STUDY ON MODELING MOBILE OBJECTS IN DISTRIBUTED COMPUTING ENVIRONMENT

Rong XIE*
Prof. Ryosuke SHIBASAKI*

*Center for Spatial Information Science,
University of Tokyo
4-6-1 Komaba, Meguro-ku, Tokyo, 153-8508
Tel: (81)-3-5452-6417 Fax: (81)-3-5452-6414
E-mail: xierong@iis.u-tokyo.ac.jp, shiba@iis.u-tokyo.ac.jp
JAPAN

KEY WORDS: mobile objects, distributed computing, mobile agent, modeling and simulation

ABSTRACT:
Currently, large quantities of position-aware mobile objects, such as GPS, mobile phone, PDA, various smart vehicles are merging in distributed environment and systems. The real world applications such as Location-Based Services etc. are being developed in an ad hoc fashion. As a result, software technologies that enable the management of positions of distributed mobile objects capable of very large numbers and very frequent changes are in increasingly high demand. We introduce to mobile agent technology for managing mobile objects in a distributed computing environment. The paper firstly presents a framework of management of distributed mobile objects. A Hierarchical Location Schema for the indexing of historic and current positions of distributed mobile objects among nodes is put forward to locate node quickly. Time-stamped E-R model is put forward as the basis of data handling to support the schema. Also, mobile agent computing model is proposed for real-time update, load balancing and robust against network troubles. In the model, techniques of naming and addressing are applied and designed. For the purpose of implementation of distributed query processing, two kinds of mobile agents: collaborative agent and information agent are implemented. Taking the examples of real-time tracking simulation and distributed query processing, the proposed method and model are finally applied to demonstrate its effectiveness and robust for the design and development of distributed system.

1. INTRODUCTION

Currently, large quantities of position-aware mobile objects, such as GPS, mobile phone, PDA, various smart vehicles etc. are merging in distributed environment and systems. The real world applications for mobile objects such as Location-based Services, dynamic micro simulation in traffic situations are being developed in an ad hoc fashion. As a result, software technologies that enable the management of positions of distributed mobile objects capable of very large numbers and very frequent changes are in increasingly high demand. In our research, we introduce to mobile agent technology for managing mobile objects in a distributed computing environment.

2. MANAGEMENT OF DISTRIBUTED MOBILE OBJECTS

2.1 Entity Types of Mobile Object

In the paper, we define *mobile objects* as real world entities moving over space, whose location changes with time. The term of mobile objects is more general, having two meanings: *moving objects* & *mobile agent*.

- Objects, such as vehicles, pedestrian, cellular phone user, change their locations continuously, we use the term *moving objects* to refer to such entities.
- Mobility tied to network hardware, data and code on moving objects may be relocated among different network nodes by *mobile agent*.

Besides, users with mobile terminals move in the coverage area of wireless networks, called *mobile users*, use the moving objects data.

2.2 Introduction to Mobile Agent Technology

We meet very large number of distributed mobile objects and very frequently changes in position of mobile objects. In order to support semi-"real time" service and computation, "current" information should be saved; on the other hand, in order to process various queries, "history" record is also necessary to be kept. Therefore, access to data in geographical close areas will be more frequently, and data are originally distributed over many nodes in the network. We suggest to managing mobile objects in a distributed computing environment.

The motivations for distributed computing applications are mainly sharing of resources and information over a computer network in a transparent way, and cooperative computing. Object-oriented distribution schemes are widely applied in current research, such as C/S, Applet, and Servlet etc.; however, a more powerful design paradigm for network computing is mobile agent technology, a natural successor to the object-oriented paradigm. We are interested in it by the benefits of its providing for the creation of distributed systems. Danny Lange [4] listed seven good reasons for mobile agent technology. Some advantages of mobile agent related to our research are:

(1) To reduce the network load. It has the unique ability to transport itself from one system in a network to another. Very large volumes of data could be processed in the local system rather than transferred over the network. It also offers a solution to critical real-time systems.
(2) To provide load balancing. Multi agents are working together towards a common goal and coordinate closely with each other. A specific larger task is easy and efficient to handle and accomplish by the cooperation of multiple agents, without resorting to larger computers.
(3) To be robust and fault-tolerant. The ability of mobile agent to react dynamically to unfavorable situations and events makes it easier to build robust and fault-tolerant distributed systems.

2.3 Distributed Computing System Framework

We would like to put forward a framework based on *Hierarchical Location Schema* (*HLS*) as shown in Figure 1. The location management is based on a hierarchical organization of the network into *domains*. *Domains* divide the network into geographical, administrative or network-topological coverage.

Within the distributed location management, a *domain* is represented by *node*. Moving objects can be a kind of real-world agent, migrating among nodes in the network. *Leaf node*, modeled by time-stamped E-R model, is responsible for keeping track of all moving objects located in its own domain. *Higher-level nodes* contain location information about moving objects located at the levels of its sub-tree. When user submits query request from him/her nearest node, inside this node, the agent manager handles tasks locally or dispatches several mobile agents to the remote nodes to search for information. After agents reach the destination, distributed or parallel processing will be handled at local node. After tasks are finished, agents will return back to the original node, the result will present to the user. The task of agent is supported by distributed query processing.

Error! No topic specified.

Figure 1 Distributed Computing Model for Mobile Objects Management

Such framework has several potential strengths, to make "real time" update, query and other services feasible, to be robust against node and network troubles, and to make load balancing efficiently.

3. DISTRIBUTED COMPUTING MODEL

3.1 Time-stamped ER model

Time-stamped model recording "history" based on the event, represents the relationship among object, event and node related in application, and time-stamped attributes record the evolution of their values. All history or event data are stored in each leaf node.

Error! No topic specified.

Figure 2 Time-stamped E-R Model

Object *table* <*object_id*, *object_name*, $characteristic_1$, $characteristic_2$, …, $characteristic_n$, *creation_time*>
Event *table* <*historical_age*, *object_id*, *x*, *y*, *z*>

***Node** table* <node_id, node_name, serviceCenter_x, serviceCenter_y, service_radius>

3.2 Mobile Agent Computing Model

Mobile agent computing model consist of two kinds of agent: agent manager and several slave agents.
(1) Agent Manager
Agent manager provides an execution environment for other slave agents. It provides the following services:

- Creating, dispatching of slave agent or event-driven agent
- Transaction between agents, data
- Controlled access to local resources

Inside agent manager architecture, task module (including naming, addressing and query processing), communication module, DB, status sets are included, which are established on mobile agent platform. By the communication module, agents communicate with each other and environment. Tasks will be solved by task module according to database, status sets and some intermediate results. If the local module cannot solve the task, agent manager dispatches some mobile slave agents to migrate to other platform to search for computing resources.

Error! No topic specified.

Figure 3 Mobile Agent Architecture

A global naming scheme and name service is needed for locating resources, specifying agent for agent migration, and establishing inter-agent communication. A naming scheme defines a name-space of contact address to identify each distributed shared agent. Typically, it defines each agent as a pathname to an *object handle*, consisting of logically *local ID* and additional information for the physically location service. A *local ID* is a global unique and is never reused after the agent is destroyed. The additional location information is allocated by the name service after agent is registered its new contact address. When mobile agent migrates among network, addressing is applied to locate node position of agent. For complex systems, each mobile agent should register its name and current location to the *Name Server* for each time it is created or migrated from one node to another.

Error! No topic specified.

Figure 4 Addressing

(2) Slave Agent
When a mobile agent is preparing for a trip, it must be able to identify its destination. Once the location of the destination is established, the mobile agent informs the local system that it wants to transfer itself to the destination system. When the system receives the request of trip, it will do suspending the execution of agent, keeping current data and status, defining the new destination node and then dispatching agent with its data and status. When agent arrives at the destination, the data and status will be recovered to resume execution in new node.

Error! No topic specified.

```
1: public class AgentMigration extends Aglet {
2:       String coor_x, coor_y;
3:       public void onCreation(Object args) {
4:             if (args!=null){
5:                coor_x=(String)((Object[])args)[0];
6:                coor_y=(String)((Object[])args)[1];
7:             }
8:             addMobilityListener(new MobilityAdapter() {
9:                   public void onArrival(MobilityEvent ev) {
10:                        doTask();
11:                  }
12:            });
13:      }
14:      public void dispatchSlave(String dest, int x, int y) {
15:            Object args[]=new Object[]{String.valueOf(x), String.valueOf(y)};
16:            try {
17:                  AgletContext context = getAgletContext();
18:                  NodeDetermine();    // which node to dispatch
19:                  AgletProxy proxy = context.createAglet(getCodeBase(), "Slave", args);
20:                   URL url = new URL(dest);
21:                  proxy.dispatch(url);
22:            } catch (Exception ex) { ex.printStackTrace(); }
23:      }
24:} // end of class AgentMigration
```

Figure 6 The Agent Migration Class

3.3 Distributed Querying Processing

Location queries in distributed systems are
assigns the task to mobile agent(s), which
the task. For the implementation of distribu
follows.

(1) Information agent

Information agent migrates between distributed information sources to perform the tasks of managing, manipulating or collecting information from many distributed sources.

Error! No topic specified.
Figure 7 Information Agent Pattern

```
1: public class InformationAgent extends Aglet {
2:     boolean _the Remote=false;
3:    public void onCreation (Object init) {
4:        addMobilityListener (new MobilityListener () {
5:              public void onDispatching (MobilityEvent e) {
6:                  // Print to the console…
7:              }
8:              public void onArrival (MobilityEvent e) {
9:                  _theRemote=true;
10:                 // Print to the console…
11:             }}
12:       );
13:   }
14:   public void run (){
15:       if (!_theRemote) {
16:          // The original aglet runs here
17:          try {
18:              URL destination=new URL((String)getAgletContext().getProperty("location");
19:              dispatch(destination);
20:          } catch (Exception e) { System.out.println (e.getMessage()); }
21:       } else {
22:          // The remote aalet runs here…
23:       }
24:   }
25: }
```

(2) Collaborative agent
Many heavy tasks may not be accomp
balancing, mobility and messaging pla
the participants in the Master-Slave pa

- A master agent creates a slav
- The slave agent moves to a re
- The slave agent returns with t

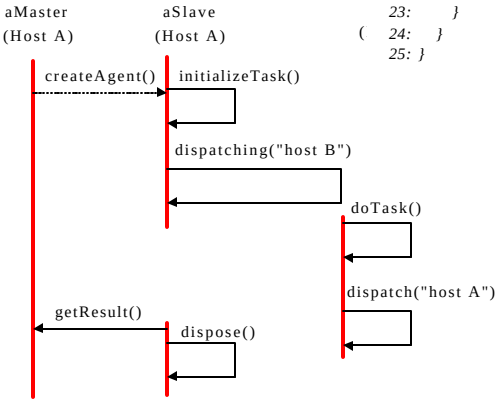


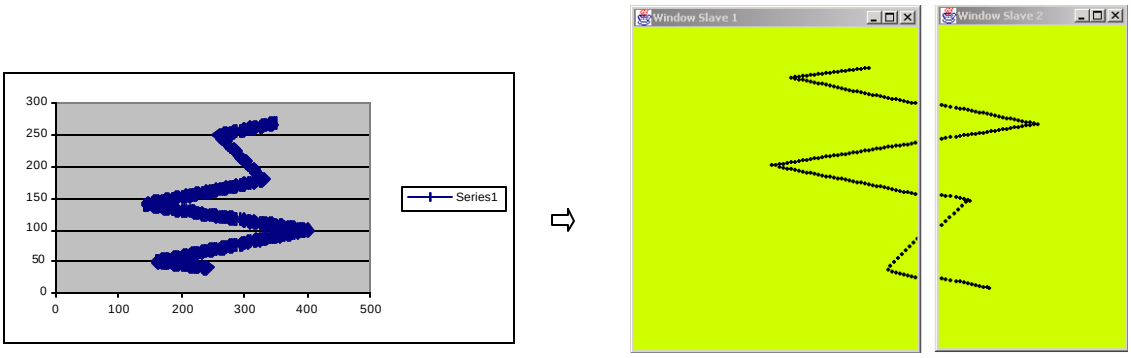
Figure 9 Collaborative Agent Pattern

```
1:  public abstract class Master {
2:       AgletProxy proxy=getAgletContext().createAglet(
                                    getCodeBase (),"Slave", null);
3:       proxy=proxy.dispatch (destination);
4: } // end of class Master
5:
6: public abstract class Slave {
7:       object result=null;
8:       public void onCreation (Object args) {
9:            // called when the slave is created.
10:           // gets the remote destination
11:      }
12:      public void run (){
13:           // at the origin;
14:           initializeTask();
15:           dispatch (destination);    // goes to destination
16:           // at the remote destination
17:           doTask ();  // Stats on the task
18:           result=…;
19:           // returns to the origin
20:           // back at the origin
21:           // delivers the result to the master and dies
22:           dispose ();
23:      }
24: } // end of class Slave
```

Figure 10 The Collaborative Agent Class

4. Simulation Results

(1) Real-time Tracking Simulation Among Nodes
Moving object can travel in its existed environment. It is autonomous to travel from one node to another as a kind of mobile agent. Once created in one execution environment, the agent transports with its state, data to another execution environment in the network, where it resumes execution. To tracking the whole trajectory of moving object, (1) calculating coordinate boundary; (2) if exceeding boundary, determining the new location of destination node; (3) dispatching itself to the new node; (4) when arriving at the new node, resume its transport. Figure 10 shows trajectory simulation of tracking a moving object transporting between two nodes.

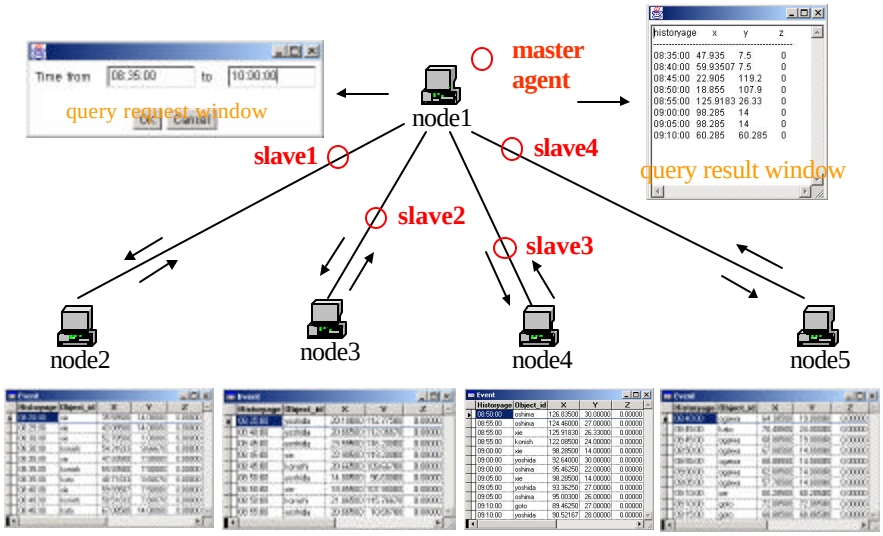


(a) Tracking Data (b) Tracking Simulation on two nodes

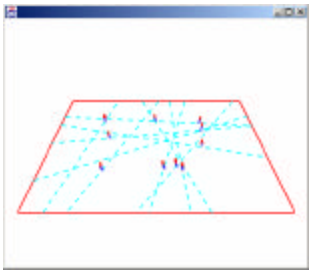
Figure 11 Example – Real-time Tracking Simulation Among Nodes

(2) Distributed Query Processing

Considering a database representing information about moving objects and their position. (1) Creating several collaboration agents to dispatch to the remote nodes; (2) handling query operation at local node after agent arrives at the local node; (3) After agents come back, results are collected and simulated. Figure 11 shows simulation by the querying result of trajectory information of moving objects.



(a) Distributed Query Processing



(b) Tracking Simulation Result on Node 1

Figure 12 Example – Distributed Query Processing

5. CONCLUSIONS AND FUTURE WORK

In this paper, we discuss managing mobile objects in a distributed computing environment and the application of mobile agent technology to mobile objects simulation. Our preliminary work shows that

the integration of Hierarchical Location Schema, time-stamped ER model, mobile agent computing model in distributed data management is feasible. Agent-based system is inherently distributed, and agent technology can reduce network load, provide load balancing and make distributed systems robust and fault-tolerant. So mobile-agent technology could be a powerful, uniform paradigm for the design and development of distributed systems.

Further research includes developing an event-based micro-simulation to represent behavior of mobile objects when mobile objects interact with each other and with environment. Agent-based behavior model will also facilitate the research with strategies of time synchronization, resource allocation and mutual exclusion by application of mobile agent technology.

REFERENCES

[1] Aline Baggio, Gerco Ballintijn etc., 2001. Efficient Tracking of Mobile Objects in Globe. The Computer Journal, 44(5), pp340-353

[2] Evaggelia Pitoura, George Samaras, 2001. Locating Objects in Mobile Computing. IEEE Transactions on Knowledge and Data Engineering, 13(4), pp571-592

[3] S. Spaccapietra, C. Parent, E.Zimanyi, 1998. Modeling Time from a Conceptual Perspective, http://lbdwww.epfl.ch/e/publications/articles.pdf/CIKM98.pdf

[4] Danny B. Lange, Mitsuru Oshima, 1998. Programming and Deploying Java Mobile Agents with Aglets. Addison-Wesley, 2000

[5] Chong Xu, Dongbin Tao. Building Distributed Application with Aglet. http://www.cs.duke.edu/~chong/aglet

[6] Jim Farley. Java Distributed Computing. O'Reilly & Associates Inc., 1998