A Flexible Network Management Environment using Mobile Agent Technology

Fu-Min Chang, Chih-Mou Shih, Shang-Juh Kao
Department of Applied Mathematics
National Chung Hsing University
250 Kuo-Kuang Rd. Taichung, Taiwan 402
Phone: +886-4-22860133 ext 506
Fax: +886-4-22873028
E-mail: {fmchang,kens,sjkao}@amath.nchu.edu.tw

Abstract

The growth of network in size, in the demand of high availability and in the complexity of management functions requires an efficient, scalable and dynamic network management architecture. Traditional centralized network management architecture can hardly satisfy with this rapid changing network environment. It could suffer from problems such as single point of failure, bottleneck at the manager site and inflexibility. By applying the technology of mobile agent, the load of computing and data processing can be shifted from the manager site to managed devices. This is a novel solution for dealing with the growing size of networks. In this paper, we build a mobile agent execution environment by using the IBM's ASDK and develop a prototyped network management system by taking the advantages of mobile agent technology. We include two management applications, monitor agent and information retrieval agent, to prove its applicability.

Keyword: mobile agent, MASE, network management,

1 Introduction

The rapid evolution in computer industry and network technology prompts the computer network to become the lifeline of business, entertainment, and information dissemination. As the number of computers and network devices increases, the necessity and complexity of network operations direct us to pursue a comprehensive and efficient network management system. Currently, most of network management systems are typically designed following the manager-agent paradigm. In this paradigm, management station polls the network components such as workstations, routers and application servers, to collect related management information. Based on these information, a manager can figure out the system status and infer corresponding events for the over-all "health" or "behavior" of the network. In recent years, there has been a considerable growth in the amount of network devices. Traditional network management system with the centralized paradigm can hardly satisfy with these rapid growths. It would cause the overwhelming usage of the network bandwidth. In addition, the static request and response management operations not only causing the system inflexible, but also suffering from the bottleneck at the manager site.

To overcome these drawbacks, many ad hoc solutions have been proposed, such as Management by Delegation (MbD) [1], CORBA-based Management [2], Web-based

Management [3], Intelligent Agents [4], Active Networks [5], and Mobile Agent [6]. By the definition by Martin-Flatin et al [7], these solutions can be classified into three different categories: weakly distributed hierarchical paradigm, strongly distributed hierarchical paradigm, and distributed cooperative paradigm. In the weakly distributed hierarchical paradigm, a management application is spread over several machines. In this way, scalability could be enhanced but the predefined management functions could lead to the limitation of robustness and flexibility. In the strongly distributed hierarchical paradigm, management stations are no longer restricted to a small set of powerful machines. All agents are capable of performing managing role. Scalability, flexibility and robustness can be easily satisfied, however, to have all managed node get ready in participating management operations is costly and could be too idealistic. Unlike aforementioned paradigms, the distributed cooperative paradigm is goal-oriented. A manager can simply send the “what and “why”, and expect managed agents to know “how” to react. Although they can also meet the management requirements, such as scalability, flexibility, and robustness, they are too complex to implement.

In this paper, we propose a flexible network management environment using mobile agent technology. By applying the technology of the mobile agent, the computing load and the data processing can be moved from the manager site to managed devices. This is a feasible solution for dealing with the growing size of network. In addition, the attractive feature of mobility is just right for solving the flexibility problem and reducing the loading at the manager site.

The remainder of this paper is organized as follows. In section 2, we describe the proposed network management architecture. A prototyped network management system based on the proposed architecture is proposed next. Two applications: Monitor Agent and Information Retrieval Agent are included. Finally, summaries and conclusions are presented.

2 The Proposed Architecture

Our proposed architecture is comprised of three components: Network Management Station (NMS), Mobile Agent Service Environment (MASE), and Mobile Agent (MA). The NMS can launch mobile agents with management tasks, policies and travel plans. It also plays the role of displaying the returned results. The MASE provides a running environment for mobile agents. It also provides an interface to allow MAs can communicate with the local managed resources. A mobile agent plays the role in actually carrying out the management services. It can migrate between the managed entities to perform a given task based on a pre-defined policy. The whole architecture is shown in Figure1. In the following, we will briefly describe the constitution of each components of proposed architecture.

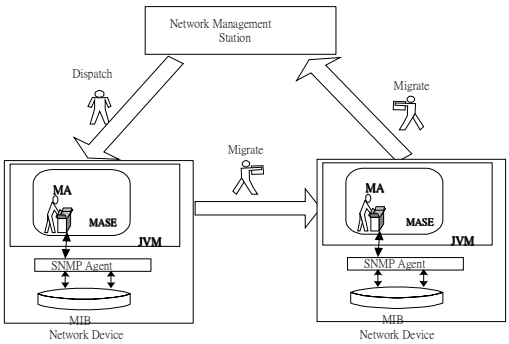

Figure 1 The Proposed Architecture

2.1 Network management station

In our architecture, NMS consists of five components: user interface, manager agent, device information agent, service information agent and service agent library. ***User interface*** provides a user-friendly interface for network administrator. It governs the displays of information collected by the MAs. ***Manager agent*** is the main agent in the NMS. It takes charge of the communication with the user interface to provide the necessary information and perform the instructions. ***Device information agent*** keeps the information of managed devices. When a managed device starts, it sends a message to device information agent for the registration. ***Service information agent*** manages the Service Agent Library in the NMS. Once a manager asks for a service, it will offer the up-to-dated service agent's information, which are currently in the service agent library, to the manager. ***Service agent library*** stores service agents that are defined in advance. The constitution of NMS is illustrated in Figure 2.

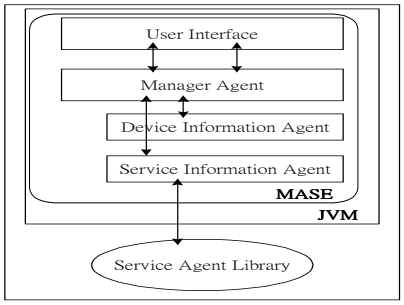

Figure 2 NMS's Constitution

2.2 Mobile Agent Service Environment (MASE)

In our architecture, every managed device runs a Java Virtual Machine (JVM) [8], which is well fit for the distributed network management, and executes a Mobile Agent Service Environment (MASE) that enables the execution of mobile agents. The MASE provides an interface for an agent to communicate with local managed resources. As shown in Figure 3, the MASE consists of seven components: security facility, communication facility, agent loader facility, class loader facility, transfer facility and agent control facility.

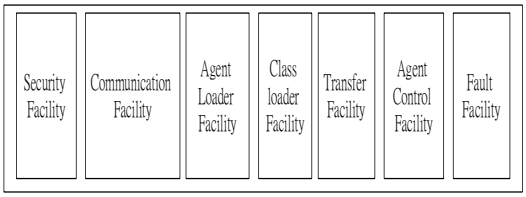

Figure 3 Structure of Mobile Agent Service Environment

Security facility*:* Mobile agents need to be authenticated at each managed device before any task is performed. This will prevent the unauthorized mobile agents.

Communication facility: The MASE provides an interface for an agent to send a message to another agent Communication is very important, especially, when a task must be cooperated by several mobile agents.

Agent loader facility: When a new agent is generated, the MASE has to load the agent into the memory and starts execution.

Class loader facility: There are some common classes, which are desired by a general mobile agent, and they can be loaded into the local class library. This can minimize the size of the mobile agent and reduce the traffic load of the network.

Transfer facility: The MASE is also responsible for sending and receiving mobile agents. A mobile agent needs to be serialized and encoded before starting migration. Similarly, an agent needs to be deserialized and decoded, after the migrating.

Agent control facility: The managed device strictly specifies the interactions between the

local environment of the managed device and the mobile agent. The MASE provides a mechanism which the local resources can be accessed by a mobile agent. The MASE program specifies the policy that governs the mobile agent's interaction with the local resources. At the same time, MASE has the authority to deny any service for the mobile agent that violates the trust.

Fault facility: It deals with the faulty situations, including connection error, transfer error and other error among others.

2.3 Management agents

Management tasks are accomplished by a set of mobile agents in our architecture. They are collaborated to support various tasks delegated by network manager. Agents are distributed and run across different managed network devices. In our architecture, other than the agents existed in the NMS, three different types of agent, which are named register agent, service agent and information agent, get involving in the management process to perform management tasks.

Register Agents

When a managed device comes up, it automatically starts the mobile agent service environment. When a MASE has been set up, it will create a register agent, as shown in Figure 4. The register agent then sends the registration message to the device information agent within the NMS. The device information agent in turn adds this device into the list of the active devices. With the assistance of register agent, network manager can keep the newest topology of managed devices and provide the device information for management applications. Meanwhile the device information agent can send a verification message to the register agent to make sure the managed device is alive, if required.

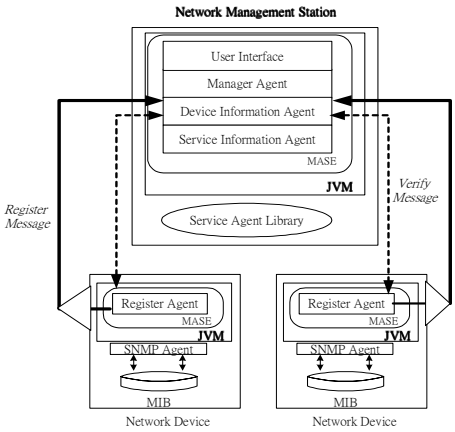

Figure 4 Register Agents

Service Agents

Service agents are responsible for performing the given task from manager. Service agents have the knowledge of task domain, which other relevant agents participate in performing part of the task. The service contents, such as formulating problem, solving plans, executing plans, querying, exchanging information with other agents, can be predefined by the manager. This makes service agents with the ability to perform most of the autonomous problem solving.

The relationship between service agent and the NMS is shown in Figure 5. When the manager asks for a service through the user interface, the manager agent loads service agent from the service agent library and lunches it to the indicated device. After passing the process of authentication, the service agent can interacts with the SNMP agent on the local device to execute the given request. The service agent can stay on this managed device or migrate to another managed device according to whatever defined in the service contents. It also has the ability to communicate with the manager agent

on the NMS, if necessary. With the ability to stay on the managed device and interact with the SNMP agent, the service agent can perform designated service given by the manager. Consequently, this significantly reduces the traffic load of network and the computation load of NMS.

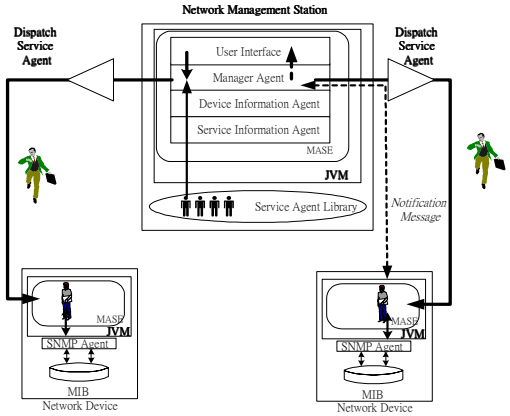

Figure 5 Service Agents

Information Agents

An information agent can be dispatched by network manager through the manager agent on the NMS to aggregate the information from the service agents on various devices, as shown in Figure 6. Information agents can intelligently access the related management information on managed device through the service agent. With the appropriate scheduling mechanism, interaction rules, and traveling plan, information agents reduce the network traffic load and make the network management more intelligent.

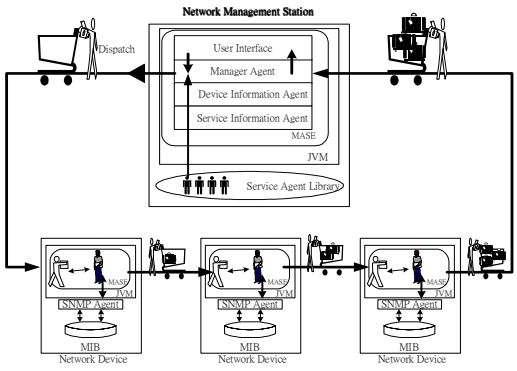

Figure 6 Information Agents

3 A Prototyped Network Management System

We have developed a prototyped network management system based on the proposed network management architecture in the distributed system laboratory at the department of applied mathematic, National Chung-Hsing University. The system contains a network management station and several managed devices with different operating systems. Each managed device contains a mobile agent execution environment (described later) and an SNMP [9] agent. The experimental system is configured as shown in Figure 7 with related system attributes listed in Table 1.

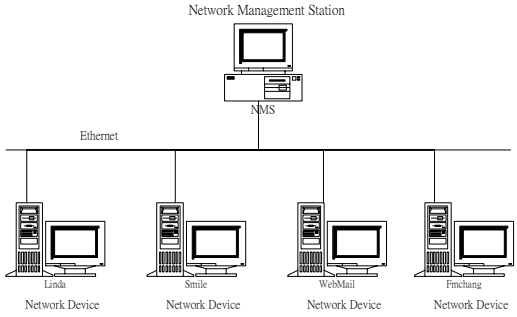

Figure 7 Experimental System Configuration

	IP	OS	SNMP Agent	IBM ASDK
Fmchang	140.120.8.26	Win2K Professional	Mirco-Soft	1.2.0
Smile	140.120.8.10	Win2K Professional	Mirco-Soft	1.2.0
Linda	140.120.8.37	Win2K Server	Mirco-Soft	1.2.0
Nms	140.120.8.8	Win2K Professional	Mirco-Soft	1.2.0
Webmail	140.120.8.6	RedHat 7.0	CMU	1.2.0

Table 1 Experimental System Attributes

3.1 System Implementation

Based on the concern of practice implementation, we choose the free mobile agent

development tool, ASDK [10], which is downloaded from the IBM website. The ASDK provides tools for developing mobile agent called Aglet [11,13], which is a Java object. There is a mobile agent execution environment, named Aglet server, running within JVM. We also set up the Java running environment by the JDK of Sun Microsystem. In order to integrate the SNMP agent into our managed device, we adopt the AdventNet SNMP utilities API from the AdventNet [12] in the development of mobile agents. In the following, we will address the details of the components of the system.

Network Management Station

Network management station is the kernel of network management system and it must provide a friendly user interface for network manager. In our experimental system, we adopt the software named **Tahiti** developed from IBM ASDK to play the role of a mobile agent execution environment. It provides a graphical interface (Figure 8) for monitoring, creating, dispatching and disposing the agents and for setting the agent access privilege for the agent environment.

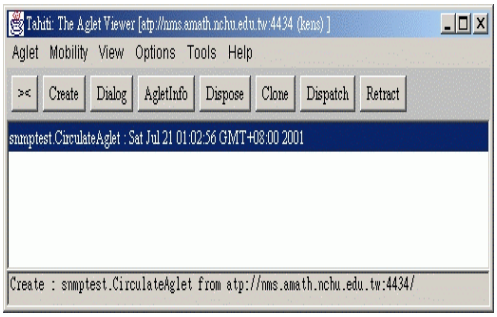

Figure 8 Graphic interface

When network manager needs to launch a new mobile agent to perform the management task, he(she) can create a predefined mobile agent from the service agent library and dispatch it to the destination network device as shown in Figure 9.

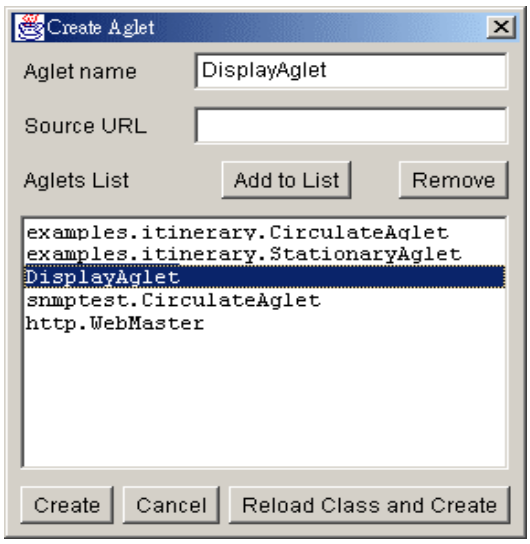

Figure 9 Creation of a mobile agent

Register Agent

In order to keep the information and the topology of managed devices, a register agent automatically creates when the managed device starts. The register agent sends a registration message to the device information agent on the NMS. The device information agent keeps the managed device information and provides it to the manager when needed. As shown in the Figure 10, managed device (Linda) creates a register agent and sends the registration message to the NMS (NMS).

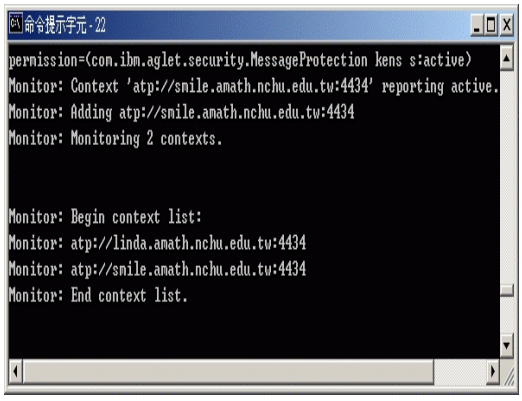

Figure 10 Register Agent

3.3 Management Applications

In our architecture, mobile agents with management tasks can be classified into service

agents and information agents. We presented two management applications: monitor agent and information retrieval agent, which belong to the service agent and information agent respectively.

Monitor Agent

An important mission of the network management is to keep monitoring the network devices or the application programs to make sure that they are working properly. One of the solutions is to monitor on the log file or the particular data, hence we designed a monitor agent, which is an example of service agents. A monitor agent is a mobile agent which can be sent by a network manager or a management application to perform the monitoring task on the managed device. The monitor agent in this example is to monitor the assigned file on the destination network device. As shown in Figure 11, the manager can chose a destination network device and assign the file with the full path. The check interval and duration are made variously which are 10 seconds and 1 minute, respectively by default.

For example, as shown in Figure 11, the manager fills in the task for the monitor agent. The agent is asked to monitor the assigned file of the destination device (Smile). The monitor agent creates a notifier agent and dispatches to the NMS. After arriving the destination, the notifier agent will check the assigned file on the local host to get the initial time stamp. Then, the notifier agent will ask the SNMP Agent for the same object every 30 seconds for a duration of 24hours for monitoring the file changes. The result is in turn sent back to the monitor agent and displayed on the manager's window.

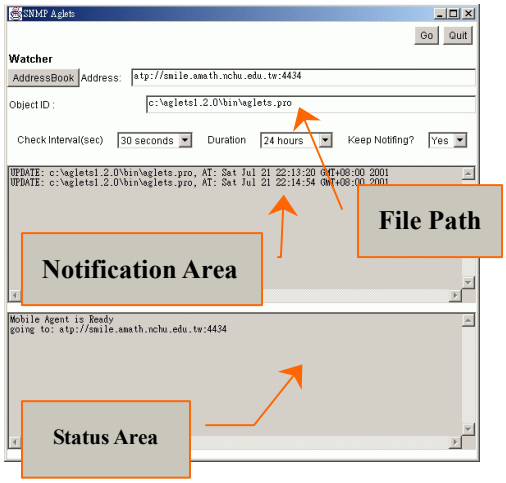

Figure 11 Monitor Agent

Information Retrieval Agent

An information retrieval agent is an application of information agents. Network manager can initiate an information agent manager to perform the information retrieval task. The manager assigns different tasks for various devices, as shown in Figure 12. We have predefined three functions for the information retrieval agent including getMibValue, getLocalInfo and printResult. The function of getMibValue is to gather the system information of the destination device. After arriving at the managed device, the agent communicates with the SNMP agent and asks for the value of sysDescr, sysObjectID, sysUpTime, sysName and sysService. The function of getLocalInfo is to gather the mobile agent execution environment information including, username, home directory, current working directory, machine architecture, OS name, OS version and Java version. The function of printResult is to print whatever the results that information retrieval agent has collected.

In our demonstration, as shown in Figure 12, the agent was assigned to retrieve various information from different network devices, as shown in Figure 13, and prints the result on the

NMS (NMS). The first destination of the information retrieval agent is “Smile” and the task is to communicate with the SNMP agent to collect the values of the getMibValue. After completion, the agent migrates to the second destination “fmchang” with the results collected from “Smile” to continue the given task “getLocalInfo”. The information retrieval agent continues the traveling by visiting “Linda” and “Webmail” for Mib Values. After visiting all destined devices, the information retrieval agent goes back to the NMS and displays the results, as shown in Figure 14.

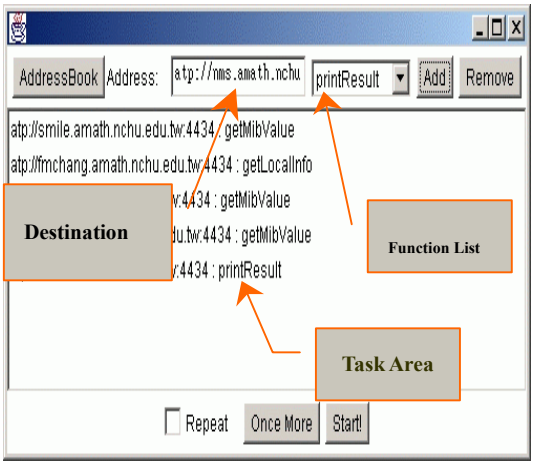

Figure 12 Information Retrieval Agent.

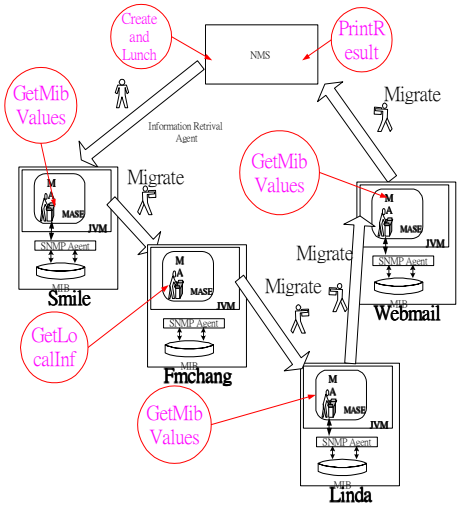

Figure 13 Task of the Information Retrieval Agent

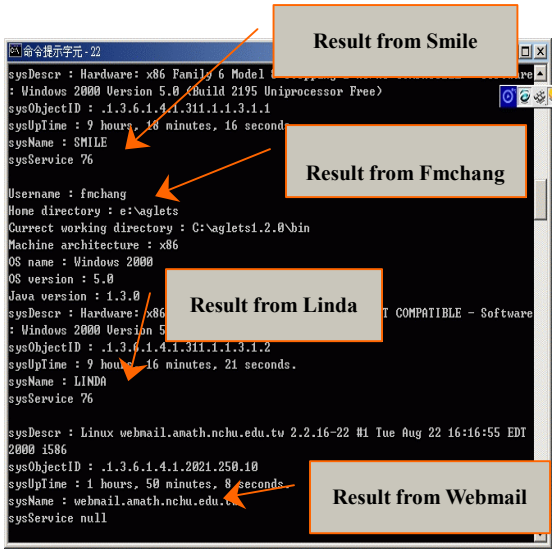

Figure 14 Result of the Information Retrieval Agent

4 Conclusions

As the number of network devices increases, the network management becomes more complicated. Due to the problems of poor scalability, flexibility, and robustness in the traditional manager-agent based network management system, a new network management system should be simple enough to cope with the problems mentioned above. In this research, we take the advantageous features of mobile agent technology and implement a mobile agent-based network management system. In the system, we can delegate the management task to a mobile agent to perform management functions. In addition to the manager agent existed in NMS, there are three other agents: register agent, information agent, and service agent. All agents execute in MASE (Mobile Agent Service Environment) to cooperate with each other to accomplish a management task.

We also include two management applications, monitor agent and information retrieval agent, to demonstrate the advantages of the mobile agent

technology. A monitor agent monitors the assigned file on the network device that reduces the polling frequency. An information retrieval agent reduces the traffic load and makes the management more flexible. The management application scenarios give a demonstration of applying the proposed architecture to build a flexible and efficient network management system. The experimental system reveals the applicability of our architecture to meet the requirements of efficiency, extensibility and flexibility.

References

[1] G. Goldszmidt, "Distributed Management by Delegation", Ph.D thesis, Columbia University, New York, NY, USA, December 1995.

[2] J. Pavon, J. Tomas, "CORBA for Network and Service Management in the TINA Framework", IEEE Communication Magazine, 36(3): 72-79, 1998.

[3] J.P. Thompson, "Web-Based Enterprise Management Architecture", IEEE Communication Magazine, 36(3): 80-86, 1998.

[4] T. Magedanz, K. Rothermel, S. Krause, "Intelligent Agents: An Emerging Technology for Next Generation Telecommunications?", Proceedings of INFOCOM'96, March 24-28, 1996, San Francisco, CA, USA.

[5] D. L. Tennenhouse, J. M. Smith, W. D. Sincoskie, D. J. Wetherall, G. J. Minden, "A Survey of Active Network Research", IEEE Communications Magazine, 35(1): 80-86, 1997.

[6] V. A. Pham, A. Karmouch, "Mobile Software Agents: An Overview", IEEE Communication Magazine, 36(7): 26-37, 1998.

[7] J.P. Martin-Flatin, S. Znaty, and J.P. Hubaux. "A Survey of Distributed Enterprise Network and Systems Management Paradigms". Journal of Network and Systems Management, 7(1): 9-26, March 1999

[8] Sun Microsystems, “Java Language Overview-White Paper”, http://www.javasoft.com/docs/white/index.html

[9] J.Case, M Fedor, M. Schffstall, J. Davin, “ A Simple Network management protocol (SNMP)”, RFC 1157, May 1990.

[10] http://www.trl.ibm.com/aglets/

[11] IBM Tokyo Research Laboratory, “About Aglet Workbench”, http://www.trl.ibm.co.jp/aglets/about.html

[12] http://www.adventnet.com

[13] M. Oshima, “Programming and Developing Java Mobile Agent with Aglets”, Addison-Wesley, 1998.

[14] M. Kahani,H.W. P. Beadle, “Decentralized Approaches for Network Management”, ACM Computer Communication Review journal, July 1997.

[15] T. Shu, C. Yang, Y. Jianhua, X. Ning “A Mobile Agent Based Approach for Network Management”, IEEE 2000.

[16] W.J Buchanan., M. Naylor, A.V. Scott. , “Enhancing network management using mobile agents “, Engineering of Computer Based Systems, 2000, (ECBS 2000) Proceedings. Seventh IEEE International Conference and Workshop on the , 2000 , Page(s): 218 –226

.