Information Sharing Mechanism among Mobile Agents In Ad-hoc Network Environment and Its Applications

Kunio Umetsuji and Kyoji Kawagoe
Computer Science Department
Ritsumeikan University
1-1-1, Noji-Higashi,
Kusatsu-city, Shiga
JAPAN

ABSTRACT

Mobile agents are programs that can move from one site to another in a network with their data and states. Mobile agents are expected to be an essential tool in pervasive computing. In multi platform environment, it is important to communicate with mobile agents only using their universal or logical name not using their physical locations. More, in an ad-hoc network environment, an agent can migrate autonomously and communicate with other agents on demand. It is difficult that mobile agent grasps the position information on other agents correctly each other, because mobile agent processes a task while moving a network successively. In order to realize on-demand mutual communication among mobile agents without any centralized servers, we propose a new information sharing mechanism within mobile agents.

In this paper, we present a new information sharing mechanism within mobile agents. The method is a complete peer based and requires no agent servers to manage mobile agent locations. Therefore, a mobile agent can get another mobile agent, communicate with it and shares information stored in the agent without any knowledge of the location of the target mobile agent. The basic idea of the mechanism is an introduction of Agent Ring, Agent Chain and Shadow Agent. With this mechanism, each agent can communicate with other agents in a server-less environment, which is suitable for ad-hoc agent network and an agent system can manage agents search and communications efficiently.

Keywords: Mobile Agent, Distributed system, Information Sharing, Communications and Web information change detection

1. INTRODUCTION

Mobile agents are programs that can move from one site to another in a network with their data and states. Mobile agents are expected to be an essential tool in pervasive computing [1]. Mobile agent platforms such as Aglet [7], Objectspace [8], and Mobidget [9] have been developed and released over the last few years [2]. However, in multi platform environment, it is important to communicate with mobile agents only using their universal or logical name not using their physical locations. Each mobile agent can migrate around autonomously. In conventional multiple agent platforms, it is necessary to construct the single integrated agent management system in a widely distributed agent network system in order to keep a mapping of physical agent locations and logical agent names. However, in an ad-hoc network environment, an agent can migrate autonomously and communicate with other agents on demand. Moreover, it is difficult that mobile agent grasps the position information on other agents correctly each other, because mobile agent processes a task while moving a network successively. So information sharing will not be able to be done and information search efficiency will be dropped if an exact position is not known mutually. Therefore, an efficient mechanism of information sharing among mobile agents is necessary to be developed. In the paper, we will present a new information sharing mechanism among mobile agents.

In the section 2, we describe mobile agent technology. Then, the section 3 shows our proposed method. The section 4 contains evaluation and applications. Finally, the section 5 is the conclusion of the paper.

2.MOBILE AGENTS AND MUTUAL COMMUNICATIONS

Mobile agents

The mobile agent is a program that contains such characteristics as autonomy, cooperativeness, adaptability, movement ability and locality. The main features of mobile agent actually move in a network, and process a given task. Each mobile agent includes the mobile agent's own program code, its own processing result and the internal state of the mobile agent. The mobile agent can process its task by autonomously migrating to an appropriate location, even if the mobile agent cannot get communications with its user as well as other agents,

It is called a multi-agent system that is composed of a set of mobile agents which perform the common task in a cooperative way with the agents. In order to realize this multi-agent system, it is important to establish sharing of the information that an agent has, among the underlying multi-agents. Moreover, mobile agents need to search for other agents to perform a cooperative task.

Information sharing among mobile agents

Sharing of the information between mobile agents is called "cooperativeness" which is one of the features of the agent technology.

Suppose that there exist several agents for some given common task. When these mobile agents work independently, there is a case such as some mobile agents move to the common location at a different time, working the same task repeatedly. In this case, when the agents can get communication with each other,

exchanging some information on the underlying task, the repeated migration will be avoided and the task will be processed more rapidly. Therefore, it is important to introduce some information sharing among mobile agents.

Mutual communications

There have been some models of information sharing among mobile agents, some of which include Agent directory server model and Hierarchical search model, described before.

Agent directory server model

Agent directory server model uses an agent directory server, which manages a directory of each agent location and name and returns a physical address of an agent from the given agent name to a requested agent. The agent directory server also manages the information directories. So, the server can return a physical address of a data from the given data name to a requested agent. By this process, the agent requesting some information to the directory server and the agent that hold the information can start communicating with each other.

However, this model has an essential problem that takes much load for the server when a large amount of agents is accessed to the server in a widely distributed agent system. Moreover, when a server breaks down, all agents that managed by agent server may stop own functions. These problems are very important problems and should be solved. Figure 1 shows a structure of the model.

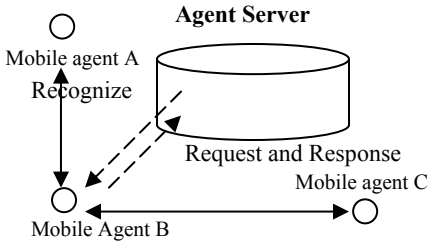

Figure 1 Directory server model

Hierarchical search model

There is no centre object in a hierarchical search model. An agent network based on the hierarchical search model depends on no server. In this model, an agent search is performed by traversing from a route agent toward leaf agents in a hierarchy of agents linked beforehand. After the target agent is found, the requested agent can communicate with the target agent. The outline of the model is described in Figure 2.

However, it is hard to maintain this agent hierarchy dynamically because each agent can autonomously migrate from one site to another and its upper agent has to manage the communication with the agent at any time.

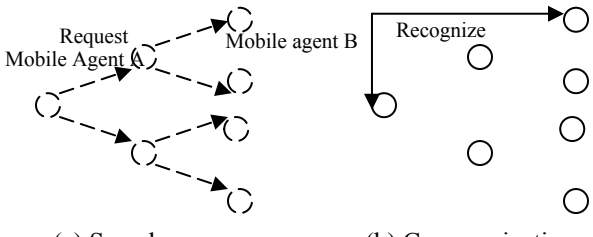

Figure 2 Hierarchical search model

3. PROPOSED METHOD

Basic concept

We propose a new mechanism of information sharing in order to achieve both server-less realization and efficient management of an agent system. The following two key ideas called Agent Ring and Agent Chain, which are important concepts regarding our proposed mechanism.

Agent ring

Agent ring is defined as a ring of mobile agents by each connecting with the next agent and with prior agent. As shown in Figure 3, each agent of a ring contains only two agent pointers pointing to the next and prior agent locations. Each time some agent in a ring changes the physical locations, the agent informs the new location to only the related two agents. When an agent intends to search agents in a ring, by requesting a search condition, the search request is sent to the adjacent agent and this adjacent agent sends the same request to the next agent in turn like going round the ring.

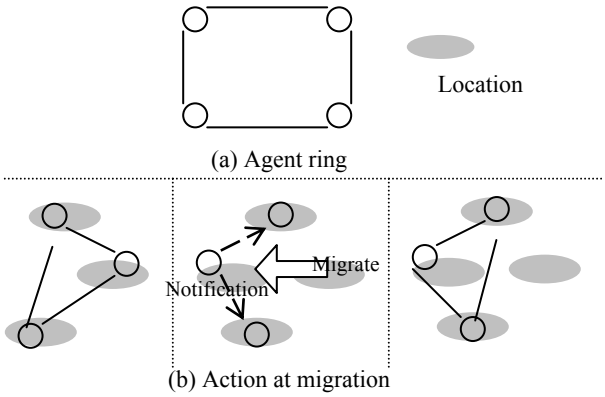

Figure 3 Agent ring and action at the time of migration of mobile agents

Agent chain

Agent chain is defined as a ring network by connecting several agent rings using gateway agents. When the number of mobile agents in a ring increases, the path length of the ring gets larger and the searching cost declines. To avoid this situation, we introduce the agent chain by dividing the agents into many small agent rings and by connecting each ring to other rings with agents called gateway agents. Some of the agents composing the ring can be selected as gateway agents as shown in Figure 4.

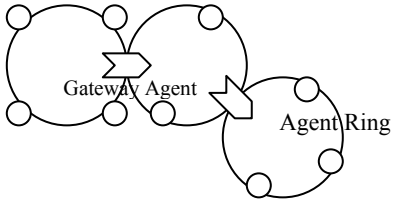

Figure 4 Agent Chain

Information sharing process

With the agent ring and agent chain concepts, agent search and agent communications are performed as follows: When some agent requests a search condition with which the agent needs to get communication with agents satisfied, the requested agent first sends this request to the next agent. The received agent checks the

requested condition and also sends the request to the next agent. By iterating this, the request is returned to the original agent within a ring containing the agent. If the received agents are satisfied with the requested condition, the agent sends its location to the original agent directly. After that these agents can get communications with each other and both agents can share the same information with this communications. When a gateway agent receives the request, the agent sends the request to the next agent in the same way as other agents and the gateway agent also sends the request to next agents within other rings linked to the current ring with this gateway agent. The Figure 5 shows a simple agent search algorithm.

```
SA: the starting agent
O: output
Hop: distance, from the ring of SA,
CA: agent to be searched
CH: hop limit as condition
X: agents checked
  Start:
  Set SA,CH,CA;
  Set Hop=0;
  Set O=NULL;
  Set X=null;
  search_agent(SA,O,CH,CA,X,Hop);
  End;

search_agent(A,O,CH,CA,X,Hop) {
  While (A is not in X or Hop is greater than CH){
       then
       {if check_agent(A,CA)=true
            then  {A is included in O}}
        A is included in X
        Set SA's next agent in the current ring as NA
search_agent(NA,O,CH,X,Hop)
        if A is a gateway agent
              then
           {Set the next agent in the next ring as NRA
            search_agent(NRA,O,CH,X,Hop+1)
           search_agent(NA,O,CH,X,Hop)}
      }
    return(true) }

check_agent(A,CA) {
    if A==CA  then return(true)
      else return(false) }
```

Figure 5 Basic Agent Search Algorithm in Agent Chains

Figure 6 shows how the searching is performed with our algorithm. In the figure 6, searching is executed in numerical order starting from 1. Once the desired agent is detected, the source agent can directly communicate with this target agent.

While the algorithm shown in Figure 5 is simple for convenience, the following improvement on the algorithm in Figure 5 has been made.

1) Location information of gateway agents is kept in cache managed by each agent. This information is generated when a gateway agent receives a request from a source agent and sends its location to the agent. The information is used when the source agent sends another request the next time, in order to reduce the number of traverse among agent chains.
2) Duplicate searching is avoided by using GUID (Globally Unique Identifier), which is generated for each agent request in order for each agent to be able to check whether a request is processed before or not. This idea is not unique but Gnutella and others have already used the GUID[4]

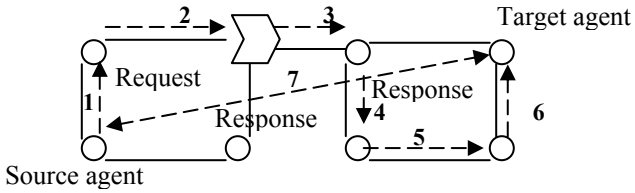

Figure 6 Information Sharing Process in our proposed method

Shadow agent

An agent ring does not function normally, when some agent in the agent ring, has some trouble and cannot work its operation. Therefore, we include fault tolerance feature in our proposed agent ring concept. Shadow Agent is introduced in order to add its fault tolerance feature. The shadow agent is defined as an agent used for an alternative agent in a time of some trouble in an agent ring. A shadow agent is created when the real agent is created. The same information is stored in the shadow agent and the real agent. When some trouble in an agent ring such as in the case that disconnection of the ring occurs, the shadow agent restores its real agent and the shadow agent becomes a bridge connecting two adjacent mobile agents. Even if an agent disappears by any occasion, the agent ring can perform its usual task with the help of the shadow agent and can recover its original structure.

Each shadow agent contains information of 1) location information of its real agent, 2) replica of link information of its real agent and 3) existence information of its real agent.

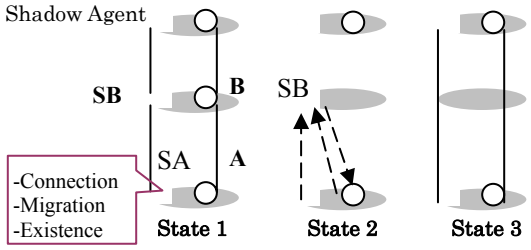

Figure 7 Shadow Agents and its Utilization

Some kind of trouble is caused in many cases by the state of each location. Therefore, if real agent and shadow agent exist in the same location, there is a possibility of disappearing simultaneously and agent ring may be unable to be restored. Therefore shadow agent can also avoid such crises by moving to various locations quite like its real agents. It is important to consider efficiency of agent ring recovery. And it is also necessary to transmit the migration information on a shadow agent to its real agent. Both shadow agent and real agent operations are described below.

Step 1) Generation of shadow agents
A shadow agent is created when a real agent is created. At the time, the shadow agent contains all the information of communication, migration and existence of its real agent. Such information as communication and migration is changed one by one depending on a subsequent operation for its real agent.

Step 2) Construction of the shadow agent ring composed of shadow agents.

The shadow agent SA of the real agent A has a relation with another shadow agent SB, which is the agent of another real agent B that has a relation with the real agent A. This is the case of State 1 in Figure 7.

Step 3) Recovery from some trouble
At a time of trouble occurring in some agent ring, the corresponding shadow agent ring composed of the corresponding shadow agents is used in order to recover the original agent ring. Some information of communication and migration on the real agent B that has caused the trouble can be acquired from its shadow agent SB, and the agent ring can then be restored. This is the case of State 2 in Figure.7. Even if, it is the case that some trouble occurs in some shadow agent, a real agent that manages the shadow agent can be utilized for restoring and generating shadow agent. The result is shown in the State 3 in Figure 7.

4. EVALUATION AND APPLICATIONS

We have made some experiments, in order to show the usefulness of our proposal. We are now under development of the application system of our proposed method.

Experiments

Performance evaluation

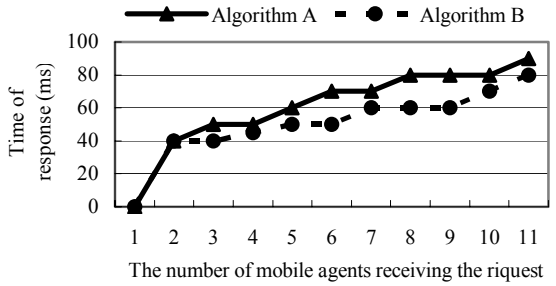

Figure 8 Performance Evaluation

We made some experiment under the following conditions:

- An agent ring consists of five agents. And an agent chain consists of two agent rings.
- Response time is calculated from the time when starting agent starts its search until the time when it receives a response from a target agent.
- The number of times of search is 20. The average value is calculated.
- Algorithm A is the simple algorithm shown in Figure 5. Algorithm B shows an algorithm with some modification of Algorithm A, described above.

This result is shown in Figure 8, which means our modification to the original algorithm is superior and effective compare with the original algorithm.

The effect of shadow agent

We tried two differing simulations, in order to confirm the effect of shadow agent. First, the simulation of the existence of some obstacles cutting an agent ring is performed. In this simulation, there are no shadow agents. This simulation result is shown as "Non shadow agents" in Figure 9. The second simulation is performed, in which an agent of agent rings has its shadow agent. This simulation result is shown in "with Shadow agent" in Figure 9.

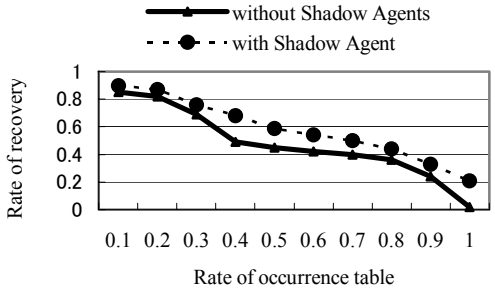

Figure 9 The effect of shadow agent

This result in Figure 9 means that, by existence of shadow agents, agent rings can be recovered with higher fault tolerance degree.

Applications

We are under development of some useful application showing our proposed mechanism. The application is related with an acquisition of the state change in an agent. We introduce our proposal method of agent rings and agent chains into a notification system of website change.

Notification system of website change

The directory server type system has already been used in the current notification systems of website change such as Mind-it [10] by JAPAN PUMATECH. The demerit on this directory server model exists in such a system as described before. The main demerit is that the total system failure is caused by server down and by increase of traffic of accessing to a centre object or a root object. So, we solve this in order to include agent rings and agent chains in a notification system.

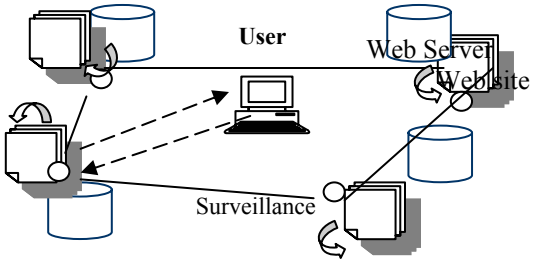

Figure10 Notification system of website change with agent rings

The outline of the notification system of website change with agent rings is shown in Figure. 10. Some of mobile agents continuously monitor many HTML texts on web servers. Each agent can acquired information of website change by cooperative operation based on agent rings and agent chains. The information that an agent acquires mainly includes change information, obtained by surveillance. The agents also exchange the change information that other agents have. It is inefficient that each agent does its notification work each time. Therefore, change information is spread from the agent who acquired the change

information in surveillance, to other related agents using agent rings and agent chains. The change information is rapidly added to agents in the middle of information propagation. Finally a user can obtain change information as someone updates his website information. Moreover, a status and result report is regularly performed to the user from a notification agent. A detailed process is described below:
Step-1) A user gives a circular agent URL information.

Step-2) The agent sends users request to other agents using the communication of information by agent ring. Each agent has user's request and returns to an agent ring. It searches an object and notifies directly to the agent nearest the user when it detects a website is changed.

Step-3) The Agent supervises website change monitoring and detects its change.

Step-4) Detected information is summarized by cooperation operation among agents. The agent who detected website change summarizes the change information using agent rings.

Step-5) Change information is notified to the user.

This notification system of website change required no centralized server as well as heavy traffic between a user side computer and many websites registered beforehand. Our approach will be effective in such environment that a lot of users would like to frequently and rapidly obtain change information of many websites.

5. RELATED WORK

Existing P2P technologies
There are many softwares in the type of P2P computing. Some examples are Napster[5], Gnutella[4], Freenet[6] and Chord[10]. Among these methods, we introduce Chord and Freenet, which both is assumed no servers in their environment.

In Chord[10], some algorithm is developed in order to search objects efficiently. Keys of objects maintained by Chord are hashed using some mathematical function. The hashed keys are stored in Chord nodes forming Chord ring. The searching is done as the same manner as our method. In order to increase the performance, Chord introduces routing information for each node called a finger table. Our proposed method is different from Chord architecture in both fault tolerance mechanism and searching process, using shadow agents and agent chains.

Freenet [6] builds its own network, originally different from Gnutella [4] or Napster [5]. In Freenet network, much information is stored in all the network nodes. The information is transmitted between network nodes. When there is one node that wants information, this node gives search demand to the node near the node and the demand is also passed to the next node. When there is some information satisfied with the demand, the information is sent to the node that receives the search demand first.

Existing technologies on Agent platforms
There are various agent platforms which have already been developed and can be used in actual applications. Among them, we describe two agent development platforms called Mobidget and Aglet.

Mobidget[9] is a programming environment developed by NEC. The main feature of Mobidget is to be able to make Mobidget objects executed wherever these objects are, without any termination. The interpreter is developed based on JavaVM.

Aglet [7] is also a type of mobile agent platforms. On Aglet, objects can be moved and executed on any node of networks. Aglets software development kit called ASDK is provided to develop any mobile agent applications. Aglet is developed by IBM Japan. Aglet execution environment is realized by Java. Aglets can be performed on almost all computers because of the high portability of Java.

When some application development is necessary to combine several mobile agent platforms such as Mobidget and Aglet, there is severe difficulty in communicate with agents in Mobidget and agents in Aglet. It is because that mobile agent identification is not common in both platforms. Our proposed method could be useful in 1) mutual communication between different platforms and 2) the common virtual platform.

6. CONCLUSIONS

In this paper, we present a new information sharing mechanism within mobile agents. The method is server-less based and requires no agent servers to manage mobile agent locations. Therefore, a mobile agent can get another mobile agent, communicate with it and shares information stored in the agent without any knowledge of the physical location of the target mobile agent. Our proposal method has solved the demerit of directry server model and hierarchical search model, and the durability to fault tolerance was also confirmed by some expriment result. We think that our proposal is useful to management of a plant of various devices. For example, we think it very worthy to introduce mobile agent technology into the system that controls each device and performs monitoring for maintenance.

REFERENCES

[1] D. Kotz and R. S. Gray: Mobile Agents and the Future of the Internet, ACM Operating Systems Review, pp.7-13, August 1999
[2] G. Cabri, L. Leonardi and F. Zambonelli: Mobile Agent Technology: Current Trends and Perspectives, AICA '98, Napoli (I), November 1998.
[3] G. Cabri, L. Leonard, and F. Zambonelli: Mobile-Agent Coordination Models for Internet Applications, IEEE Computer, Volume 33 (2): pp.82-89, 2000
[4] Gnutella: http://www.gnutella.com/
[5] Napster: http://www.napster.com/
[6] Freenet: http://freenetproject.org/
[7] Aglet: http://www.trl.ibm.com/aglets/
[8] Objectspace: http://www.objectspace.com/
[9] S. Fujita, K. Koyama, T. Yamanouchi, S. Jagannathan, R. Kelsey, J. Philbin: Mobile and Distributed Agents in Mobidget, Proceedings of the First International Symposium on Agent Systems and Applications Third International Symposium on Mobile Agents, pp.276-292,1999
[10] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOMM 2001, San Deigo, CA, August 2001, pp. 149-160, 2001