Agents Jumping in the Air: Dream or Reality?*

Oscar Urra, Sergio Ilarri, and Eduardo Mena

Department of Computer Science and Systems Engineering, University of Zaragoza,
María de Luna 1, 50018, Zaragoza, Spain
ourra@ita.es,{silarri,emena}@unizar.es

Abstract. Mobile agent technology has traditionally been recognized as a very useful approach to build applications for mobile computing and wireless environments. However, only a few studies report practical experiences with mobile agents in a wireless medium. This leads us to the following question: is mobile agent technology ready to be used in this environment?

In this paper, we study existing mobile agent platforms by analyzing if they could be used effectively in a wireless medium. We identify some key missing features in the platforms and highlight the requirements and challenges that lie ahead. With this work, we expose existing problems and hope to motivate further research in the area.

1 Introduction

Mobile agents [6] are programs that execute in contexts called *places*, hosted on computers, and can autonomously travel from *place* to *place* resuming their execution there. Thus, they are not bound to the computer where they were created; instead, they can move freely between computers.

Mobile agents provide interesting features, thanks to their autonomy, adaptability, and capability to move to remote computers. They can carry the computation wherever it is necessary, without the need of installing specialized servers there (only a generic *mobile agent platform* [12] is needed). In particular, the interest of mobile agent technology for wireless environments has been emphasized in the literature (e.g., see [10]). Thus, for example, instead of communicating a large amount of data from a computer to a mobile device, a mobile agent can move to that computer to process the data locally, filtering the non-relevant data that should not be communicated through the network. As another example, a mobile device could use a mobile agent to perform a processing-intensive task on a fixed computer with the required resources, relieving the overload of the mobile device. This will increase, in turn, its battery life, which is an important limitation on these devices.

However, and despite mobile agent technology has been evaluated in many research works in the context of distributed systems, there are only a few reported experiences on the use of mobile agents in real wireless environments with

* This work was supported by the CICYT project TIN2007-68091-C02-02.

J. Cabestany et al. (Eds.): IWANN 2009, Part I, LNCS 5517, pp. 627–634, 2009.

mobile devices. In most cases, a simple and static wireless environment is considered (e.g., we performed an experimental evaluation in this context in [13]). Moreover, we also believe that it is important to study the challenges that mobile environments imply for a mobile agent platform. For example, in ad hoc networks multi-hop protocols may be required to reach a certain node (as each node can only communicate with other nodes within its communication range) As another example, it becomes apparent that mobile agent platforms should provide services for agents to discover other nodes.

As far as we know, no other work has studied in depth the requirements and current limitations of mobile agent platforms to be usable in a mobile environment[1], which is the goal of this paper. We analyze the challenges that need to be solved and highlight the advantages and disadvantages of existing platforms in a mobile environment. The structure of the rest of this paper is as follows. In Section 2, we introduce the concept of mobile agent platform and motivate the development of this work. In Section 3, we study security issues. In Section 4, we consider the challenges of using wireless communications and mobile ad hoc networks. In Section 5, we focus on architectural elements that must be considered. Finally, in Section 6, we summarize our conclusions.

2 Mobile Agent Platforms in Wireless Environments

There exist several *mobile agent platforms*, which allow the execution of mobile agents and provide them different services (e.g., communication services, transportation services, security, etc.). These platforms differ in different aspects (such as their general architecture, communication style, etc.) and compare differently in terms of performance, reliability or scalability [12].

Mobile agent technology has been proposed as a key element in the development of many applications, for a variety of reasons (e.g., their capability to exploit the locality of data by moving to the data source instead of interacting remotely using a network). With the increasing popularity of mobile devices, it was a natural step forward to try to apply mobile agents to the mobile environment. Given its portable nature, mobile devices use wireless communications, creating a scenario completely different from a traditional distributed environment with fixed networks. Such an environment has a number of advantages (e.g., the processing is not tied to a fixed location) but also some drawbacks, such as the limited computational power of mobile devices and a short communication range based on wireless technologies –that usually offer a low bandwidth, a high latency, and intermittent/unreliable connectivity–. Thanks to their features, mobile agents can be very useful in wireless applications (e.g., see [10]), as they could help to reduce the negative effects of such limitations.

However, despite the increasing popularity of wireless services and the advantages of mobile agents in these environments, there are not many practical

[1] We are only aware of the work presented in [1], which focuses only on security issues over a P2P communication model for mobile agents in ubiquitous computing and does not analyze the existing platforms.

applications of this technology, except for some proofs of concept. A possible explanation could be that it is very difficult to develop and maintain such mobile agent based applications because existing platforms lack a number of features that should be present for their use in a wireless and mobile environment. Among them, we could emphasize:

- Features related to security, since the use of wireless communications broadcast data that could be intercepted or altered without having any notice.
- Features related to special network topologies, since there are multiple mobile nodes with short range and unstable communications, which can make the process of transfer data between two nodes challenging.
- Features related to the way the platform itself works, since the mobile agents need some services to perform their tasks efficiently, such as the transportation to other places or the communication with other agents in the network.

Not providing the features listed above is an important difficulty to develop applications based on mobile agents for mobile environments. Thus, for example, an agent should be able to detect the availability of new nearby devices (e.g., to travel to them) and to communicate with other agents easily and efficiently. Mobile agent platforms have usually been developed with a static context in mind and now they must be adapted to a more open and dynamic environment.

Besides, scalability and reliability are two key features that should also be provided. They are important even when the mobile agents rely on fixed networks, and therefore even more critical in a challenging environment with wireless networks. Some tests indicate that the scalability/reliability of some platforms should be improved [3]. However, as this need is not tied to the wireless case –although heightened by it– we will focus on the other features in the rest of this paper. We will consider the following platforms: SPRINGS (`http://osiris.cps.unizar.es/SPRINGS/`), Voyager Community Edition (`http://www.recursionsw.com/`), and JADE with LEAP (`http://jade.tilab.com/`). SPRINGS provides a high scalability in distributed environments [3]. Along with SPRINGS, Voyager is one of the most reliable platforms according to experiments performed in [12]. Finally, JADE is currently the most popular agent platform and its extension LEAP supports agents on mobile devices. Therefore, we believe that these platforms will be representative of the state of the art.

3 Security Issues

Security is always a major concern. Besides security problems with mobile agents in fixed networks [2,14], other problems particular to wireless environments arise.

3.1 Communication Encryption

Wireless communications usually broadcast the transmitted data in an omnidirectional way. The disadvantage is that these data will be received not only by

the intended destination but also by anyone within the range of the originating communication device. Although almost every wireless communication protocol (such as Wi-Fi or Bluetooth) can encrypt the data, it is not mandatory and in some circumstances an unencrypted communication can be the only form available (e.g., with public access points or *hotspots*). If a mobile agent is transferred using an insecure channel, its code and data will be exposed to any nearby device. To avoid this problem, the mobile agent platform should be able to encrypt all its communications when connecting to others.

Regarding the considered existing mobile agent platforms, both Voyager and JADE-LEAP can natively use SSL connections –which assure data flow encryption–, whereas SPRINGS lacks this feature.

3.2 Code Integrity

The code of a mobile agent can be altered, either intentionally or accidentally, in many ways. For example, in the first case, a malicious user could modify it while it is running on its device. In the second case, a failure such as a memory loss or a problem with the wireless transmission of the agent could lead to a corruption of its code. The execution of an agent whose code has been altered can lead to unpredictable consequences and should be avoided. Thus, the mobile agent platform should verify the agent's code integrity before starting its execution, for example signing the code with a X.509 digital certificate.

Thanks to the use of SSL by Voyager and JADE-LEAP, any attempt to tamper with an agent while it is being transmitted will be detected. Moreover, Voyager provides a signing mechanism to assure that the agent is not modified while it is stored in the device's memory. SPRINGS lacks both features.

3.3 Authentication and Trust

A mobile device has a number of resources (such as the CPU, the memory, the file system, the user interface, etc.) susceptible to be used by an agent to accomplish its task. To avoid abuses on the use of these resources, the platform should state the extent to which it trusts an incoming agent –this is critical in an open environment–. Depending on it, the platform should allow, limit or even deny the mobile agent the access to the resources. There already exist mechanisms of authentication for distributed environments (e.g., Kerberos) that could be used to determine the owner of an agent. Once it is authenticated, different access control policies can determine the resources that the agent can use.

Voyager and JADE-LEAP have mechanisms to verify the identity of agents and other platform components, and grant different privilege levels. However, SPRINGS has no authentication mechanisms and just a basic access control.

4 Network Issues

Existing mobile agent platforms assume the existence of a TCP/IP network where every node can potentially connect to any other. This approach has the

advantage of its simplicity, but prevents considering explicitly other forms of communication that could be beneficial in a context with wireless mobile agents.

One limitation of mobile devices is that their communication interfaces (usually Wi-Fi or Bluetooth) have a relatively short range (a few hundred meters). Therefore, in a mobile ad hoc network a multi-hop routing protocol may be needed for two nodes to communicate. For a mobile agent platform, it would be interesting to access information about: whether the connection is fixed or wireless, the bandwidth available, if an established link (e.g., a Wi-Fi connection) is encrypted or not, the coverage level or the strength of the received signal in the case of a wireless communication, the identifiers of nearby wireless networks or nodes that could be contacted, etc.

With all this information, the platform would be able to take different useful decisions. For example, it could select the most appropriate communication link if there are several options available, in the case of a secure channel it would avoid encrypting the communicated data –reducing the CPU utilization and therefore saving battery power in the mobile device–, it could decide not to retry a failed communication if the coverage is poor, etc. Making this information accessible to the agents would also be interesting. For example, an agent could decide to jump to another device/node if its current device is getting out of coverage.

As far as we know, current mobile agent platforms do not have the ability to obtain this kind of information, which would be important in order to make the platform truly adaptive to different network environments. The closest related features that the considered platforms have are the following: Voyager can measure the network latency and detect that a communication is broken when an abnormally high value is obtained; JADE-LEAP can get basic information about the link status (connected or disconnected); finally, SPRINGS retries failed communications automatically according to some predefined policy.

5 Architecture Issues

In this section, we provide an overview of some architectural issues that should be considered in a mobile agent platform suitable for a mobile environment.

5.1 Automatic Discovery Service

As opposed to a fixed distributed infrastructure for mobile agents, a mobile context usually presents an open environment where many different computers/devices may appear and disappear at any time. Therefore, it is of paramount importance to provide agents with appropriate mechanisms to locate nodes to where they can travel. Moreover, the capabilities/services offered by these nodes should be advertised to allow the agents to decide a convenient target node.

Discovery of Services. There exist several service discovery protocols, such as the *Service Location Protocol* (*SLP*), the *Universal Plug and Play* (*UPnP*),

or the use of the *Jini* technology. The suitability of these protocols for ambient intelligence is analyzed in [9].

Mobile agents should be provided an automatic discovery mechanism to allow them the detection of services of interest[2]. However, as far as we know, these protocols have not been integrated in any existing mobile agent platform. Moreover, it is not clear if they are the best choice in a mobile environment. For example, JADE-LEAP has a federated Yellow Pages (YP) service, but allocating it to a node is not transparent to the programmer and movements of the mobile device may invalidate the convenience of using that allocated YP service.

Finally, we would like to highlight the importance of providing a *semantic matching* of services [11], not only *syntactic matching*, if we want to enable dynamic and flexible interactions among the mobile agents. Thus, *Agent Communication Languages* (*ACLs*) [4] could play an important role. The language proposed by the *Foundation for Intelligent Physical Agents* (the *FIPA* ACL –see `http://www.fipa.org/`–) is the most popular proposal and it is supported by several mobile agent platforms (e.g., JADE). If services for mobile agents are implemented as web services, as suggested in [15], then existing techniques for web services (UDDI, WSDL, SOAP, OWL-S, etc.) could also be adopted.

Discovery of Nodes. It is necessary for an agent to be able to detect the nodes that are reachable and which services/features provide them. Thus, nodes must be auto-descriptive. This is particularly important in a heterogeneous environment where there will be computers/devices with different processing and communication capabilities.

However, no platform provides mechanisms to allow an agent to detect other potential target computers/devices. Although some platforms provide name services to query the computers that can host an agent (e.g., the Region Name Server in SPRINGS [3]), they do not consider that some computers/devices may not be accessible from a given location (e.g., if the mobile agent is executing on a device that can only communicate with other devices within communication range); indeed, the name service itself may be unreachable.

5.2 Tracking of Mobile Agents and Name Services

As mobile agents move from one computer/device to another, tracking their locations efficiently (which is required if we want to be able to allow communications with those agents from any computer/device) is challenging. To solve this problem, different approaches have been considered. Some mobile agent platforms provide a naming service that can be used to locate an agent and then send a message to its address. An alternative approach supports communications by using *proxies* (similar to the *stubs* in *RMI*) as a convenient abstraction to refer to remote agents (e.g., to send them messages or call methods remotely). Several platforms, such as SPRINGS and Voyager, support proxies. Related to the

[2] Some works propose using mobile agents to implement service discovery [5,8].

Table 1. Comparison of mobile agent platforms: features for mobile environments

Feature		Voyager	JADE/LEAP	SPRINGS
Security Issues	Encryption	Yes (SSL)	Yes (SSL)	No
	Code integrity for transmissions	Yes (SSL)	Yes (SSL)	No
	Code integrity for execution	Some (signing)	No	No
	Authentication and trust	Yes (very rich)	Yes (rich)	Very basic
Network Issues	Access to link layer	No	No	No
	Adaptability	Latency measure	Conn. status	Conn. retrying
Architectural Issues	Discovery of nodes	No	No	No
	Transparent service discovery	No (YP)	No (DF)	No
	Adapted tracking approach	No (AgentSpaces)	No (AMS)	No (RNS)
Strong points		Messaging, devices	Messaging, ontologies	Scalable, reliable

concept of proxies, *dynamic proxies* remain valid independently of the agents' migrations (i.e., the reference is updated automatically) [3].

However, existing strategies have been developed with a fixed network in mind and are not appropriate in a mobile environment. For example, SPRINGS assumes the existence of stable *Region Name Servers* (*RNSs*) and *location servers* with tracking responsibilities. This could be unsuitable in a dynamic context because nodes can leave/enter the network at any time and some nodes could be temporarily unreachable. Similarly, according to [7], Voyager uses forwarding chains of proxies, which may be inconvenient because any link (pointer) in the chain could disappear at any time. Thus, we believe that new tracking techniques are needed in this context, avoiding mechanisms that rely on the availability of certain nodes or centralized approaches, in favor of adaptive tracking approaches.

Finally, we should mention that ensuring the uniqueness of agent names and at the same time providing user-friendly mechanisms to address the agents is a challenge in an open and dynamic environment. A potential solution would imply a shift in the way agents communicate. At present, it is usually assumed that agents identify their partners by name. An alternative would be to identify partners by service, which would make user-friendly agent names unnecessary.

6 Conclusions

Mobile agent technology has been highlighted as a very interesting approach to build applications for mobile environments. However, despite many theoretical studies proving the usefulness of this alternative, it is hard to find practical applications with real prototypes and using the available mobile agent platforms. One reason is probably that such platforms have been developed with a fixed distributed environment in mind, but not considering the features that may be of special interest in a mobile environment (e.g., reliance against security threats, adaptation to the network technology, and service/node discovery). Considering these features would make the adoption of the technology much easier.

In this paper, we have identified the requirements and desired features of mobile agent platforms to be used in a mobile environment. In light of these requirements, we have analyzed the missing features in some popular mobile agent platforms (see Table 1 for a summary). Thus, we cover a gap in the literature,

where many works propose the use of mobile agents in wireless environments but without analyzing the applicability of current mobile agent platforms.

As future work, we plan to study these issues in detail and propose solutions for SPRINGS [3]. We hope that this work will encourage research and development efforts that will eventually lead to a big leap forward into the wireless.

References

1. Bagci, F., Schick, H., Petzold, J., Trumler, W., Ungerer, T.: Communication and security extensions for a ubiquitous mobile agent system (UbiMAS). In: 2nd Conf. on Computing Frontiers (CF 2005), pp. 246–251. ACM Press, New York (2005)
2. Greenberg, M., Byington, J., Harper, D.: Mobile agents and security. IEEE Communications Magazine 36(7), 76–85 (1998)
3. Ilarri, S., Trillo, R., Mena, E.: SPRINGS: A scalable platform for highly mobile agents in distributed computing environments. In: 4th Intl. WoWMoM 2006 Workshop on Mobile Distributed Computing (MDC 2006), pp. 633–637. IEEE, Los Alamitos (2006)
4. Kone, M.T., Shimazu, A., Nakajima, T.: The state of the art in agent communication languages. Knowledge and Information Systems 2(3), 259–284 (2000)
5. Meier, R.T., Dunkel, J., Kakuda, Y., Ohta, T.: Mobile agents for service discovery in ad hoc networks. In: 22nd Intl. Conf. on Advanced Information Networking and Applications (AINA 2008), pp. 114–121. IEEE, Los Alamitos (2008)
6. Milojicic, D., Douglis, F., Wheeler, R.: Mobility: processes, computers, and agents. ACM Press, New York (1999)
7. Moreau, L.: A fault-tolerant directory service for mobile agents based on forwarding pointers. In: Symp. on Applied Computing (SAC 2002), pp. 93–100. ACM, New York (2002)
8. Nehra, N., Patel, R.B., Bhat, V.K.: MASD: Mobile agent based service discovery in ad hoc networks. In: Aluru, S., Parashar, M., Badrinath, R., Prasanna, V.K. (eds.) HiPC 2007. LNCS, vol. 4873, pp. 612–624. Springer, Heidelberg (2007)
9. Preuveneers, D., Berbers, Y.: Suitability of existing service discovery protocols for mobile users in an ambient intelligence environent. In: Intl. Conf. on Pervasive Computing and Communications, pp. 760–764. CSREA Press (2004)
10. Spyrou, C., Samaras, G., Pitoura, E., Evripidou, P.: Mobile agents for wireless computing: the convergence of wireless computational models with mobile-agent technologies. Mobile Networks and Applications 9(5), 517–528 (2004)
11. Sycara, K.P., Widoff, S., Klusch, M., Lu, J.: Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. Autonomous Agents and Multi-Agent Systems 5(2), 173–203 (2002)
12. Trillo, R., Ilarri, S., Mena, E.: Comparison and performance evaluation of mobile agent platforms. In: 3rd Intl. Conf. on Autonomic and Autonomous Systems (ICAS 2007), p. 41. IEEE, Los Alamitos (2007)
13. Urra, O., Ilarri, S., Mena, E.: Testing mobile agent platforms over the air. In: 1st ICDE Workshop on Data and Services Management in Mobile Environments (DS2ME 2008), pp. 152–159. IEEE, Los Alamitos (2008)
14. Vigna, G. (ed.): Mobile Agents and Security. LNCS, vol. 1419. Springer, Heidelberg (1998)
15. Zhang, J., Wang, Y., Varadharajan, V.: A new security scheme for integration of mobile agents and web services. In: 2nd Intl. Conf. on Internet and Web Applications and Services (ICIW 2007), p. 43. IEEE, Los Alamitos (2007)