

Obstacles to the adoption of mobile agents

Volker Roth
Fraunhofer IGD, Germany
vroth@igd.fhg.de

Abstract

The idea of mobile software agents has inspired many researchers ever since the term was coined probably by Jim White of General Magic around 1994, although the principal idea is much older. Shoch and Hupp at Xerox PARC published their idea of the Worms programs already in 1982, and they were in turn inspired by John Brunner's 1975 science fiction novel "The Shockwave Rider." However, almost thirty years after their inception, the only widespread incarnation of mobile software agents is malware. In this brief invited essay I take a personal and biased stab at commenting this situation.

Keywords: *Mobile agents.*

1. Introduction

The elegance and flexibility of the mobile agent approach goes largely undisputed while even in the prime of the mobile agent activities researchers had difficulties to come up with a compelling reason why mobile agents are necessary. Holger Peine gives a critical and highly recommendable account of the arguments that are generally brought forward in favor of the mobile agent approach – and of the misconceptions that go along with them [6]. A somewhat perplexing justification of mobile agents has been that, while each individual advantage of mobile agents can be realized equally or better by traditional approaches, their adoption would facilitate a very large number of network services and applications [3]. Yet, reality shows that the incentives offered by mobile agents have not been sufficient to stimulate their widespread deployment.

Despite this bleak prospect, a constant trickle of mobile agent related submissions, workshops, and panel discussion about their past and future demonstrates that the topic has not yet been abandoned by research, and a variety of reasons have been offered why mobile agents have not had their breakthrough yet. In this invited essay I would like to briefly discuss two such reasons: the lack of applications

versus the lack of a sufficient installation base, and security considerations.

2. Chickens and Eggs

Among other reasons, mobile agents are said to suffer from a very typical *chicken-and-egg* situation. The argument goes as follows: mobile agent applications require a large infrastructure of servers and “interesting” resources to demonstrate their benefits. However, without the applications there is little incentive to make servers and resources available. While there is certainly a grain of truth in this argument, other technologies have successfully taken this hurdle, namely file swapping by peer-to-peer systems, which derives its popularity largely from the amount and quality of files available for download (other factors such as usability issues play a role as well). The difference is that, first, peer-to-peer file swapping implements an infrastructure service (peer-to-peer networks) with a compelling application (*e.g.*, accessing large volumes of music files easily), and second, the advantages of the application quickly manifest even in small communities. Put simply, it is not the chicken and egg situation that deters the deployment of mobile agents, it is the lack of a compelling robust application. It should be noted, though, that whether or not this application actually *requires* mobile agents is principally irrelevant, mobile agents can be adopted in the wake of a successful application which acts as a door opener.

3. Security or Lack Thereof

Another frequent explanation why mobile agents have not yet been adopted is their inherent security risks: first, malicious hosts may tamper with agents, and second, malicious agents may attack their hosts. I would hesitate to regard the first risk as a primary reason. Current Internet users generally appear to have little risk awareness; they surf the Web with insecure browser configurations, never encrypt electronic mail, install software of dubious origin without hesitation, or click on the OK button if they do not understand what they are supposed to do (but what to get

the operation done). Even users with higher sophistication than average and a feeling for the risks may accept a higher risk for the sake of convenience. It is unconceivable why Internet users would (or even should) be concerned more about their mobile agents than they are about their other activities. If, on the other hand, mobile agents are deployed within an organization then the threat model changes and the aforementioned risk becomes even more irrelevant.

I believe that the second risk is a graver issue. In any real setting it is imperative that a mobile agent system retains a high availability, integrity, and scalability while imposing a low maintainance overhead. This requirement conflicts with the near-ubiquitous adoption of Java as the implementation platform for mobile agent systems because Java falls short of providing the level of security that is required [2]. The standard edition of Java is voluminous and contains but is not limited to *e.g.*, the following deficiencies:

- lack of resource control; respective extensions cannot be expected in the near future;
- lack of application separation; this is addressed in JSR 212 [1], but respective extensions will not likely be included before JDK Version 1.6 (current version is 1.4);
- there is no safe method to force a Java thread to stop; adverse code may easily catch any exceptions pertaining to its elimination;
- the Garbage collector thread may be hijacked by directly or indirectly overriding finalization methods;
- adverse code may block on globally visible class locks, thereby locking other threads vital to the functioning of the runtime system;
- the security model is flexible but access control checks are dispersed throughout the installed classes; a single unguarded privileged action implementation easily undermines the security of the virtual machine.

The list could be continued, but it is already obvious that the limitations of Java for building safe and secure mobile agent systems will not be eliminated in the near future. The Micro Edition, on the other hand lacks vital features required for mobile code such as custom class loaders. While Java initially started out as a language with mobile code support, it becomes clear that Java is evolving into a general-purpose programming language with objectives that diverge from the requirements of mobile agents.

Hence, one might say that Java, as a basis for mobile agents, has simultaneously been a fortune and a misfortune. On the one hand Java makes it easy to build mobile agent systems, and hence allowed a larger community to experiment with the technology. On the other hand Java makes it next to impossible to build and maintain a publicly deployed and dependable mobile agent system. The mobile agent community should again consider alternatives (past

examples include [6, 5, 4, 7, 8]) that do one thing and do it really well: supporting mobile agents efficiently, safely, and securely. Notwithstanding operation on systems with plentiful resources, this alternative should be tailored to mobile, embedded, and resource-constrained devices on which mobile agents can leverage their advantages. The alternative should lend itself to compact, efficient, and secure implementation, if necessary, at the expense of a reduced feature set.

References

- [1] Application isolation API specification. <http://www.jcp.org>, October 2003. JSR 121.
- [2] Walter Binder and Volker Roth. Secure mobile agent systems using Java – where are we heading? In *Proc. 17th ACM Symposium on Applied Computing, Special Track on Agents, Interactions, Mobility, and Systems (SAC/AIMS)*, Madrid, Spain, March 2002. ACM.
- [3] Colin G. Harrison, David M. Chess, and Aaron Kershbaum. Mobile agents: Are they a good idea? Technical report, IBM T. J. Watson Research Center, P.O. Box 704, Yorktown Heights, NY 10598, March 1995.
- [4] Robert S. Gray. Agent Tcl: A flexible and secure mobile-agent system. In *Proc. of the 1996 Tcl/Tk Workshop*, pages 9–23, July 1996.
- [5] Dag Johansen, Robbert van Renesse, , and Fred B. Schneider. An introduction to the TACOMA distributed system version 1.0. Technical Report 95-23, Department of Computer Science, University of Tromsø, June 1995.
- [6] Holger Peine. *Run-Time Support for Mobile Code*. Ph.d. thesis, University of Kaiserslautern, Kaiserslautern, Germany, 2002. ISBN 3-925178-93-7.
- [7] Christian Tschudin. The Messenger Environment M0 - a condensed description. In J. Vitek and C. Tschudin, editors, *Mobile Object Systems*, volume 1222 of *Lecture Notes in Computer Science*, pages 149–156. Springer Verlag, 1997.
- [8] Pawel Wojciechowski and Peter Sewell. Nomadic Pict: Language and Infrastructure Design for Mobile Agents. In *Proc. First International Symposium on Agent Systems and Applications, and Third International Symposium on Mobile Agents (ASA/MA '99)*, volume 2, pages 821–826, October 1999.