Mobile Agents: Ten Reasons For Failure

Giovanni Vigna
Reliable Software Group
Department of Computer Science
University of California, Santa Barbara
vigna@cs.ucsb.edu

Abstract

Mobile agents have often been advocated as the solution to the problem of designing and implementing distributed applications in a dynamic environment. Mobile agents provide a very appealing, intuitive, and apparently simple abstraction. Unfortunately there are many difficult problems that have to be addressed in order to make mobile agent-based applications work reliably. This paper discusses some of the problems associated with mobile agents and argues that other forms of code mobility can provide similar advantages while raising much fewer issues.

1. Introduction

Code mobility can be defined informally as the capability to dynamically change the bindings between code fragments and the location in which they are executed. The ability to relocate code is a powerful concept that started a very interesting range of developments. Different forms of code mobility have been identified [3]. The most common form of code mobility is *code on demand*, which is the download of executable content in a client environment as the result of a client request to a server. A well-known example of this approach is the download of Java applets or JavaScript code in a WWW browser. A different form of code mobility is represented by the upload of code to a server. The uploaded code is executed by the server and possibly the results of the computation are sent back to the client. This form of mobility, also known as *remote evaluation* [6], allows the client to execute a computation close to the resources located at the server's side so that network interaction can be reduced. Common examples are represented by the use of SQL to perform queries to a remote database or the upload of PostScript code to a remote printer. A third form of code mobility is represented by the *mobile agent* paradigm. In this case, mobile components can explicitly relocate themselves across the network, usually preserving their execution state (or part thereof) across migrations. Examples of systems supporting this type of mobility are Telescript [7] and D'Agents [4].

2. Ten Reasons for Failure

Mobile agents are the most powerful form of code mobility. Unfortunately, while other forms of code mobility are in wide-spread use, mobile agents have not been well received by the Internet community. The following is a discussion of the reasons why mobile agents didn't succeed. This discussion is largely based on the author's personal experience with mobile agent design, development, and testing.

Mobile agents do not perform well. Mobile agents have often been advocated as a way to optimize resource access and reduce network overhead. Unfortunately, these statements are very seldom supported by a quantitative, precise evaluation of the overhead introduced by moving both the code and the state of an agent. Even simplified, qualitative evaluations have shown that mobile agents, in the general case, provide worse performance than other mechanisms, such as remote evaluation and remote procedure invocation [1]. The reality is that, in the general case, mobile agents are very expensive.

Mobile agents are difficult to design. Mobile agents seem to provide a very natural and intuitive concept: component autonomy. Unfortunately, most distributed applications have a degree of complexity that is more easily addressed using well-known software architecture concepts, such as modularization and modeling of inter-component interactions. In a mobile agent-based design, it is difficult to clearly identify which components will be interacting and how this interaction can be modeled.

Mobile agents are difficult to develop. The development of a mobile agent-based application is a daunting task. The code has to be implemented so that it runs in unpredictable environments, possibly interfacing with other mobile agents and static components never seen before. Being able to foresee the type of operating environment the mobile agent will be running in is very difficult. In addition, the technologies that should support the development process are often just prototypes that do not provide the type of support needed to develop real-world applications.

Mobile agents are difficult to test and debug. Testing and debugging software is an art in itself. There are many

pitfalls in testing and debugging simple, single-threaded, non-distributed software. Distribution and mobility add complexity to this process. Mobile agents may move from one node to another in a non pre-defined order. Understanding and testing an agent's implementation throughout such a complex execution history is incredibly hard.

Mobile agents are difficult to authenticate and control. Mobile agents need to be authenticated when entering an environment that enforces access control mechanisms based on some identity concept. The problem is that there are many identities associated with a mobile agent. For example, an agent may be associated with the agent developer, the agent's code signer, the agent dispatcher, and the host the agent visited last. It is not clear which identities should be authenticated and how the access control mechanisms should take into account this information. Even if a suitable general model is devised, the complexity of such a model would make the access control configuration process extremely error-prone.

Mobile agents can be brainwashed. Mobile agents that travel across multiple hosts to complete their tasks are vulnerable to a number of attacks coming from malicious executing environments. For example, a malicious host can modify the code or memory image of an agent to change the way the agent behaves. The result of this "brainwashing" attack would be the creation of a malicious agent whose actions will be attributed to one of the identities initially associated with the agent. This type of attack is extremely difficult (if not impossible) to detect and prevent.

Mobile agents cannot keep secrets. Mobile agents have been advocated as a means to implement e-commerce and other critical applications. To perform sensitive transactions (e.g., signing a contract), it is often necessary to perform operations that require secret material, such as a private key. Unfortunately, a secret cannot be effectively concealed if it has to be used by an agent on a remote host and the agent cannot interact with the "home base". Even though some solutions based on the evaluation of encrypted functions have been proposed [5], no practical applications of this mechanism have been devised.

Mobile agents lack a ubiquitous infrastructure. Mobile agents require an infrastructure that supports the tasks of marshaling, transfer, and unmarshaling an agent's representation. This infrastructure needs to be deployed on every host that can possibly be the recipient of an agent. This is a requirement that is difficult to meet especially because existing infrastructures have been proven to be vulnerable to a number of attacks [2].

Mobile agents lack a shared language/ontology. Mobile agents need to interact with the environment they visit in order to achieve their goals. This interaction requires that the format used to exchange data and the meaning that is associated with the data is understood and agreed upon by both the agent and the partner of the interaction. Even though a number of these shared languages/ontologies have been proposed, there is still no widely accepted language or ontology.

Mobile agents are suspiciously similar to worms. Mobile agents are components that autonomously trigger the transfer of their image to a remote host where they restart execution. This mechanism has some striking similarity to the way malicious worms spread across networks. Worms have proven to be extremely difficult to eradicate and some worms are nowadays considered part of the "background noise" of the Internet. A mobile agent infrastructure would support the execution of both benign and malicious agents and, therefore, it would be prone to be leveraged to launch worm-like attacks.

3. Conclusions

Mobile agents are an intriguing concept that appeared to be a viable solution to the problem of distributing computations in a very dynamic, networked environment. Unfortunately, the apparently simple concept of having a component "jump" from host to host requires a substantial amount of infrastructure and poses hard (maybe unsolvable) challenges from both the security and the performance standpoints. Because of these limitations, mobile agents didn't meet the expectations they raised years ago in terms of widespread deployment and use.

We advocate that other (simpler) forms of mobility, like remote evaluation or code on demand, can provide support for the users' requirements in terms of service customizability, optimized access to distributed resources, and deployment of applications in a mobile networking environment, without raising many of the issues that have to be addressed when using mobile agents.

References

[1] A. Carzaniga, G. Picco, and G. Vigna. Designing Distributed Applications with Mobile Code Paradigms. In *Proceedings of the 19^{th} International Conference on Software Engineering (ICSE '97)*, Boston, MA, April 1997.

[2] S. Fischmeister, G. Vigna, and R. Kemmerer. Evaluating the Security Of Three Java-Based Mobile Agent Systems. In *Proceedings of the 5^{th} International Conference on Mobile Agents (MA '01)*, Atlanta, GA, December 2001.

[3] A. Fuggetta, G. Picco, and G. Vigna. Understanding Code Mobility. *IEEE Transactions on Software Engineering*, 24(5):342–361, May 1998.

[4] R. Gray, D. Kotz, G. Cybenko, and D. Rus. D'Agents: Security in a multiple-language, mobile-agent system. In *Mobile Agents and Security*, volume 1419 of *LNCS*, pages 154–187. Springer, 1998.

[5] T. Sander and C. Tschudin. Towards Mobile Cryptography. In *Proceedings of the IEEE Symposium on Security and Privacy*, Oakland, CA, 1998.

[6] J. Stamos and D. Gifford. Implementing Remote Evaluation. *IEEE Transactions on Software Engineering*, 16(7):710–722, July 1990.

[7] J. White. Telescript Technology: Mobile Agents. In J. Bradshaw, editor, *Software Agents*. AAAI Press/MIT Press, 1996.