Mobile Agents: Right Concept, Wrong Approach

Dag Johansen
Dept. Computer Science, University of Tromsø, Norway
dag@cs.uit.no

Abstract

This position paper looks back and takes stock in the context of mobile agent technologies[1]. Our claim is that mobile agent technologies received wrong attention. Partially because of this, the technology is still too premature. Though, software mobility is a far too fundamental structuring abstraction to be considered useless. Our conjecture is that this concept can be applied in larger scale once web service architectures become an Internet commodity.

1. Introduction

Distributed systems architects use a number of fundamental structures and principles. One example is *caching* used to support disconnected operations and improve availability. Another is *replication* used to achieve improved reliability, both for processes and data. A third example is *multiplexing* used to share computer resources among a set of users. Interest for and applicability of these and similar fundamental principles vary over time, *migration* being a good example.

Migration has surfaced several times as a structuring abstraction at least the last 30 years. The basic idea is to move some computational structure over the network and execute it remotely. Internet worms, remote evaluations, SQL, mobile code, viruses, mobile objects, Postscript, migrating processes, applets, and servlets are broadly speaking examples of code (and state) that can be sent over the network for remote execution.

The latest incarnation, developed both to reduce network latency and as a location sensitive programming abstraction, is *mobile agents*. A mobile agent is a piece of code and its associated data moving about executing autonomously on behalf of its owner.

[1]This research was funded in part by the Norwegian Research Council (IKT-2010 Program).

2. Wrong Approach

Advocates of mobile agents find this concept appealing for a number of reasons. One is for installing computations close to data or computing resources to make more efficient use of available communication bandwidth, another is to cope with network partitions.

Opponents, however, focus more on problems, problems that are closely related to general mobile software structures. Host and agent integrity are obvious technical ones, where a visiting host must be safeguarded from a malevolent visiting agent and vice versa. Lack of a widely accepted agent deployment infrastructure, and no common mobile agent standard are others. Closely related, and often quoted as the main problem with this concept, is lack of a mobile agent killer application.

At a first glance, opponents of mobile agents might seem right. Mobile agent advocates have not satisfactorily solved the problems outlined above. And, it is to a large extent our own fault as a community, since far too many groups have focused their efforts in building yet another (Java-based) mobile agent system. More than 100 mobile agent systems have been built, where the majority, to be modest, hardly provides any incremental contribution to the community as a whole. Yet another Java-based mobile agent system taking advantage of, for instance, Java serialization and reflection support, does not provide much new insight into the fundamental mobile agent challenges outlined above.

The initial industrial [1][2] and academic [3][4] mobile agents systems developed in the mid-90ties washed out many of the fundamental abstractions and problems. The majority of the following projects, however, should have adopted one or a few of these systems and focused on the more fundamental problems yet to be solved.

3. Right Concept

The interesting question is to where from here with the mobile agent concept. We conjecture that mobile agents, even if not wrapped as that, will emerge and have

Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM'04)
0-7695-2070-7/04 $20.00 © 2004 **IEEE**

impact soon. This is because industry is busily deploying mobile agent platforms for us, in the guise of web services. Fundamentally, a web service down-loaded over the network resembles a single-hop mobile agent. Hence, once a platform like, for instance, Microsoft .Net is widely deployed and used, we have a potential platform for mobile agents. This is not a surprise since the first .Net architecture came from Microsoft's initial attempt to build a mobile agent system supporting multiple languages.

Our conjecture is based on applicability experience with mobile agents in the project TACOMA[2] for 10 years. In TACOMA, we used mobile agent technologies as a web service infrastructure. Initially, we called software modules and their associated state mobile agents. In retrospect, we could rather have named them web services. TACOMA is basically a remote installation toolkit for software extensions (which can contain data) written in almost any language, whether it is legacy code or a new agent written for extending a remote server with a new API. Thus, a TACOMA server is extended and configured at run-time based on client web services and their systems needs. These web services can be fetched from a trusted third-party over the network, or from the client itself. A special case is to move these web services about among multiple servers being used.

We still also conjecture that this concept should find its applicability in asymmetric, ubiquitous distributed computing. Cellular and PDA technologies are about to merge, and our hypothesis is that it should be possible to off-load computations from these light clients into the Internet. One example of how we envision this (and actually build an infrastructure for) is outlined below.

In WAIF[3], we demonstrate how the next generation Internet can be made programmable and extensible with personalized, mobile software. Our goal is to replace the old, time-consuming pull-based Internet, with a push-based one delivering high-precision information in a timely manner. Our infrastructure in this proactive web service environment is extremely asymmetric. The client is basically a proxy storage device, much like a Giga-storage equipped PDA-cellular. The servers, though, act as virtual extensions to these portable client devices by accepting client code and data for execution. The infrastructure is much like in Oxygen[4], but we use this pervasive infrastructure embedded in our environment as a virtual remote computer. The client now rents services from the environment, the environment guarantees resilience of this on-demand operating environment.

To do this, we need to off-load computations from the client. We install them close to data sources as leaf nodes in a personal overlay network system (PONS) serving a single user. Additionally, fusion, distribution, and content-based filtering components are installed as single-hop mobile agents. This way, we build autonomous distributed filtering networks that work autonomously on behalf of a user. This is not a single agent jumping itinerantly about, but a troop of agents extending Internet servers. Notice again how this mobile agent inspired approach converges with how web services architectures potentially can be used.

4. Concluding Remarks

To conclude, a main reason for mobile agent technologies being considered a neat, but not a necessary paradigm, is that we did not converge on a complete programming, operating and management environment for mobile agents. We as a community did not solve properly the hard problems like, for instance, security and fault-tolerance. Neither have we converged on a standardized protocol and programming model for Internet applications using our technology.

The positive message is that we probably have had in-direct impact through conceptual contributions and trained way of thinking; mobile agent structures might be found in, for instance, industrial web services architectures, IBM's autonomic computing initiative, or Intel's pervasive computing initiative. Hence, we still consider software mobility as a fundamental structure that will find its applicability in future real-world applications. However, we are less sure that they will be named mobile agents.

References

[1] J.E. White, Telescript Technology: Mobile Agents, General Magic White Paper, appeared in J. Bradshaw, *Software Agents*, AAAI/MIT Press, 1996.

[2] D. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents with Aglets*, Addison Wesley, IA, 1998.

[3] D. Johansen, R. van Renesse, and F.B. Schneider, Operating System Support for Mobile Agents, *Proc. of the 5th IEEE HOTOS*, Orcas Island, Wa, May 1995.

[4] R. Gray, Agent Tcl: A Transportable Agent System. *Proc. of CIKM Workshop on Intelligent Information Agents*, Baltimore, MA, December 1995.

[2] http://www.tacoma.cs.uit.no

[3] http://www.waif.cs.uit.no

[4] http://www.oxygen.lcs.mit.edu