

Title	モバイルエージェント・マネジメントシステムに関する研究
Author(s)	越原, 智弘
Citation	
Issue Date	2001-03
Type	Thesis or Dissertation
Text version	none
URL	<a href="http://hdl.handle.net/10119/1466">http://hdl.handle.net/10119/1466</a>
Rights	
Description	Supervisor:片山 卓也, 情報科学研究科, 修士

# Mobile Agent Management System

Tomohiro Koshihara

School of Information Science,  
Japan Advanced Institute of Science and Technology

February 15, 2001

**Keywords:** mobile agents, agent systems, mobile computing, mobile users, agent management, user interface.

## Background

The ubiquitous use of the computer and the Internet introduces a new computing environment. This new environment requires support for code and user mobility. In the last decade, the mobile computing paradigm has received much attention in the literature. The most promising paradigm for supporting user and code mobility is the mobile agent paradigm.

Mobile agents are software programs that can migrate autonomously from node to node in the distributed system to perform computation on behalf of their owners, either a program or a user. Mobile agents have several properties including autonomy and mobility. They offer several advantages including support for weak clients and disconnected operations, support for dynamic environments, scalability, higher performance and lower bandwidth usage. Mobile agents can move from host to host in the network looking for data and processing the data locally without putting further load on the network. In order to execute a mobile agent at a host, the host must support an agent execution environment also called an agent system. Several agent systems have been developed including Aglets, Voyager, Bee-gent, Plangent, AgentSpace and many more.

Typically, a user creates a mobile agent and sends it to the network to perform some tasks. The mobile agent will then move from host to host in order to perform the assigned tasks and will return to the user once it completes its tasks. During its migration from host to host, the user does not have access to information on the current state or location of the mobile agent making it impossible for the user to manage and/or have control over his remote agents. Furthermore, each mobile agent system has its own user interface. In order to manage his local agents, a user must then use several agent system interfaces,

one for each agent type he might create.

Recently, there is a large increase in the usage of mobile terminals. Such devices can be frequently disconnected from the network and may have poor network connectivity. Moreover, they can connect to the network from different locations with different address. After disconnecting from the network, it is important to ensure that the mobile user can still retrieve its mobile agents when he reconnects to the network possibly at a different location.

## Proposal

To address the issues and problems described above, we propose the Mobile Agent Management System (MAMS). The proposed system provides an elegant, modular and flexible solution to these problems.

The proposed MAMS system provides a unique framework from which a user can manage and control its mobile agents whether its mobile agents are being executed locally or at a remote location. The proposed system introduces a mechanism that allow the user to manage and have control over his local and remote mobile agents. The user can then dynamically communicate with his agents, receive information on the current status of its agents, and possibly assign new tasks to its local and/or remote agents. The proposed system is also highly modular and flexible in that it supports different types of mobile agents (i.e. Aglets agents, AgentSpace agents) and provide the user with a unified framework and interface from which he can control all his agents.

MAMS allow the user to easily manage and control his agents regardless of the following points:

- The mobile agent location (local or remote agent).
- The mobile agent type.
- The mobile user location and connection point.

## Mobile Agent Management System

The mobile agent management system consists of three layers. Each layer represents and implements a different functionality in the system. The three layers consist of the user interface layer, the agent management layer, and the agent system interface layer.

The user interface layer provides a unified interface to the user through which he can manage and control his mobile agents. It also allows the user to retrieve information on its mobile agents and to assign new tasks to its mobile agents. This layer allows the user to manage its mobile agent independently from the mobile agent type and/or location.

The agent management layer provides information on the agent selected at the user interface and sends this information and/or requests to the agent system interface layer. It also implements some operations that might be performed on the mobile agents and

that appear on the MAMS user interface. This layer ensures communication with local and remote mobile agents and provides information on the current status and location of the user's agents. The agent management layer supports the repository agent, which is an agent that provides the MAMS system with the support required for disconnected operations. Finally, this layer interacts with the agent system interface layer in order to send requests to the supported agent systems depending on the type of the mobile agent.

The agent system interface layer allows the above layers to interact with different agent systems in order to support different types of mobile agents. It wraps and transmits requests from above layers depending on the interface defined at each supported agent system.

To allow remote mobile agents to execute a new operation, that is an operations that is assigned to the mobile agent after it migrates from the user's local host, the MAMS system must be able to retrieve the current location of the remote mobile agent. In the proposed system, we use a method called "registry" to retrieve the agent current location.

The agent system sends information including the agent state and its location to a host called "repository". At the repository host we implement a database called "register" where we save the information received from the agent system. We call a "state update" the operation that consists on saving the mobile agent's information in the register at the repository. A repository agent that is executing at the repository host implements the mechanisms used to support mobile users and disconnected operations. Mobile agents that completed their tasks and that cannot return to the user's host because it is disconnected from the network are directed to the repository where they can wait for the user to reconnect to the network before they finally migrate to the user's host.

## Conclusion

The proposed MAMS system enables the user to have better control and to easily manage its mobile agents. The proposed system offers several advantages including:

- Easily manage and control local and remote agents.
- Provide support for mobile users.
- Handle different types of mobile agents created using different agent systems
- Provide a unified framework and interface for the user to manage and deal with his agents.

The MAMS system as defined in this research work does not consider some issues and does not provide solution to some problems such as the security problem and fault tolerance. As a future research work, it is important to address and explore such issues and to provide solutions for these problems within the MAMS framework.