A Lightweight Mobile Agent Framework for Distributed Resource Discovery in LAN/WAN Networks

Srikrishna Sadula
Department of Computer Science
National Institute of Technology Calicut
Email: m060035cs@nitc.ac.in

Project Guide: Saidalavi Kalady
Department of Computer Science
National Institute of Technology Calicut

***Abstract*—Mobile agent technology gained significant attention in the early 2000s due to its potential to enable distributed, autonomous, and flexible computing across heterogeneous networked systems. This paper presents an academic prototype that demonstrates distributed resource discovery using a mobile agent capable of migrating across hosts in LAN/WAN environments. The agent performs localized checks on shared resources such as printers and server machines, aggregates results, and returns them to the dispatcher node. The work draws conceptual inspiration from IEEE FIPA-compliant platforms such as Mobile-C and related systems including Aglets and D'Agents, emphasizing code mobility, task segmentation, and platform-independent execution. This paper provides an architectural study, academic motivation, system design, workflow modeling, and evaluation of the prototype within the broader context of mobile agent research.**

I. Introduction

Distributed systems in the early 21st century began shifting from static client-server models toward more adaptive, mobile, and autonomous paradigms. Mobile agents—software programs capable of migrating across network nodes and executing tasks locally—emerged as an attractive solution for distributed resource management, network intelligence, and remote execution. Foundational systems such as Mole, Aglets, Concordia, D'Agents, Ara, and later Mobile-C demonstrated how mobility, lightweight execution, and message-based coordination could simplify distributed tasks and reduce network overhead.

Mobile agents support several advantages: (1) reduced network traffic by moving computation to data sources, (2) asynchronous and autonomous task execution, (3) adaptability to network conditions, (4) encapsulation of logic for mobility, resource discovery, and monitoring.

Motivated by these research directions, a simple academic prototype was developed in 2006 to study mobile agent behaviors for distributed resource discovery in LAN/WAN environments. This paper presents a structured academic treatment of the system, including design choices, architectural modeling, workflow representation, and alignment with the conceptual foundations of mobile agent research.

II. Related Work

Mobile agent systems have been explored extensively across the 1990–2006 period. Some of the notable platforms include:

A. Aglets (IBM)

Aglets introduced weak mobility using Java object serialization. Agents traveled across Aglet servers and performed tasks locally. Aglets helped popularize mobile agent programming and provided a sandboxed Java-based execution model.

B. D'Agents

D'Agents introduced strong mobility for Tcl and Java, supporting full continuation capture. This enabled seamless migration of execution state, providing a more powerful mobility model.

C. Ara

Ara provided multi-language support (Tcl, Java, C/C++) and strong migration using MACE. Its emphasis on heterogeneous support made it unique among contemporary systems.

D. Mobile-C

Mobile-C introduced a FIPA-compliant C/C++ mobile agent platform emphasizing portability, XML-based ACL message encoding, an embeddable interpreter, and integration with low-level hardware environments. It heavily influenced the conceptual framework of this academic prototype.

E. FIPA Standards

IEEE FIPA defined standardized ACL messaging, agent lifecycle structures, directory services, and platform interoperability, guiding the architectural organization followed in modern agent systems.

III. System Architecture

The academic prototype implements a minimal yet functional model inspired by the principles observed in Mobile-C and similar systems. The aim was not to replicate full FIPA compliance but to understand the mechanics of mobility, task execution, and host-level interactions through a simple demonstrator.

A. Design Goals

- Enable autonomous resource discovery across distributed nodes.
- Support task-based mobility with sequential host traversal.
- Minimize network bandwidth usage by executing checks locally.
- Provide simple message-driven migration without heavy frameworks.

B. Components

The system consists of:

1) **Agent Dispatcher**: Initializes the agent and defines the host traversal path.
2) **Mobile Agent Module**: Contains logic for printer/server availability checks.
3) **Host Execution Environment**: Receives agents, executes tasks, and forwards results.
4) **Result Aggregator**: Collects results returned by the mobile agent.

IV. Mobile Agent Workflow

The workflow of the agent incorporates dispatch, migration, local execution, and return. A diagram illustrating the core workflow is shown below.

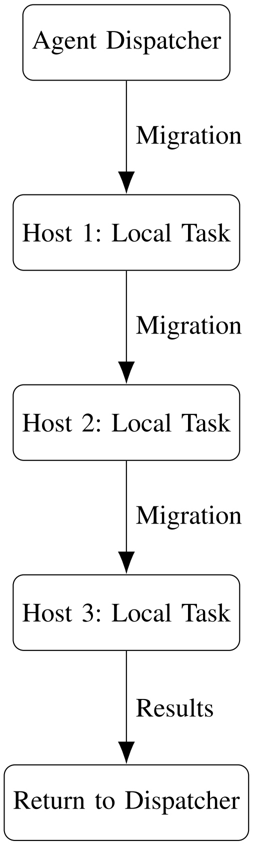

Fig. 1. Mobile Agent Workflow for Distributed Resource Discovery

V. Agent Mobility Model

The prototype adopts a weak mobility approach similar to that seen in Aglets and Mobile-C. Execution state is not transferred; instead, tasks are broken into discrete subtasks executed independently at each host. Mobility occurs only between completed tasks.

A. Task List-Based Execution

Each mobile agent maintains:

- A list of hosts to visit.
- A set of tasks corresponding to resource checks.
- A results buffer for aggregating local execution outputs.

A task pointer determines the current subtask. After execution, the agent serializes its data state and transmits itself to the next host.

VI. Evaluation

The academic prototype demonstrated the following:

A. Functional Validation

- Successful agent migration across LAN/WAN nodes.
- Correct detection of printer and server availability.
- Proper aggregation of distributed results.

B. Observed Advantages

- Significant bandwidth savings by executing checks locally.
- Fully autonomous execution after initial dispatch.
- Minimal load on central dispatcher.

VII. Conclusion

The mobile agent prototype providing an hands-on exploration of mobility concepts and distributed resource discovery. Its architecture reflects the influence of foundational mobile agent systems while offering a simplified academic model suitable for experimentation. The study helped understand the principles of agent migration, distributed task execution, and autonomous network traversal.

References

[1] D. Lange and M. Oshima, *Programming and Deploying Java Mobile Agents With Aglets*. Addison-Wesley, 1998.

[2] R. Gray et al., "D'Agents: Applications and performance of a mobile-agent system," *Software—Practice and Experience*, 2002.

[3] H. Peine, "Run-time support for mobile code," PhD Thesis, University of Kaiserslautern, 2002.

[4] B. Chen, H. H. Cheng, and J. Palen, "Mobile-C: a mobile agent platform for mobile C/C++ agents," *Software—Practice and Experience*, 2006.

[5] IEEE FIPA Standards. Available: http://www.fipa.org