

Seven Good Reasons for Mobile Agents

Dispatch your agents; shut off your machine.

Danny B. Lange and Mitsuru Oshima

Mobility is an orthogonal property of agents, that is, not all agents are mobile. An agent can just sit there and communicate with its environment through conventional means, such as remote procedure calling and messaging. We call agents that do not or cannot move “stationary agents.” A stationary agent executes only on the system on which it begins execution. If it needs information not on that system or needs to interact with an agent on another system, it typically uses a communication mechanism, such as remote procedure calling.

In contrast, a mobile agent is not bound to the system on which it begins execution [1]. It is free to travel among the hosts in the network. Created in one execution environment, it can transport its state and code with it to another execution environment in the network, where it resumes execution. The term “state” typically means the attribute values of the agent that help it determine what to do when it resumes execution at its destination. Code in an object-oriented context means the class code necessary for an agent to execute.

A mobile agent has the unique ability to transport itself from one system in a network to another in the same network. This ability allows it to move to a system containing an object with which it wants to interact and then to take advantage of being in the same host or network as the object.

Our interest in mobile agents is not motivated by the technology per se but rather by the benefits agents provide for creating distributed systems. There are at least seven main benefits, or good reasons, to start using mobile agents:

They reduce the network load. Distributed systems often rely on communication protocols involving multiple interactions to accomplish a given task. The result is a lot of network traffic. Mobile agents allow users to package a conversation and dispatch it to a destination host where interactions take place locally. Mobile agents are also useful when reducing the flow of raw data in the network. When very large volumes of data are stored at remote hosts, that data should be processed in its locality rather than transferred over the network. The motto for agent-based data processing is simple: Move the computation to the data rather

than the data to the computation.

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

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

They execute asynchronously and autonomously. Mobile devices often rely on expensive or fragile network connections. Tasks requiring a continuously open connection between a mobile device and a fixed network are probably not economically or technically feasible. To solve this problem, tasks can be embedded into mobile agents, which can then be dispatched into the network. After being dispatched, the agents become independent of the process that created them and can operate asynchronously and autonomously. The mobile device can reconnect at a later time to collect the agent.

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

They are naturally heterogeneous. Network computing is fundamentally heterogeneous, often from both hardware and software perspectives. Because mobile agents are generally computer- and transport-layer-independent (dependent on only their execution environments), they provide optimal conditions for seamless system integration.

They are robust and fault-tolerant. Mobile agents' ability to react dynamically to unfavorable situations and events makes it easier to build robust and fault-tolerant distributed systems. If a host is being shut down, all agents executing on that machine are warned and given time to dispatch and continue their operation on another host in the network.

But be warned: Do not waste your time searching for the killer application for mobile agents. There are no mobile agent applications, but there are plenty of applications that benefit from using mobile agents. Several applications clearly benefit from the mobile agent paradigm:

E-commerce. Mobile agents are well suited for e-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 have different goals and implement and exercise different strategies to accomplish them. We envision agents embodying the intentions of their creators, acting and negotiating on their behalf. Mobile agent technology is a very appealing solution for this kind of problem.

Personal assistance. Mobile agents' ability to execute on remote hosts makes them suitable as assistants performing tasks in the network on behalf of their creators. Remote assistants operate independently of their limited network connectivity; their creators can even turn off their computers. For example, to schedule a meeting with several other people, a user can send a mobile agent to interact with the agents representing each of the people invited to the meeting. The agents negotiate and establish a meeting time.

Secure brokering. An interesting application of mobile agents is in collaborations in which not all the collaborators are trusted. The parties could let their mobile agents meet on a mutually agreed secure host where collaboration takes place without risk of the host taking the side of one of the visiting agents.

Distributed information retrieval. Instead of moving large amounts of data to the search engine so it can create search indexes, agent creators can dispatch their agents to remote information sources where they locally create search indexes that can later be shipped back to the system of origin. Mobile agents can also perform extended searches that are not constrained by the hours during which a 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 function as the glue keeping the systems flexible yet effective.

Workflow applications and groupware. The nature of workflow applications includes support for the flow of information among coworkers. Mobile agents are especially useful here, because, in addition to mobility, they provide a degree of autonomy to the workflow item. Individual workflow items fully embody the information and behavior they need to move through the organization— independent of any particular application.

Monitoring and notification. This classic mobile agent application highlights the asynchronous nature of these agents. An agent can monitor a given information source without being dependent on the system from which it originates. Agents can be dispatched to wait for certain kinds of information to become available. It is often important that the life spans of monitoring agents exceed or be independent of the computing processes that created them.

Information dissemination. Mobile agents embody the so-called Internet push model. Agents can disseminate information, such as news and automatic software updates, for vendors. The agents bring the new software components, as well as installation procedures, directly to customers' computers where they autonomously update and manage the software.

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

REFERENCE

1. Lange, D., and Oshima, M. *Programming and Deploying Java Mobile Agents with Aglets*. Addison-Wesley Longman, Reading, Mass., 1998.

DANNY B. LANGE (danny@genmagic.com) is director of agent technology in General Magic, Inc. in Sunnyvale, Calif.

MITSURU OSHIMA (oshima@genmagic.com) is a technical lead in General Magic, Inc. in Sunnyvale, Calif.

© 1999 ACM 0002-0782/99/0300 \$5.00