieley. "Resource access control in systems of mobile agents" in U. Nestmann and B. C. Pierce, editors, Proc. of HLCL '98: High-Level Concurrent Languages volume 16.3 on ENTCS. Elsevier 1998