

“Technology of Mobile Agent in Software Agent: A Case Study of Future Distributed Computing”

Paritosh Kumar Bansal

Research Scholar, CMJ University, Shillong, Meghalaya, India

Abstract – *This paper will lead you into the world of mobile agents, an emerging technology that makes it very much easier to design, implement, and maintain distributed systems. You will find that mobile agents reduce the network traffic, provide an effective means of overcoming network latency, and perhaps most importantly, through their ability to operate asynchronously and autonomously of the process that created them, helps you to construct more robust and fault-tolerant. Read on and let us introduce you to software agents - the mobile as well as the stationary ones. We will explain all the benefits of mobile agents and demonstrate the impact they have on the design of distributed systems before concluding this paper with a brief overview of some contemporary mobile agent systems.*

In this paper, we discuss the potential uses of mobile agents in Network Management. We define software agents and a navigation model, which determines agent mobility. We list a number of potential advantages and disadvantages of mobile agents. We include a short commentary on the ongoing standardization activity. The core of the paper comprises descriptions of several actual and potential applications of mobile agents in the five OSI functional areas of Network Management.

Over the last couple of years we have been working on the development of mobile agents systems and its application to the areas of telecommunications and network management. This work path produced positive results: a competitive mobile agent platform was built, the run-time benefits of mobile agents were proved, and our industrial partners have developed practical applications that are being integrated into commercial products.

However, despite the positive results, we feel that mobile agent technology is still not ready to enter the path of mainstream software development. In our perspective, one of the main reasons for this situation arises from the traditional approach to mobile agent technology. This approach, based on the familiar concept of the mobile agent distributed platform as an extension of the operating system, focuses too much on the mobile agents and associated issues (mobility, agent lifecycle, security, coordination, etc.) and provides poor support for the development of applications where mobile agents are just one of several available technologies.

Learning from past experience, we are now working on a new approach where the focus is brought back to the applications and mobile agents become just one the tools available to develop distributed systems. This provides a much lighter framework for application-based mobile agent systems.

INTRODUCTION

A mobile agent is not bound to the system where it begins execution. It has the unique ability to transport itself from one system in a network to another. The ability to travel, allows a mobile agent to move to a system that contains an object with which the agent wants to interact, and then to take advantage of being in the same host or network as the object.

Software agents are a major innovation in how people use information systems, and they have parallels with how Decision Support Systems (DSS) support human decision-making. A DSS approach to the development of software agents suggests a highly interactive and flexible interface between the agent and its user, and addresses some potential barriers to the successful adoption of agent technologies

Within a DSS model, agents can be classified as providing

search, choice or interface support. Each of these classifications uses techniques originating from separate disciplines and requires different performance measures. We use a real estate agent as a metaphor to examine the descriptive, procedural and semantic knowledge bases that agents can use to support search and choice activities in an e-commerce domain.

Software agent system designers must address the "connection problem" - how does the agent find information sources and other agents to assist in achieving its goals. In controlled systems, collaborative agents can request and obtain descriptive knowledge that has been collected by other agents [Ackerman et al. 1997]. In open environments, most agent systems use directories, matchmakers, and brokers to identify potential information sources [Brenner et al. 1998],

Software agents can ask clients to state their information needs.² A form or questionnaire can be used to elicit the representation requirements where information is highly structured. However, where information needs are complex and ill-structured, more open processes of collection may be required and these processes can be time consuming and inaccurate. Significant efforts have been made to design software agents that use proxy information to develop profiles [Rucker et al. 1997] or learn their user's preferences by observing behavior [Lieberman; Ngu et al. 1997]. Collaborative filtering compares profiles to find users with similar information needs so that information judged relevant by one user can be shared with others.

This is another way to reduce the profiling effort required by each user [Balabanovic et al. 1997].

Although an agent-based system can be implemented with any client/server technology, it differs from classical client/server systems, because there is no clear distinction between a client and a server. All agents participate in the computation according to the role statically assigned by the designer or dynamically assigned by a human or agent supervisor. In this paper, we argue that agent mobility benefits agent-based network management systems with additional advantages over classical client/server (i.e., manager/agent) implementations, as well as systems built with stationary agents.

PLATFORM-BASED MOBILE AGENT SYSTEMS

The JAMES project took place from March 1998 to December 1999, and involved two teams. The first team developed a MA infrastructure specifically tuned for NM applications. Like other known MA implementations, the idea was to have a set of platforms providing an execution

environment for mobile agents, controlling their migration and lifecycle. These platforms are seen as an extension of the host's operating system (typically one platform per host) where mobile agents from different applications coexist (see Figure 1). Following this *platform-centered* design we focused on issues like performance and robustness, integration with legacy management technologies, infrastructure manageability, agent coordination and security.

The second team used this platform to develop management applications. Four prototypes were produced and evaluated, and two of them have been selected for integration into commercial products. This separation between platform and applications, as well as the desire to produce commercial products, provided a good perspective on the reasons why there are so many mobile agent infrastructures and so few real world applications. This perspective can be described in three dimensions: the programmer, the user (costumer) and the application field (in our specific case network management).

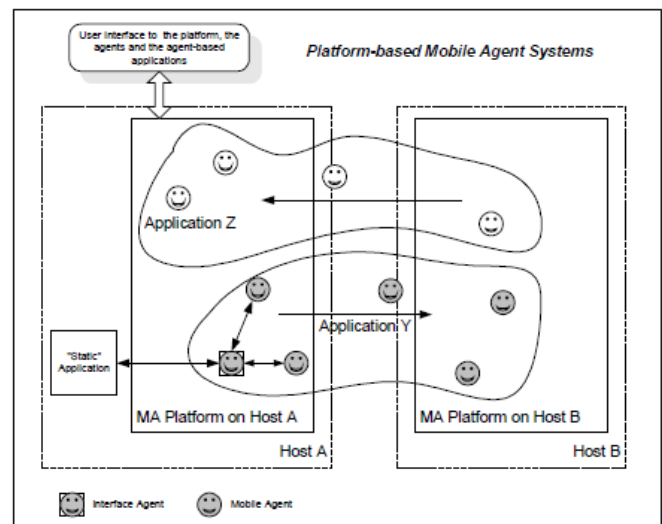


Figure 1 – Platform-based mobile agent systems

A. *The Programmer* - From the point of view of the programmer, constructing an application that uses mobile agents is a difficult process. Current mobile agents systems force the development to be centered on the agents, many times requiring the applications themselves to be coded as a special type of agents – *stationary agents*. When this does not happen, special interface agents (*service agents*) have to be setup between the application and the incoming agents. These agents must know how to speak with the mobile agents and with the application.

Although the mobile agent concept – *a thread that can*

move to another node, is a very useful structuring primitive, all the currently required setup to use it is an overkill that prevents acceptance by the developers.

B. The User - From the viewpoint of the user, if an application will make use of mobile agents it is necessary to first install an agent platform. The security permissions given to the incoming agents must also be configured and the proper hooks necessary to allow the communication between the agents and the application must also be setup. While some of these tasks can be automated using installation scripts, this entire setup package is too much of a burden. Usually, the user is not concerned with mobile agents nor wants to configure and manage mobile agent platforms. The user is much more concerned with the applications than with the middleware they are using in the background. In the currently available mobile agent systems, the agents are central and widely visible. They are not the background middleware but the foreground applications.

The term mobile code also has very strong negative connotations that make the dissemination of the MA technology difficult. The user is afraid of installing a platform capable of receiving and executing code without his permission. This happens even though the existence of mobile code is present in technologies like Java, in particular in RMI and JINI. The fundamental difference is that in those cases, the user is shielded from the middleware being used. In many cases, using mobile agents does not pose an increased security threat, especially if proper authentication and authorization mechanisms are in place¹. However, because the current agent platforms do not shield the user from the middleware, the risk associated with this technology is perceived as being higher, which causes users to back away from applications that make use of mobile agents.

C. The Application Field - The primary application field of the JAMES project was network management. The mobile agent paradigm fits very well into the conceptual foundations of distributed and delegated network management. This is one of the reasons why NM is so often considered one of the most attractive demonstration fields for MA technology. Nevertheless, in general, the NM community is still reluctant to accept the advantages of mobile agents, even when Active Networks, which share some of the same principles and technologies, is one of the most promising NM research areas.

One of the main reasons is that mobile agents tend to be exclusively associated to multi-hop migration, where an agent successively visits several network nodes, while performing a given task. This multi-hop model is seldom the most appropriate delegation technique for NM

applications. Nevertheless, there are other models for using mobile agents.

SOFTWARE AGENTS

A *mobile agent* is a software agent that can move between locations. This definition implies that a mobile agent is also characterized by the basic agent model. In addition to the basic model, any software agent defines a *life-cycle model*, a *computational model*, a *security model* and a *communication model* (Green et al.,). A mobile agent is additionally characterized by a *navigation model*.

Mobile agents can be implemented using one of two fundamental technologies: *mobile code* (Baldi et al.,) or *remote objects* (Vinoski,). Examples of the former approach include AgentTCL (Gray) and Telescript (White,), and the latter - Aglets (Lange,).

To make use of mobile agents, a system has to incorporate a mobility framework. The framework has to provide facilities that support all of the agent models, including the navigation model. For the life-cycle model, we need services to create, destroy, start, suspend, stop, etc., agents. The computational model refers to the computational capabilities of an agent, which include data manipulation and thread control primitives. The security model describes the ways in which agents can access network resources, as well as the ways of accessing the internals of the agents from the network. The communication model defines the communication between agents and between an agent and other entities (e.g., the network). All issues referring to transporting an agent (with or without its state) between two computational entities residing in different locations are handled by the navigation model. Obviously, the framework incurs certain costs including increased memory requirements and execution and access delays on every participating device. The underlying technology, however, is evolving rapidly. For example, the footprints of certain Java Virtual Machines (JVM), which are the basis for many mobile agent frameworks, are very small making them suitable for embedded systems. We believe that the use of Java chips will be important in the future networked devices. In addition, forthcoming new software packages like Jini address many of the needs of agent-based systems.

DSS are software applications that have been used over the last few decades to provide support for many structured and unstructured problems such as strategic planning, investment planning, stock portfolio management, enterprise planning, human resources management, supply chain planning, knowledge management, case-based reasoning and help desk automation (Clemen, 1996; Mallach, 2000; Marakas, 1998;

Mora, Forgionne, & Gupta, 2002; Turban and Aronson, 1997). DSS components such as knowledge management systems, model management systems and data management systems aid humans in making better decisions by incorporating previous knowledge and information about the domain. Over the past two decades, decision support systems and software agent enabled systems have been researched independently both in academia and in commercial applications. Several businesses have successfully implemented decision support systems to solve problems including human resources management, supply chain planning, help desk automation, and placement of new office locations.

Intelligent software agents provide a technology that can be used to obtain knowledge from dynamically changing environments and thus potentially allowing DSS users to make more informed and accurate decisions (Mora et al., 2002). Software agents (Russell & Norvig, 2003; Weiss, 1999) are used widely in various applications such as searching information on the Web (Dasgupta, Narasimhan, Moser, & Melliar-Smith, 1999; Knapik & Johnson, 1997), tracking browsing behavior of online users (Sahai, Billiard & Morin, 1997), implementing trading algorithms for online auctions (Sandholm, 2002), assisting humans in online activities such as filling forms or presenting information in a concise form (Padgham & Winikoff, 2004), and, even for security and privacy applications such as detecting spyware, implementing security policies on Web servers, and, filtering spam (Dasgupta, Moser & Melliar-Smith, 2000).

Good Reasons for Using Mobile Agents

Although mobile agent technology sounds exciting, our interest in mobile agents should not be motivated by the technology *per se*, but rather by the benefits they provide for the creation of distributed systems. So here are seven good reasons for you to start using mobile agents.

They reduce the network load. Distributed systems often rely on communications protocols that involve multiple interactions to accomplish a given task. This is especially true when security measures are enabled. The result is a lot of network traffic. Mobile agents allow you to package a conversation and dispatch it to a destination host where the interactions can take place locally, see Figure 2. Mobile agents are also useful when it comes to reducing the flow of raw data in the network.

When very large volumes of data are stored at remote hosts, these data should be processed in the locality of the data, rather than transferred over the network. The motto is simple: move the computations to the data rather than the data to the computations.

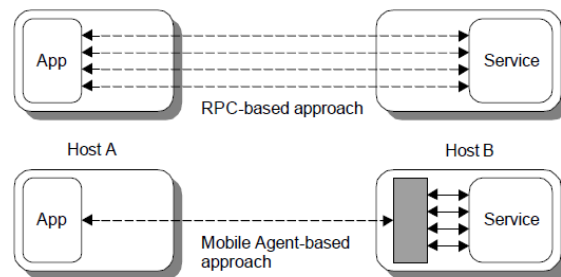


Fig. 2. Mobile Agents Reduce Network Load.

They overcome network latency. Critical real-time systems such as robots in manufacturing processes need to respond to changes in their environments in real time. Controlling such systems through a factory network of a substantial size involves significant latencies. For critical real-time systems, such latencies are not acceptable. Mobile agents offer a solution, since they can be dispatched from a central controller to act locally and directly execute the controller's directions.

They encapsulate protocols. When data are exchanged in a distributed system, each host owns the code that implements the protocols needed to properly code outgoing data and interpret incoming data, respectively. However, as protocols evolve to accommodate new efficiency or security requirements, it is a cumbersome if not impossible task to upgrade protocol code properly. The result is often that protocols become a legacy problem. Mobile agents, on the other hand, are able to move to remote hosts in order to establish "channels" based on proprietary protocols.

They execute asynchronously and autonomously. Often mobile devices have to rely on expensive or fragile network connections. That is, tasks that require a continuously open connection between a mobile device and a fixed network will most likely not be economically or technically feasible. Tasks can be embedded into mobile agents, which can then be dispatched into the network. After being dispatched, the mobile agents become independent of the creating process and can operate asynchronously and autonomously, see Figure 3. The mobile device can reconnect at some later time to collect the agent.

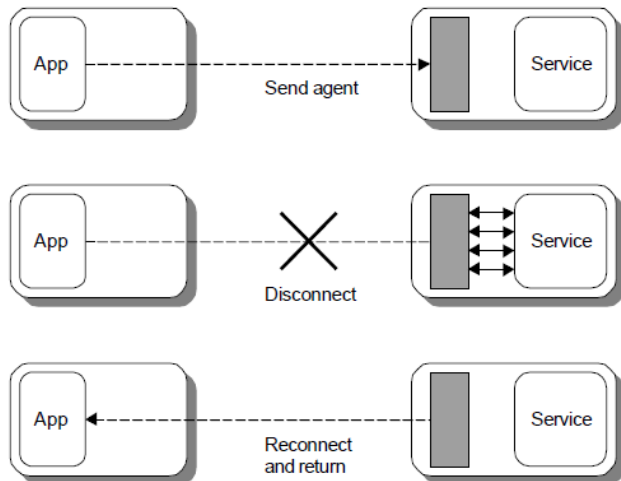


Fig. 3. Mobile Agents Allow Disconnected Operation.

They adapt dynamically. Mobile agents have the ability to sense their execution environment and react autonomously to changes. Multiple mobile agents possess the unique ability to distribute themselves among the hosts in the network in such a way as to maintain the optimal configuration for solving a particular problem.

Software Agent properties

Software agents differ from traditional software because they exhibit three traits not commonly found in the traditional software, which include:

- the ability to personalize,
- the ability to be proactive, and
- the ability to be adaptive.

Personalization is the capturing of the unique goals of the user and acting accordingly on his/her behalf. Personalization is apparent for example in a user ability to customize informational searches. The ability to personalize software agents allows the user to construct their individual preferences for searching an on-line database, a list of sites, and even transacting on your behalf.

A software agent is proactive if it has the ability to complete its task independent of the user activities. An agent that finds new information for the user, without the user requesting a response is an example of the ability for the agent to be proactive.

The ability to be adaptive is described as the software agent ability to learn from its past actions and from its environment. Learning from past actions is characteristic

of the artificial intelligence (AI) technology which preceded much of today agent technology. Software agents have the following properties, which distinguish them from other programs:

Intelligence - Software agents employ techniques from the field of artificial intelligence, which empower them with a fair degree of intelligence and common sense. For example, the travel agent program should realize that people generally do not prefer travelling by flights that depart or arrive at the airport late in the night and the agent should avoid booking tickets on such flights. The travel agent program should be smart enough to bargain and arrange the trip so that the overall expenditure for the trip is as low as possible without compromising on the user's preferences.

Autonomy - The agents themselves decide the sequence of actions to be performed to achieve the user's task. This autonomy enables agents to operate without requiring human intervention. Once the specifications are given to the travel agent, it should be able to proceed on its own to arrange the trip for the user without requiring the user to constantly monitor the agent.

Responsiveness - Agents perceive their environment (which may be the Internet, a collection of other agents, etc.) and respond in a timely fashion to changes that occur in it. At the same time, agents should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-oriented behavior and take the initiative when appropriate.

Communicative Ability - Software agents should provide a user friendly interface so that the user can easily interact with the agent. Agents are social entities and often communicate and collaborate with one another in order to complete their tasks. For example, the travel agent program of one user must be able to communicate with other travel agents to find out about hotels which customers disliked and avoid such hotels.

Mobile Agent Applications

We will now take a closer look at some applications that benefit particular from the mobile agent paradigm. Please note that this is by no means intended to be an exhaustive list.

Electronic commerce. Mobile agents are well suited for electronic commerce. A commercial transaction may require real-time access to remote resources such as stock quotes and perhaps even agent-to-agent negotiation. Different agents will have different goals, and will implement and exercise different strategies to

accomplish these goals. We envision agents that embody the intentions of their creators, and act and negotiate on their behalf. Mobile agent technology is a very appealing solution to this kind of problem.

Personal assistance. The mobile agent's ability to execute on remote hosts makes it suitable as a "assistant" capable of performing tasks in the network on behalf of its creator. The remote assistant will operate independently of its limited network connectivity, and the creator can feel free to turn his or her computer off. To schedule a meeting with several other people, a user could send a mobile agent to interact with the representative agents of each of the people invited to the meeting. The agents could negotiate and establish a meeting time.

Secure brokering. An interesting application of mobile agents is in collaborations where not all the collaborators are trusted. In this case, the involved parties could let their mobile agents meet on a mutually agreed secure host, where collaboration can take place without the risk of the host taking the side of one of the visiting agents.

Distributed information retrieval. Information retrieval is an often-used example of a mobile agent application. Instead of moving large amounts of data to the search engine so that it can create search indexes, you dispatch agents to remote information sources, where they locally create search indexes that can later be shipped back to the origin. Mobile agents are also able to perform extended searches that are not constrained by the hours during which the creator's computer is operational.

Telecommunication networks services. Support and management of advanced telecommunication services are characterized by dynamic network reconfiguration and user customization. The physical size of these networks and the strict requirements under which they operate call for mobile agent technology to form the "glue" that keeps such systems flexible yet effective.

Workflow applications and groupware. It is in the nature of workflow to support the flow of information between co-workers. The mobile agent is particularly useful here since, in addition to mobility, it provides a degree of autonomy to the workflow item. Individual workflow items fully embody the information and behavior needed for them to move through the organization independent of any particular application.

Monitoring and notification. This is one of the "classical" mobile agent applications that highlight the asynchronous nature of mobile agents. An agent is able to monitor a given information source without being dependent on the location from which it originates. Agents can be dispatched

to wait for certain kinds of information to become available. It is often important that monitoring agents have life spans that exceed or are independent of the computing processes that create them.

Information dissemination. Mobile agents embody the so-called Internet "push" model. Agents are able to disseminate information such as news and automatic software updates for vendors. The agents will bring the new software components as well as the installation procedures directly to the customer's personal computer and will autonomously update and manage the software on the computer.

Parallel processing. Given that mobile agents can create a cascade of clones in the network, one potential use of mobile agent technology is to administer parallel processing tasks. If a computation requires so much processor power as to that it must be distributed among multiple processors, an infrastructure of mobile agent hosts could be a plausible way to get the processes out there.

CONCLUSION

With agent *mobility* being the focus of this paper we defined a mobile agent as an agent that is not bound to the system where it begins execution. It has the unique ability to transport itself from one system in a network to another. The ability to travel, allows a mobile agent to move to a system that contains an object with which the agent wants to interact, and then to take advantage of being in the same host or network as the object. We gave you seven good reasons for you to start using mobile agents: they reduce the network load, they overcome network latency, they encapsulate protocols, they execute asynchronously and autonomously, they adapt dynamically, they are naturally heterogeneous, and they are robust and fault-tolerant.

Among some of the application domains that benefit from mobile agent technology are: electronic commerce, personal assistance, secure brokering, distributed information retrieval, telecommunication networks services, workflow applications and groupware, monitoring and notification, information dissemination, and parallel processing.

In this paper, we discussed the use of mobile agents for managing networks. Much of the research in this area is in its infancy, so the available resources are scarce. We used our own experience to illustrate application ideas in several areas of network management. Throughout the paper, we provided a number of references to related activities at other research centers. In the preceding

section, we included an overview of the research conducted at several academic and industrial institutions, which are involved in the research on network and system management with mobile agents. Over the past two years we have been trying to bring mobile agent technology to the mainstream development of management applications. We have gathered a strong experience on the advantages and shortcomings of current mobile agent technology and its deployment on real systems.

The main conclusion from that experience is that although it is complicated to develop systems on the current state of the technology, the mobile agent paradigm provides important advantages in the context of network management and other application domains.

REFERENCES

- Aridor, Y. and Lange, D.B.: Agent Design Patterns: Elements of Agent Application Design, In Proceedings of the Second International Conference on Autonomous Agents (Agents '98), ACM Press, 1998, pp. 108-115.
- Cheng, D. T. and Covaci, S., *The OMG Mobile Agent Facility : A Submission*, in Rothermel, K. and Popescu-Zeletin, R., Eds., *Mobile Agents*, Springer-Verlag, 1997.
- D. Kotz, R. Gray, "Mobile Agents and the Future of the Internet", in ACM Operating Systems Review, 33(3), 1999.
- El-Darieby, M., *Intelligent Mobile Agents for Network Fault Management*, Technical Report SCE-98-13, System and Computer Engineering, Carleton University, 1998.
- Karjoth, G., Lange, D.B., and Oshima, M.: A Security Model for Aglets, IEEE Internet Computing 1, 4, 1997, pp. 68-77.
- Lange, D.B. and Oshima, M.: Programming and Deploying Java™ Mobile Agents with Aglets™, Addison-Wesley, 1998.
- L. Silva, G. Soares, P. Martins, V. Batista, L. Santos, "Comparing the Performance of Mobile Agent Systems: A Study of Benchmarking", Computer Communications, Volume 23, Issue 8, April 2000.
- Magedanz, T., Rothermel, K. and Krause, S., *Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?*, IEEE INFOCOM 1996, San Francisco, USA, March 24-28, 1996
- Nwana, H. S., *Software Agents: An Overview*, Knowledge Engineering Review, Vol. 11, No 3, pp.1-40, Sept 1996.
- P. Simões, L. Silva, F. Boavida, "A Generic Management Model for Mobile Agent Infrastructures", in Proceedings of SCI'2000/ISAS'2000, Orlando, July 2000.
- The Object Management Group: The Mobile Agent System Interoperability Facility, OMG TC Document orbos/97-10-05, The Object Management Group, Framingham, MA., 1997.
- White, J.: Mobile Agents, In Software Agents, Bradshaw, J. Ed., MIT Press, 1997.