Mobile Agent Organizations

Giacomo Cabri, Letizia Leonardi, Marco Mamei, Franco Zambonelli

Dipartimento di Scienze dell'Ingegneria – Università di Modena e Reggio Emilia
Via Vignolese 905 – 41100 Modena – ITALY
name.surname@unimo.it

1. Introduction

The issues related to the modeling and the implementation of agent-based Internet applications, as well as the issues related to the impact of the agent-paradigm in the software development, are still being analyzed and studied along several dimensions. *Mobility* is one of such dimensions, and it is certainly one of the most debated. We think that mobility – other than a middleware mechanism – is also a necessary abstraction to model both the network-aware activities of autonomous application components – i.e., from a broad perspective, agents – as well as the presence, in applications, of (software running on) mobile computing devices. In other words, mobility is not only a useful feature, but it can be considered an intrinsic characteristic of Internet agents and a useful abstraction for the high-level design of wide-area and wireless applications.

Unfortunately, mobility of agents does not come for free. Among several problems it introduces, one of them relates to the handling of the agents coordination activities, and it affects both the design and the development of applications. In general, to proceed with their execution, agents may be in need of both accessing resources distributed in a set of independently managed sites and interacting with other agents, not necessarily of the same application. Therefore, these coordination needs introduce all the typical problems of open systems, such as heterogeneity of languages and protocols, opportunistic behavior in interactions and security. Finally, agents that are part of a specific multi-agent application may require their coordination activities to occur according to specific application needs, despite the different characteristics of the Internet sites where they execute and constraints there imposed.

2. Organizational Contexts

To face the above problems within a uniform framework, we propose modeling the Internet as a multiplicity of *local interaction contexts*, representing the logical place in which agent coordination activities occur, which can change depending on the agents' movement. Depending on its current position, an agent situates on a given interaction context and there will be enabled to access local resources and to interact with local executing agents. However, in our framework, the interaction context is considered more than simply a logical place, indeed is an *active context*, in charge of mediating all coordination activities of application agents and of enacting specific local rules on the agent's coordination activities. Such a modeling promotes thinking mobile agents as organizational entities, moving across local *organizational contexts*, each enacting specific organizational rules on local coordination activities. Moreover, by conceiving agents as capable of configuring the activities of interaction contexts, one can think at mobile agents as entities that still belong to their original application/organization, and that can enact application-specific organizational rules on their coordination activities, independently of their current position (see Figure 1).

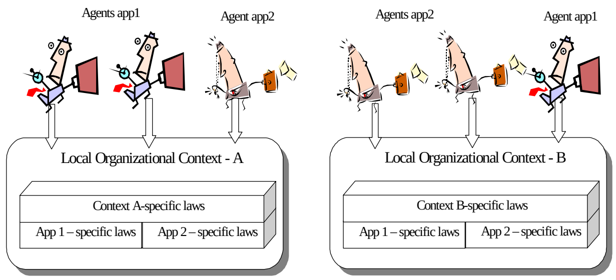

Fig 1: Local organizational contexts

3. Impact on Design and Development

The separation of concerns between intra-agent computational issues and inter-agent organizational ones promoted by our organizational framework simplifies the analysis and the high-level design of mobile agent applications.

Without the ambition of introducing a detailed methodology, we can sketch the general guidelines for the identification of intra-agent and inter-agent issues. These guidelines are coherent with the ones identified in previous works in the area of agent-oriented software engineering and methodologies.

From the point of view of application developers, the design of a mobile agent application can be organized as follow:

- at the intra-agent level, one has to analyze the global application goal and decompose it into sub-goals. This process should lead to the identification of the agents that will be instantiated in the application, and of their relevant internal characteristic. Such characteristics may include both the functional ones required for the achievement of the assigned sub-goal as well as the non-functional ones, such as the internal agent architecture;
- at the inter-agent level, one should identify how agents relate to each other, and how interaction between roles must occur for the global application goal to be correctly achieved. In particular, this amounts at defining the protocols to be used by agents to interact with each other as well as those to be used to access to the resources of the visited sites. Moreover, this implies defining the constraints under which the protocols can be executed and the computational activity that must be issued in organizational contexts to properly mediate and support the execution of protocols.

The potentials of a model based on local organizational contexts can be fully exploited in application development and execution only via the availability of a proper coordination infrastructure. This infrastructure has to be an implementation of an organizational context. It will be in charge to monitor all the interaction events performed by the agents and to react to them in order to let the coordination laws be respected. Moreover the infrastructure should be programmable to allow agents and site administrators to set up their own coordination laws.

Regarding this point, we implemented a programmable coordination space by means of a programmable tuple space. This system, called MARS, integrates with available agent infrastructures, like IBM's Aglets, providing a programmable tuple space service. Agents, running on those infrastructures, can connect to MARS to coordinate both with local resources and with other agents by exchanging tuples. Moreover, agents can implant coordination laws within the infrastructure. These coordination laws are triggered automatically by specified coordination activities (i.e. tuple insertions and extractions) and thus are enacted by the infrastructure itself.

4. Current Research Work

Our current research deals with two main topics. On the one hand we are working on the implementation of a more general and flexible architecture based on programmable coordination media. On the other hand we are studying how our model can be exploited in applications' development to achieve complex coordination tasks, and how the adoption of such model can impact on large-scale mobile agent applications.

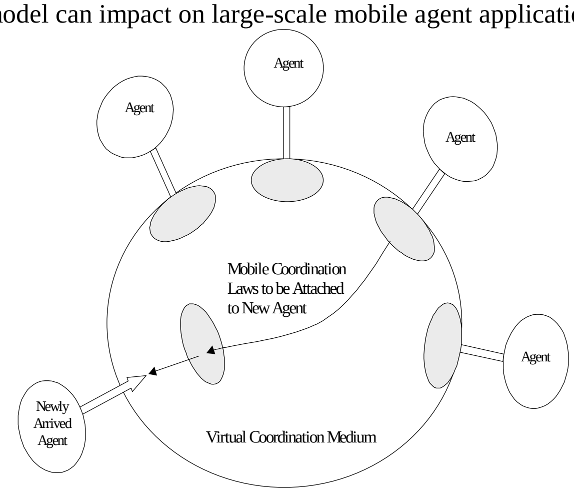

Fig. 2. Programmable Coordination Infrastructure for a MANET

Regarding the first topic, the definition of a more general programmable coordination architecture, we are currently challenging the case in which agents have to interact without the presence of any fixed infrastructure, i.e. in the context of Mobile Ad-Hoc Networks (MANET). The problem, in that case, is that the absence of the fixed infrastructure makes it hard to find a place where to actually allocate coordination media and, consequently, organizational laws.

Our goal would to maintain the same basic conceptual organizational framework, so not to influence the application developers' point of view. However, due to the lacking of a physical network infrastructure, the coordination media via which agents have to interact and in which coordination/organizational laws reside becomes only a *virtual medium*, implemented in a distributed way. The idea is to have a group of agents interacting in the context of a mobile ad-hoc network being somehow “attached” to a software substrate capable of influencing their coordination activities and, thus, of enacting specific coordination laws. When an agent arrives within a MANET, i.e., get connected to a new group and to the corresponding virtual coordination medium (see Figure 2), the code implementing the coordination laws and the software substrate needed to enact them is dynamically uploaded locally to the agent and attached to him to filter all its coordination activities.

Moreover, the type of interaction model that one may wish to adopt for the virtual interaction media may strongly influence the complexity and the efficiency of the implementation. A peer-to-peer interaction model it is likely to limit the state size of the virtual interaction medium, and this it is like to reduce the problem of maintaining the consistency of the state of the medium and of properly enacting the coordination laws. However, as MANET typically defines a very dynamic scenario, in which agent can connect and disconnect very frequently from a group, relying on direct peer-to-peer agent interactions may be problematic. Conversely, a data-oriented model such as a tuple-based one, by fully uncoupling interacting entities, alleviates the problem related to the dynamism of agents' connection and disconnection. However, it introduces the problem of maintaining the consistency of a possibly very large state size of the coordination medium. In between, the adoption of an event-based model, exploiting a global data status of limited size, may achieve a good trade-off between the two issues of uncoupling and state consistency.

Regarding the second topic, we are studying how the proposed model can be exploited in applications' development to achieve complex coordination tasks. In particular we are building simulations of large mobile applications, including hundreds of mobile agents, with the goal of analyzing the global and concurrent behaviors of these agents' ensembles, and of verifying how programmable coordination infrastructures can be used to control the global behaviors of these large applications.

The approach we are considering follows the perception that this kind of applications requires a deep exploitation of the notions of emergent behaviors and auto-adaptiveness. In particular we feel that the lessons of swarm-based or amorphous computing, where engineering of complex – and useful – behaviors is achieved via environment-mediated interactions of a multitude of very simple agents, can be of much use in this direction.

Moreover, we believe that lessons learnt in solving particular (simulated) applications could be possibly extended to more general software engineering principles. This is also very attractive because we expect that in the next few years advances in wireless and embedded technologies will change completely the information technology application scenarios. On the one hand, new, small, powerful and intelligent components will be distributed and embedded everywhere in the environment and in our everyday objects. On the other hand, thanks to the advances in wireless communication, it will become less expensive and easy to connect all these systems with each other and to the Internet, and to integrate inherently mobile entities like appliances, robots, people, etc.

We expect that the number of these components will somewhen reach an upper threshold over which radically new problems will arise, and conventional approaches and solutions will become inapplicable. In particular, we think it won't be possible to program and control these kinds of embedded applications in a conventional manner, but radically new software engineering principles will be required. Our feeling is that these principles will be based of the very same notions of emergent behaviors and auto-adaptiveness that we are investigating in our simulated application scenarios.

References:

The MOON Project Home Page: http://polaris.ing.unimo.it/MOON

The Authors: "Engineering Infrastructures for Mobile Organizations", Proceedings of the ESAW 2001 Workshop, Prague (CZ), July 2001, to appear in LNAI.