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
E-mail: {giacomo.cabri, letizia.leonardi, mamei.marco, franco.zambonelli}@unimo.it

Abstract

Mobile agents are a useful paradigm – other than a useful technology – for the development of complex Internet applications. However, the effective development of mobile agent applications requires suitable models and infrastructures. This paper proposes an organizational approach to the high-level design of mobile agent applications. The idea is to models the Internet as a multiplicity of local and active organizational contexts, intended as the places where coordination activities of application agents occur and are ruled. The paper discusses the advantages and the generality of such an approach, also with the help of a case study in the area of tourist assistance.

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. In most of the literature [5, 10], agent mobility refers only to the capability of an active object of dynamically transferring its execution onto different sites. Instead, 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 and opportunistic behavior in interactions [6]. Moreover, security or resource-control policies may constrain the behavior of an agent in accessing the resources of a site and in coordinating with other local agents. 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 [12].

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.

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, as shown via an application example. However, 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. For example, the latest implementation of the MARS coordination infrastructure [2], by implementing the concept of organizational contexts in terms of programmable tuple spaces, and by supporting mobility not only as a mechanism but also as a design abstraction, preserves during development the separation of concerns enforced during design, and promotes the writing of modular, easy to change and maintain, code.

2 Modeling Mobility

2.1 Virtual, Actual, and Physical Mobility

Mobility of agents may appear in different flavors in today's Internet computing environments:

- *Virtual agent mobility*. The wideness and the openness of the Internet scenario make it suitable to develop applications in terms of *network-aware* agents, i.e., agents that are aware of the distributed nature of the target and explicitly locate and access resources, services, and agents. From a different perspective, agents "navigate" the Internet and virtually move across its resources [3, 9].
- *Actual agent mobility*. This refer to the capability of agents (or, more generally, of compound active components) of moving across Internet sites while executing by dynamically and autonomously transferring their code, data, and state, toward the resources they need to access [5].
- *Physical agent mobility*. Mobile devices accessing the Internet – such as palmtops, cellular phones, etc. – can be modeled in terms of autonomous agents whose position, with respect the fixed network infrastructure (and thus the point from which to access it), changes accordingly to their geographical movements.

Virtual agent mobility is a useful abstraction to be exploited in the design of Internet-agent applications. Although it is possible to think at network-aware agents as autonomous components that have the possibility of globally interacting with a world-wide environment from a fixed perspective, minimization of conceptual complexity suggests conceiving network-aware agents as situated entities, that move across a world-wide network of independent sites, no matter if their execution is tied to a fixed Internet node [9].

Actual agent mobility – the one which is often considered as *the only* type of mobility that actually makes an agent a mobile agent – is a useful dimension of autonomy for the effective execution of application agents. In fact, despite the technological challenges (in terms of security and resource control) that agent mobility introduces, its wise exploitation can provide for saving network bandwidth, increasing both the reliability and the efficiency of the execution, handling mobility of users [7, 10]. Moreover, even if similar advantages and functionality can be exploited by weaker forms of low-level mobility (e.g., code mobility [5]), only agent mobility can naturally map at the execution level the virtual mobility already exploited at the design level.

Physical mobility is likely to become an increasing issue to be taken into account in the design of Internet applications, as more and more there will be the need of mobility-enabled services and applications. Many applications and services that are currently designed and developed to be exploited via non-mobile Internet-connected devices are likely to be exploited in the near future in a mobile setting. Devices will become mobile and will be in need of accessing the Internet from different – and dynamically changing – access points. Thus, all software agents that are currently non-mobile will become mobile agents simply because agents will execute in a mobile device. Moreover, even if neither agent-technology nor object-technology is exploited inside a mobile device, the observable behavior of a mobile-device with regard to the fixed infrastructure is actually those of an autonomous component, i.e., of a mobile agent, and will have to be modeled accordingly.

2.2 Local Interaction Contexts

To deal with all the above types of mobility in a uniform way, a suitable modeling of the scenario in which applications execute is required. In particular, it is suitable to model and develop mobile-agent-based applications by modeling agents' movements as movements across a multiplicity of different sites or environments, e.g., Internet nodes or administrative domains of nodes (see Figure 1). A site defines the *context* in which an agent executes and interacts. In other words, a site acts both as *execution context*, i.e., the place in which agents executes, whether virtually, actually, or physically, and as *interaction context*, i.e., the place in which agent coordination activities occur. Coordination activities may include accessing the local resources of a site and communicating and synchronizing with other locally executing agents, whether belonging to the same application or being foreign Internet agents.

What is the interaction model to be actually exploited for modeling interactions in a context is not of primary influence. Interactions may occur via message passing and ACLs [4], via meetings [10], or via shared dataspaces [1]. What really matters, from the software engineering perspective, is the locality model enforced by local interaction spaces, which reflects – at the level of application modeling – a notion of context intrinsic in mobility. In fact, for the very fact of moving across the Internet, agents access different data and services, and interact with different agents, depending on their position, i.e., of their current interaction context, independently of the model via which interactions occur.

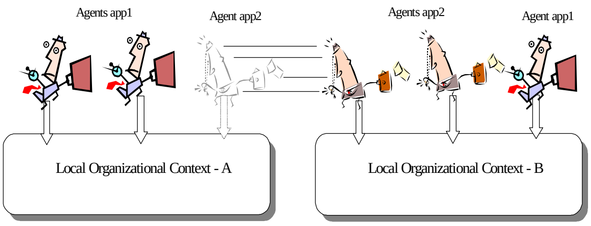

Figure 1. Mobility across interaction contexts

2.3 The Case Study: Tourist Assistance

As a case study, we focus on the problem of accessing the information and services related to the tourist resources of a city (or of a country), with the goal of properly organizing (also on the fly) a visit. We suppose that the city is provided with a suitable distributed infrastructure. In particular, let us suppose that the city is furbished with "information towers", distributed in the city, providing information and services related to the local region of the city in which they are situated (see Figure 2). As an example, should such an infrastructure be found in Rome (Italy), then an information tower by Città del Vaticano would provide information about visiting the Vatican museums and the Cappella Sistina; an information tower by Trastevere would provide information about the Santa Maria Church and services to reserve seats in a typical Trastevere restaurant.

We suppose that the access to information and to services on the information towers is provided via different means. First, information towers are Internet nodes too, and can be exploited by fixed agents/users to navigate (i.e., virtually move) across them. Second, information towers are enabled to host the execution of actually mobile agents, so that users can send their personal assistant mobile agents to roam across the information towers and there access information and services. Finally, information towers are enabled to provide information and services via wireless connections. So, tourists that are actually visiting a city by moving around in its streets can exploit a personal digital assistant (i.e., a physically mobile agent) to access local information towers and there discover what's around. For instance, a person walking in Trastevere can connect to the local information tower to discover if there is a Chinese restaurant in the neighborhood, and possibly reserve a seat.

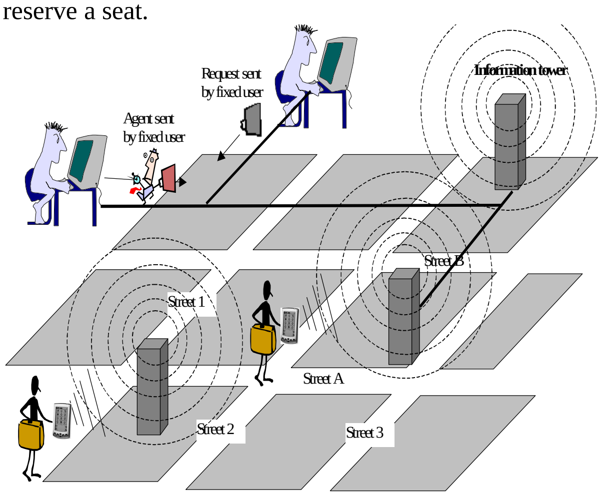

Figure 2. The scenario of the Case Study

The above scenario can be considered as representative of several other application areas, such as intelligent homes, traffic information systems, intelligent museums, all characterized by the presence of a fixed infrastructure based on a multiplicity of nodes, each associated to a specific location and providing location-sensitive information and services. For all these cases, the modeling of applications in terms of autonomous agents moving across local interaction contexts is undoubtedly the most suitable one. On the one hand the "information towers" providing information and services can be naturally modeled in terms of local interaction contexts in which to access information, services, and to interact with other agents. On the other hand, any software component accessing information and exploiting services can be modeled in terms of an autonomous agent that moves across different interaction contexts. The high-level design of application is not influenced by the specific type of mobility exhibited by agents, enabling to model all types of mobility in a uniform way. In particular, the high-level design of applications deals with mobility in terms of how and where agents interact in a given interaction context, disregarding the modeling of the execution context of agents, which is dependent on the specific type of mobility.

3 Designing Mobile Agent Organizations

3.1 Interaction Contexts and Organizations

Mobile agents, during their lives, will access to different resources and will interact with different agents, depending on their position. In other words, interactions intrinsically depend on the context in which they occur, even if the interaction contexts do not play any active role in it. In fact, an interaction context may act as a simple data and service repository, or as an event and message forwarder. In other words, an interaction context is simply a *place* rather than an *organization*. However, more sophisticated forms of context-dependency may be needed in Internet applications suggesting an active role of interaction contexts and leading to an *organizational perspective*:

- each context has its own specific characteristics and security policies, and it may be somehow in need of enacting specific coordination laws to rule the coordination activities of the locally executing agents (*environment-dependent coordination*).
- application agents belonging to a specific multi-agent application may require their coordination activities to occur according to specific application needs, e.g., agents may wish to enact specific coordination laws on their coordination activities, independently of the interaction context in which they actually take place and of the coordination laws there enacted (*application-dependent coordination*).

With regard to the former point, as all coordination activities of mobile agents on a site occur via the local interaction context, it is conceptually wrong to think at local interaction policies and coordination laws as something unrelated to interaction contexts. Instead, by assuming an organizational (or social) perspective, one can consider the interaction context in terms of an organizational (or social) context. In other words, one must consider that an agent, by entering via a movement an interaction context, enters a foreign organization (society) where specific organizational rules (or social conventions) are likely to be intrinsically enacted in the form of coordination laws. Thus, for the sake of conceptual simplicity, one must consider the interaction context as the locus in which the organizational laws ruling the activities of the local organization reside.

With regard to the latter point, one must take into account that mobile agents executing within a distributed multi-agent applications, still logically belong to the organization (society) defined by the multi-agent application in itself, despite the fact that they may are currently executing in a foreign organization. In other words, the context in which an agent executes and interacts is not only the one identified by the local interaction context, but is also the one of its own organization. As that, a local interaction context should not only be though as the place in which local organizational rules reside, but also as a place in which application agents may enact their own, application-specific, organizational rules in the form of coordination laws.

Putting all together, the introduced concepts lead to the scenario depicted in Figure 3, which represents a usable and modular conceptual framework for the analysis and design of mobile agent applications. Interaction contexts associated to different sites may enact specific organizational rules and, thus, may exhibit different behaviors in response to the same interaction events. Application-specific organizational rules can be enacted for all the agent of that application on any visited site, to act concurrently with the local organizational rules.

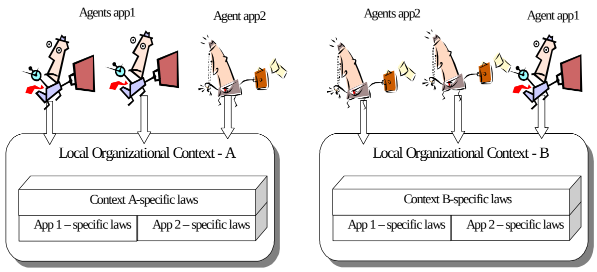

Figure 3. Local organizational contexts

3.2 Organizational Contexts and Application Design

The above framework defines useful abstractions to guide and simplify the process of developing mobile agent applications. In fact, the model naturally invites in designing an application by clearly separating the *intra-agent* issues – related to the specific computational roles of the application agents – and the *inter-agent* ones – related to the interaction of the agents with the other agents of the same application and with the visited execution environments. In other words, the framework promotes a clear separation of concerns, which is likely to reduce the complexity of application analysis and design.

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 [11, 12]. However, in our framework, the process of designing and deploying a mobile agent application must consider not only the role of application developers but also the one of system administrators.

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 [11]. 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 and the type of agent mobility (virtual, actual, or physical);

- 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 [12]. 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.

Independent of the duty of application designers is the duty assigned by our model to *site administrators*. When new kinds of application agents are known to be going deployed on the Internet (or when sites are going to federate to share a common infrastructure) the administrator of one site can identify all the local organizational rules that (s)he may find it necessary to enact for the execution of the application agents in the site. These rules can be used to facilitate the execution of the agents on the site, and/or to make the structure of the local organization homogeneous to agents' expectations (or to the requirements of the forming federation), and/or to protect the site from improper exploitation of the local contexts. As for the case of application developers, the identification of the local organizational rules are intended to define the local computational activity to be associated in local contexts to the interaction protocols performed by agents.

3.3 Design Issues in the Case Study

As an example related to the use of the organizational contexts abstraction in the case study, let us consider the problem of an agent-based application in charge of properly organizing a city tour for a tourist or for a group of tourists. Tourists can define personal preferences, time constraints, budget limitations, and the application should organize a tour of the city accordingly to the user requirements.

By adopting the introduced framework, designing this application amounts at designing the intra-agent level and the inter-agent, organizational, one.

At the intra-agent level, one has to decide whether and according to which criterion to subdivide the application goal into sub-goals, to be assigned to different agents. For instance, one could think at one agent in charge of finding and selecting a set of interesting sites to be visited ("visit agent"), while another agent is in charge of looking for and reserving seats in restaurants ("restaurant agent"), accordingly to both users' preferences and the decision of the visit agent. On the basis of the goals to be achieved by the identified agents, one can then decide what internal architecture for agents is better suited.

At the inter-agent level, one has to analyze in which coordination activities the application agents should be involved and which organizational rules should be enacted on coordination activities. For instance, coordination activities may include:

- retrieving from information towers the information needed to successfully organize a visit, e.g., visiting time and admittance prices for museums, bus schedules, restaurant information, and so on;
- coordinating with other agents either of the same application, e.g., the visit agent will have to coordinate its decisions with the restaurant agent;
- coordinating with foreign agents, e.g., the restaurant agent will have to coordinate itself with some agents of a restaurant to make a reservation.

Once the coordination activities of the application have been identified, one has to identify the organizational rules that should be enacted on the accessed organizational contexts. Let us focus on the first two points.

With reference to the first point, let us suppose that information towers provide access to restaurant information in terms of "type of restaurant" only (e.g., Italian, Indian, Chinese, etc.) while the agent would like to have access to this information in terms of "class of restaurant" (e.g., cheap, expensive, high class). In that case, the trivial and ineffective solution is to force the agent retrieve information about all restaurants of all types, and then select all the information retrieved on the basis of prices criterion. The alternative and effective solution would be for the agent to "adapt" the behavior of the accessed organizational context to its own need. In particular, this implies integrating an application-specific computational activity in organizational context, devoted to accept requests for restaurant information in terms of restaurant classes, and handle them accordingly.

With reference to the second point, identifying that the visit agent and the restaurant agent will have to coordinate together, leave open the question about how this coordination process should be actually ruled. One can decide that the visit agent is more important, and thus a restaurant can be reserved only once the schedule of the visit has been decided. In other words, one defines the organizational rules avoiding the restaurant agent to proactively initiate a coordination protocol involving restaurant reservation before the visit agent has fixed a schedule. Viceversa, one can opt for giving food more importance and organizes a visit on the basis of the selected restaurants, and constrains the execution of the protocols accordingly. Whatever the choice, it is worth that the clear separation of concerns enables changing the design of organizational rules without changing the agents' internal design, and viceversa.

Separated from the above design process, the perspective of the local administrator of an information tower is to enact organizational rules on the information towers, to control accesses and the coordination activities of application agents. As an example, the administrator can impose a rule that avoids a "restaurant agent" with a given identity to initiate a protocol for the reservation of a restaurant if it has already reserved a place in another restaurant during the same time span.

As a final note, the above analysis does not have mentioned any problems related to the specific type of mobility of application agents, which should have been addressed in the intra-agent design. Actually, this has been a deliberated choice. In fact, once again, all the above considerations abstract away from any low-level mobility detail. The discussed application can be based on virtually mobile agents, on actually mobile agents, or even on physically mobile agents, which plan the visit of a tourist on the fly, while the tourist is already walking in the city.

4 Conclusions and future work

In this paper, we have introduced an organizational approach for the design of Internet applications based on mobile agents and have shown how it can promote a modular approach to application design. A programmable coordination infrastructure, by mapping at the infrastructure level the abstractions of the framework, can lead to the development of easy to program and easy to maintain mobile agent applications.

In the future, we intend to develop a formal model of organizational contexts, possibly by exploiting already defined models for mobility and mobile agents [3, 8]. In addition, we intend to better investigate a possible integration of our model within standard middleware technologies like CORBA and Jini.

Acknowledgements

Work supported by Italian Ministry for University and Science and Technology Research (MURST) within the project "MUSIQUE: Infrastructure for QoS in Web Multimedia Services with Heterogeneous Access" and by the University of Modena and Reggio Emilia with a fund for young researchers.

References

[1] G. Cabri, L. Leonardi, F. Zambonelli, "Mobile-Agent Coordination Models for Internet Applications", IEEE Computer, 33(2):82-89, Feb. 2000.

[2] G. Cabri, L. Leonardi, F. Zambonelli, "MARS: a Programmable Coordination Architecture for Mobile Agents", *IEEE Internet Computing*, 4(4):26-35, July-Aug. 2000.

[3] L. Cardelli, A. D. Gordon, "Mobile Ambients", *Theoretical Computer Science*, 240(1), July 2000.

[4] T. Finin at al., "KQML as an Agent Communication Language", *3rd International Conference on Information Knowledge and Management*, November 1994.

[5] A. Fuggetta, G. Picco, G. Vigna, "Understanding Code Mobility", *IEEE Transactions on Software Engineering*, 24(5):352-361, May 1998.

[6] N. R. Jennings, "On Agent-Based Software Engineering", *Artificial Intelligence*, 117(2):277-296, 2000.

[7] D. B. Lange, M. Oshima, *Programming and Deploying Java™ Mobile Agents with Aglets™*, Addison-Wesley, August 1998.

[8] G. P. Picco, A. M. Murphy, G.-C. Roman, "Software Engineering for Mobility: A Roadmap", in *The Future of Software Engineering*, A. Finkelstein (Ed.), ACM Press, pp. 241-258, 2000.

[9] J. Waldo, G. Wyant, A. Wollrath, S. Kendall, "A Note on Distributed Computing", *Mobile Object Systems*, Lecture Notes in Computer Science, No. 1222, pp. 49-64, Springer-Verlag, Berlin (D), 1997.

[10] J. White, "Mobile Agents", in *Software Agents*, J. Bradshaw (Ed.), AAAI Press, pp. 437-472, 1997.

[11] M. Wooldridge, N. R. Jennings, D. Kinny, "The Gaia Methodology for Agent-Oriented Analysis and Design", *Journal of Autonomous Agents and Multi-Agent Systems*, 3(3):285-312, 2000.

[12] F. Zambonelli, N. R. Jennings, A. Omicini, M. J. Wooldridge, "Agent-Oriented Software Engineering for Internet Applications", in *Coordination of Internet Agents*, A. Omicini et al. (Eds.), Springer-Verlag, 2001.