A Logic-based Framework for Mobile Intelligent Information Agents

Naoki Fukuta, Takayuki Ito, and Toramatsu Shintani
Nagoya Institute of Technology
Dept. Intelligence and Computer Science
Gokiso-cho, Showa-ku, Nagoya, 466-8555, JAPAN
{fukuta,itota,tora}@ics.nitech.ac.jp

ABSTRACT

Information agents are computational software systems that provides transparent access to many different information sources in the Internet, and to be able to retrieve, analyze, manipulate, and integrate heterogeneous data and information on demand. Mobile agents are an effective choice for many applications to improve latency and bandwidth of client-server applications and to reduce vulnerability to network disconnection. In this paper, we propose MiLog, a logic-based mobile agent framework for constructing intelligent information agents. In the framework, inference and planning process of an agent is written in a compact logic program. A new generation logic program execution engine provides strong migration capability with reasonable execution speed. The framework has been used to develop a web-based decision support system 'BiddingBot'.

Keywords

Information Agents, Mobile Agents, Logic Programming

1. INTRODUCTION

Information agents have been investigated very actively. Information agents are computational software systems that provides transparent access to many different information sources in the Internet, and to be able to retrieve, analyze, manipulate, and integrate heterogeneous data and information on demand. Mobile agents are an effective choice for many applications to improve latency and bandwidth of client-server applications and to reduce vulnerability to network disconnection[1]. However, it needs more expressive power of its programming language to realize complex behaviors of the agents. In this paper, we propose MiLog[2], a logic-based mobile agent framework for constructing intelligent information agents. In the framework, inference and planning process of an agent is written in a compact logic program. The framework contains a new generation logic program execution engine which provides a strong migration capability with reasonable execution speed. The framework helps developers to construct more responsible and flexible information gathering systems.

The rest of this paper is organized as follows: In section 2 we presents the architecture of mobile agents on the MiLog framework. In section 3 we discuss the features of the MiLog by presenting a real world application which supports online auctions on the Internet. Section 4 summarizes this paper, and show some possible directions for future researches.

2. MOBILITY

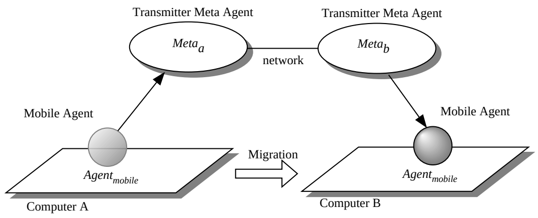

Figure 1: The implementation model of the mobile agents.

Figure 1 shows the architecture for implementing mobile agents. Each computer has one meta agent which is a transmitter. A mobile agent migrates from computer 'A' to computer 'B' as following steps :

(STEP 1) The mobile agent $Agent_{mobile}$ requests the meta agent $Meta_A$ on the computer 'A' to migrate to the computer 'B'. The agent $Meta_A$ interrupts the execution of the mobile agent $Agent_{mobile}$.

(STEP 2) The meta agent $Meta_A$ captures the state of the mobile agent $Agent_{mobile}$ by using special built-in predicates for controlling other agents.

(STEP 3) The meta agent $Meta_A$ negotiates with the meta agent $Meta_B$ on the computer 'B'. If the negotiation is successful, the mobile agent $Agent_{mobile}$ is transported to the meta agent $Meta_B$ on the computer 'B' by using a transport method (e.g. HTTP or SMTP). If the negotiation or the transportation fails, the mobile agent $Agent_{mobile}$ is resumed its execution on the computer 'A'.

(STEP 4) The meta agent $Meta_B$ on the computer 'B' receives the mobile agent $Agent_{mobile}$, checks the agent program, and then resumes the mobile agent $Agent_{mobile}$ on the computer 'B'. The mobile agent $Agent_{mobile}$ continues its execution on the computer 'B'.

Figure 2 shows a sample code of a simple meta agent. On the STEP1, the mobile agent requests the predicate `move/2` to the meta agent. For example, if mobile agent named `mover` intends to move the host `123.45.67.8:9012` (this means that IP address is 123.45.67.8 and port number is 9012), the following request is established.

```
move( mover, '123.45.67.8:9012' )
```

The meta agent captures the status of the agent (Figure 2 line:2), sends them to an appropriate meta agent on the target host(Figure 2 line:3), and then delete the agent(Figure 2 line:4). When the transportation fails, the agent is not migrated and continues its execution on the local host (Figure 2 line:6–7). A meta agent on the target host receives the request to resume the mobile agent. For instance, the following predicate is called on the meta agent (the string '....' means the encoded binary data of the mobile agent.).

```
receiveAgent( mover, '....' )
```

The meta agent decodes and verifies the status of the mobile agent, and then resume its execution on the target host (Figure 2 line:9–11).

```
1  move( Agent, Host ) :-
2    serialize( Agent, Bin ),
3    transmitterMetaAgentName(T),
4    request( T, receiveAgent( Agent, Bin ), Host),
5    deleteAgent(Agent),!.
6  move( Agent ,_ ) :-
7    resume( Agent ).
8  receiveAgent( Agent, Bin ) :-
9    deserialize( Agent, Bin ),
10   verify( Agent ),
11   resume( Agent ).
```

Figure 2: An example of transmitter meta agent

3. *BIDDINGBOT* : AN APPLICATION

We have proposed *BiddingBot*[3], a system which can support bidding to several auction sites simultaneously by the cooperative bidding agents[4]. In this section, we show how *BiddingBot* system is implemented by using the MiLog framework.

Figure 3 shows the architecture (on the left of the Figure 3) and the user interface (on the right of the Figure 3) of *BiddingBot*. *BiddingBot* consists of one *leader agent* and several *bidder agents*. Each bidder agent is assigned to an auction site. Each agent is implemented by using the MiLog framework. The bidder agents cooperatively gather information, monitor, and bid in the multiple auction sites simultaneously. The communications among agents are implemented by using shared clause databases and querying to other agents. The leader agent facilitates cooperation among the bidder agents as a match-maker, sends a user's request to the bidder agents, and presents bidding information to the user. The leader agent interacts with a user as a WWW server on the MiLog framework. The cooperation protocol are implemented by finite state machines in which the state is changed by receiving a certain query from other agents. In *BiddingBot*, each of the bidder agents is specified to her auction site and can behave as a flexible wrapper, since different auction sites represent information in different forms. Wrapper functions on the bidder agents are implemented by using a pattern description language in the MiLog.

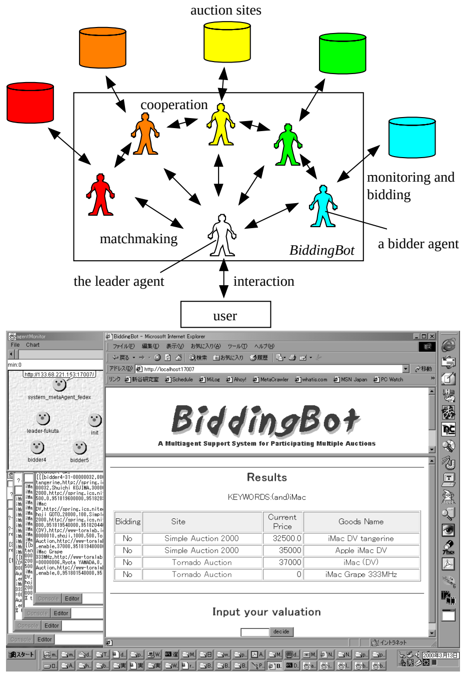

Figure 3: The BiddingBot System

4. CONCLUSION

In this paper, we proposed MiLog, a logic-based framework for constructing mobile intelligent information agents. Since the MiLog supports multi-thread programming and messaging mechanisms among agents, we can effectively implement cooperation mechanisms among agents. We demonstrated a simple meta agent program which can be extended by applying models for security, authorization, and transportation management mechanisms for realizing efficient migrations of agents. We have implemented *BiddingBot*, an online auction support system by using the MiLog framework. We demonstrate that our approach is useful to incorporate intelligent behaviors into mobile agents.

5. REFERENCES

[1] B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko and D. Rus: "Mobile Agents for Distributed Information Retrieval", M. Klusch(Ed.) Intelligent Infromation Agents –Agent-Based Information Discovery and Management on the Internet– , Springer, pp.355–395, 1999.

[2] N. Fukuta, T. Ito, and T. Shintani: "MiLog: A Mobile Agent Framework for Implementing Intelligent Information Agents with Logic Programming," *Proc. of the First Pacific Rim International Workshop on Intelligent Information Agents(PRIIA'2000)*, pp.113-123, 2000.

[3] T. Ito, N. Fukuta, T. Shintani, and K. Sycara, "Multiagent Cooperative Bidding Support for Electronic Auctions," *Proc. of ICMAS2000*, pp.435-436, 2000.

[4] T.Ito, N. Fukuta, R. Yamada, T. Shintani, and K. Sycara,"Cooperative Bidding Mechanisms among Agents in Multiple Online Auctions," *Proc. of PRICAI2000*, pp.810, 2000.