International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 12
ISSN: 2454-4248
190 – 196

Operability of Mobile Agent Applications in a Protected Environment

B. M. G. Amosa, J.B, Ekuewa, O.O. Oyetunji, C. Nwaekpe and T. Ogunleye
Department of Computer Science, Federal Polytechnic, Ede. Nigeria
amosabmg@gmail.com

Abstract **-** There is a shift toward increasingly heterogeneous networks in today's communications environment. Such diversity requires that network operators have greater experience and increased training. Managing these diverse networks especially in institutions requires the collection of large quantities of data from a dependable network that must be analyzed before management of any activity can be comenced. In this research, we have identified the operability of mobile Agents in a protected network environment.

Keywords **-** *Operability, Agent, Protected Environment, Networks.*

*****

I. INTRODUCTION

Managing and monitoring applications in the networks with hundreds of computers has become a challenging and tedious task for today's system administrators. A general computing infrastructure in a medium to a large organization with many nodes, possibly of different kinds, organized into multiple local-area networks and administrative domains.

A mobile agent represents a program capable of migrating from one node to another in a network to perform certain designated tasks [1]. The ability to migrate code and processing functions to a remote node offers the potential benefits of reduced network traffic and bandwidth requirements.

This research is motivated by many factors. Many computing environments, such as campus networks, tend to be relatively open. It is the duty of a system administrator to monitor the

environment for suspicious activities actively. Large distributed systems require dynamic and scalable architectures for monitoring. Dynamic structures are required to support changes to policies for monitoring, collection, and processing of information at all levels of a system's organizational hierarchy. It should support the definition of new event types and installation of specific detection mechanisms at specified nodes. It should be possible to install a new monitoring agent at a node, change an existing one, or within a domain, update the current event notification policies to implement new data management structures. It should also be possible to enforce desired security policies for event reporting and processing functions across different administrative domains. For scalability, the infrastructure should support any desired hierarchical and decentralized organization for information collection and processing. Moreover, the system should support the incorporation of new correlation and search functions across different event databases.

A computer network is a collection of computers connected and separated by physical distance primarily to search for, share and exchange computer resources.

Over the years, monitoring and searching for resources on the network often involved the physical movement of the network administrator from one computer to another [2]. Human administrators of network systems have been used in gathering data for network management. Their work involves monitoring, evaluating and analysis of the various nodes attached to the network or intending to resolve the problems and ensuring optimal performance and efficiency. This function can be tiring, stressful and cumbersome, especially in a large network. A major limitation of the manual approach is that humans cannot monitor events on the network real-time, that is, as the events occur. The network administrator can also be bored and confused about which node to monitor next. It is, therefore, apparent that manual network management cannot efficiently satisfy the requirements of the modern complex network systems.

II. NETWORK SECURITY AND MANAGEMENT

There exist many contemporary approaches to network security categorized as Host based and Network based. However, they work for Intrusion detection and not for overall security management. As mentioned earlier, Mobile Agents can be useful in such places where we need network security as well as network management. We can use different mobile agents for securing the network from the threats as well as detect the threats. For example, network sniffing detector mobile agent used to find the network sniffer program in the network. Mobile agents can be used in above context as follows:

Network load Reducing- Due to the multiple interactions in the network, it creates excess network traffic. A mobile agent through the package conversation they dispatch the packets on

the destination host at that time locally interaction happens, and it helps to reduce the network load.

Overcome Network Latency- In real time systems, with the help of mobile agents overcome the network latency, because mobile agents dispatch from the central controller and acts locally.

Tolerant to Network Faults- Without an active connection between clients and server mobile agents can operate.

Encapsulate Protocols- When data is being exchanged in the network at that time every host has a code, for this code needs protocols e.g. incoming and outgoing. When these protocols require security at that time, protocol code becomes cumbersome and creates problems. Mobile agents move on the remote host and using specific channels creates new protocols.

Execute asynchronously and autonomously- It is possible that the embeded different tasks in the mobile agents are likely dispatched on the hosts. When agents are sent, they become independent from the process and due to this mobile agent become asynchronous and autonomous.

Adapt dynamically- Mobile agents have their sense about execution environment because they react autonomously to the changes. They solve a particular problem in the network by their own.

They are Naturally Heterogeneous- Network computing is itself heterogeneous in respect of hardware and software; therefore, mobile agents are also heterogeneous in nature [3]. In the case of network management, the Mobile agents assist the network administrator to manage the network security [4]. For security management mobile agent's team launched in the network, this team visits all the computers in the network and different services security software analyzes and install. For this, mobile agents uses following techniques;

- Connectivity and states of remote hosts are checked and reported.
- The configuration of remote hosts is checked and recorded.
- Security configuration management related tasks are applied.
- Mapping of Snort rules and identified vulnerabilities.

For completing the above four function, mobile agents team automating launched, these teams interact with all the system and install security tools on the remote hosts and complete the desired network security management tasks. Similarly, we can detect intrusions also. Intrusion detection is implemented by an intrusion detection system and today there are many commercial intrusion detection systems available. In general, most of these commercial implementations are relatively ineffective and insufficient, which gives rise to the need for research on more dynamic intrusion detection systems [5]. Mobile agents play a crucial role in the network security. Mobile agent searches the malicious activity in the network, for these work mobile agents provides following three groups;

- Analysis of large volume of data in various logs generation of effective reports.
- Detection of and reaction to host-based intrusion attempts in real time.
- Detection of and reaction in real time of distributed intrusion attempts.

For completing the above three functions, the different mobile agent teams assembled and launched. The capacity of these teams to analyzes the logs, these logs created from sensors e.g. Snort, Osiris and MS Windows firewall that are present on the host computers. In [6], Mobile agent reaches on the remote hosts, analyzed the logs and if any serious problem then reports to the security administrator. At the same time, the second team of mobile agent reaches the remote host and continuously snorts, monitor and analyze if this team finds any suspicious activity calls the new mobile agents, and lastly the third team of mobile agents detects intrusion activity.

Above case can be extended to Distributed systems also. Today, the computer system has evolved into a distributed computing machine, nothing is static now, not even the security threats and attacks. The security issues are of high concern today. In the world of open environment, the problem faced widely by the computer system and network intrusion [7].

Intrusion detection system is the security mechanism that gathers and analyzes the information to detect unwanted attempts at accessing and manipulating the user and system activities and report it to the management section.

In his view, [8], network management is a means to deploy effectively and coordinate network resources. That is, it helps to plan, administer, analyze, detect, evaluate, design and expand communication networks to meet demands at all times, at a reasonable cost and optimum capacity. Also, network management can be described as the activities, methods, procedures and tools that deal with the operation, administration, maintenance and provisioning of networked systems [9]. Network management has been extensively discussed by [10], [11]. Effective management will require monitoring and controlling the resources of the network. Monitoring of software tools in a computer network environment is, therefore, a crucial part of network management not addressed in that research work [12]. The Specific objectives of this research is to identify the operability of mobile Agents in a secure (protected) network environment.

 ISSN: 2454-4248
190 – 196

III. LITERATURE REVIEW

The use of mobile agents for network applications management has been proposed and investigated by several researchers in the recent years with the primary goal of reducing network traffic and building scalable systems[13]. Mobile agents can be implemented using one of two fundamental technologies: mobile code, [13] or remote objects [14]. Electronic commerce transactions in [15], [16], [17], [18], [19], [20]. Distributed information Retrieval in [21], [22], [23], [24]. Network management in [25]. Mobile computing in [26], [27]. Workflow management as presented in [28]. Internet chat applications presented in [29].

Some reasons for using mobile agents have been identified in [30], [31],[3], [32] [33],[34]. There are several advantages of using mobile agents [35], [36]. Issues of disadvantages cannot be overlooked in Mobile Agents. Some of these are summarized are [37].

IV. MOBILE AGENT CYCLE

Mobile agents go through some processes to get the job done. A complete process termed Mobile Agent Life Cycle is depicted in (Figure 1), and as highlighted in [38], [39], [40] are as follows:

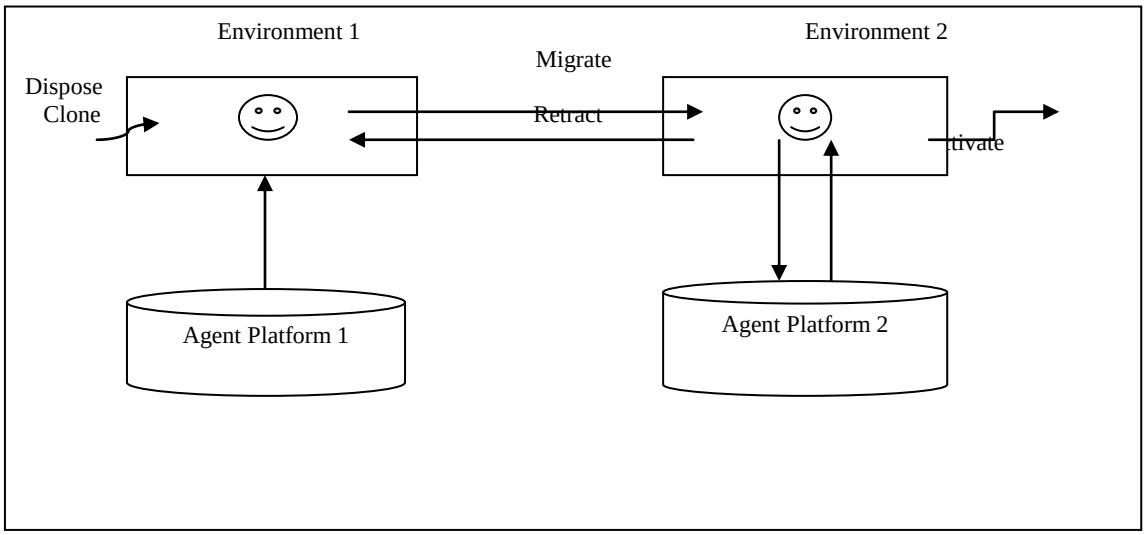

Figure 1. Mobile Agent Life Cycle

Creation: this is the first phase of a mobile agent life cycle. Once a request is made to a mobile agent, an instance of the mobile agent is created, and its state is initialized.

Migrate: this involves the movement of the mobile agent from one node to the other and can be achieved by specifying the address of the destination.

Cloning: this refers to creating a copy of the original mobile agent object. That is to say, a twin agent is born, and the current state of the original is duplicated in the clone.

Deactivation: a mobile agent is put to sleep, and its state is stored on a disk of the host.

Activation: a deactivated mobile agent is brought back to life, and its state is restored from disk.

Retraction: an agent is brought back from a remote host along with its state to the home machine after the completion of its job.

Disposal: this is done at the end of the mobile agent life cycle. The agent is terminated, and its state is lost forever.

V. PERFORMANCE TOOL OF THE MOBILE AGENT IN A PROTECTED ENVIRONMENT

The main goal of this model is to increase the performance of the mobile agent. After the mobile Agent completes a part of its journey; this model can reduce a mobile agent size by removing some unwanted parts from its body. Consequent on when a mobile agent data is reduced, automatically the mobility of the agent consumes less time. Also, the mobile agent size will make it acceptable to all the hosts. As defined earlier, the mobile agent comprises of several components. These components represent tasks at each place. After the agent completes a part of its tasks during a journey, some of its components are not useful for the rest of its journey. Therefore, these components are overhead to the mobile agent.

Basic assumptions for consideration [41];

- Mobile agents visit many places on a journey.
- Mobile agents may perform different tasks in those places.
- Mobile agent system uses robust mobility mechanism.
- The mobile agent system is made up of many

controllers (machines) that are distributed in a system domain.

- A controller is a secured place and the main role of this place is to reduce mobile agent size (Location for decreasing weight).

A. Components of the Model

The role of each component is as follows:

- Agent catalog

The component contains information about elements that are included in the mobile agent and there status. They are; Element ID, Place, and Status. The Element ID is represented by a method or a variable ID. The Place specifies on which places the element will be used. The Status takes value ON in case the agent still needs this element and value OFF in case the element will not use again. Each component in the mobile agent is represented as a record in the agent catalog. The agent catalog is distributed over mobile agent body. According to mobile agent content, the mobile agent Base creates and encrypts Agent catalog by using Symmetric Encryption mechanism. Only controllers can deal with Agent catalog for security reasons. No host can deal with the agent catalog to protect the catalog against any attack (deletion or modification).

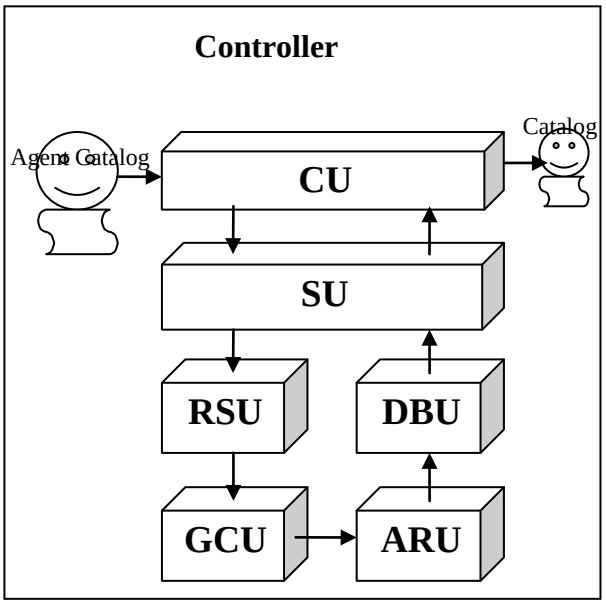

Figure 2. Controller architecture

- Controller

A controller (Figure 2) is a safe place for reducing an agent's size. The main aim of this place is to perform the operation of reducing size securely. The mobile agent system distributes some controllers around hosts. The mobile agent can visit many controllers during its journey. After the mobile agent visits some hosts, it may migrate to the controller to remove some unused parts from the agent. Each controller in the system knows a secret key that is used in the agent catalog encryption.

To achieve its duties, This place introduces some services to the visitor agents as follows:

- Decrypting Agent catalog with the secret key and according to journey history records, it updates Agent catalog status.
- Specifying all elements that will not be used in the remaining journey and assigning them to be deleted
- Rebuilding the mobile agent in a new form by eliminating the deleted elements.
- It rebuilds a new version of the agent catalog according to the new mobile agent.
- Encryption of the Agent catalog again with the secret key.
- Enclosing Agent catalog with the mobile agent.
- It allows the mobile agent to continue its journey.

To achieve its duties, the controller uses some units as follows:

i. Garbage collection unit (GCU)

This unit make use of a mobile agent, an agent's itinerary and an agent catalog to specify the deleted Items that should be removed to reduce the agent's size.

ii. Agent rebuilds unit (ARU)

After GCU has specified the deleted elements, the ARU removes all these elements from the mobile agent and rebuilds a new version of it. Also, it updates the agent

catalog according to existent agent elements. With this operation, the size of the mobile agent is reduced.

iii. Results Summary Unit (RSU)

Before the mobile agent arrives at the controller, it visits some hosts. So that, to collects some result. RSU can carry out and make a summary of these results. This way allows the controller to assign more agent elements as deleted elements.

iv. Dynamic Behaviour Unit (DBU)

In some situations, mobile agents may be interested in visiting new places that are not scheduled for their itinerary tables. Without gainsaying, these agents need behaviours that allow them to deal with these places. DBU can play this role as dynamic behaviour provider according to the mobile agents.' requests.

v. Communication Unit (CU)

It is the role of this unit is to receive and dispatch mobile agents to/ from the controller. Also, it can provide an agent's owner with some immediate results. CU allows the controller to receive many agents at the same time through multithread paths.

vi. Security Unit (SU)

Most mobile agent systems encrypt their mobile agent for security reasons. Also, Agent catalog is moved in encryption form. SU can be used to decrypt the mobile agent and the agent

 ISSN: 2454-4248
190 – 196

catalog when the mobile agent arrives at the controller. It can also be used to encrypt them when the mobile agent leaves the controller.

B. *Migration of the Agent*

In [42] and [43] mobility allows the transfer or migration of a mobile agent to another host, as well as the resumption of execution at the new host. The execution state is migrated with the code in order for the computation to resume at the destination. According to the amount of detail captured in the state, we can classify agent migration into two types: strong and weak.

i. Strong migration is the ability of an agent to migrate to a network, carrying the code and execution state, where the state includes the program counter, saved processor registers, and local variables, which correspond to variables allocated in the stack frame of the agent's memory space, global variables. These correspond to variables allocated in the heap frame. The agent is suspended, marshaled, transmitted, unmarshaled and then restarted at the exact position where it was previously suspended on the destination node without loss of data or execution state.

ii. Weak migration is the ability of an agent to migrate to a network, carrying the code and partial execution state, where the state is variables in the heap frame, e.g., instance variables in object-oriented programs, instead of its program counter and local variables declared in methods or functions. The agent is moved to and restarted on the destination with its global variables. The runtime system may explicitly call for the special agent methods.

Strong migration can overcome the weak migration, but it is a minority. It is because the execution state of an agent tends to be large and the marshaling and transmitting of the state over a network need heavy processing. Moreover, like the latter, the former cannot migrate agents that access the computational resources only available in current computers, e.g., input-and-output equipment and networks. The former unfortunately has no significant advantages in the development and operation of real distributed applications as discussed by [44].

VI. EVALUATION OF THE MODEL PERFORMANCE OF THE MODEL

The implementation of this model is in a protected environment. When the agent visits a node, it will collect the information on security applications and then visit the controller component attached to it. The controller will remove all the unwanted items from the database; this will lead to a reduction in the size of the Mobile Agent. The passing of this process from node to node will give a reduced average load during the migration of the mobile agent in a protected environment The LAN environment of the Federal Polytechnic, Ede, Nigeria has been used for the Implementation of the model. The model was tested in protected environments with 15 nodes and the host.

A. *Verification of the Mobile Agent in a protected environment.*

The mobile agent base creates a mobile agent. The agent will visit Node1, Node2, Node3, Node4, Node5 ….Node15, to collect information about security applications in each of them. After completing the migration, the agent returns to home. Table 1 presents the mobile agent size during the journey while a corresponding graph is in Figure 3

Table 1: Mobile Agent size in a protected environment

Nodes	Load
Home	7710
1	7300
2	6885
3	6475
4	6065
5	5650
6	5235
7	4825
8	4410
9	3995
10	3585
11	3170
12	2755
13	2345
14	1940
15	1530
Average	**4616**

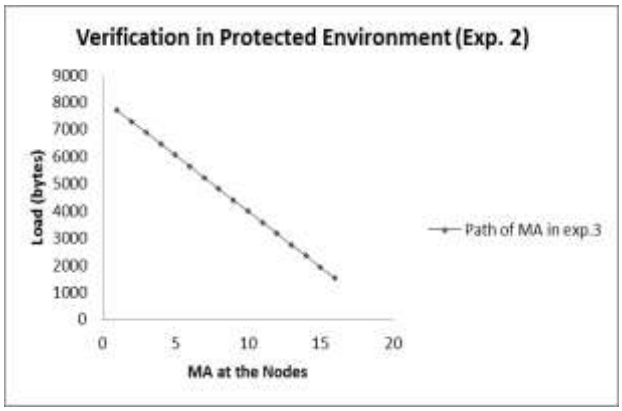

Figure 3; Verification of Mobile Agent in a Protected Environment

B. *Protected environment*

In the protected environment, the size of the mobile agent (exp.1), increased by 6180 bytes (80.2%); as presented in Table 1. The average size of the mobile agent during the migration is 4616 bytes. This increase is because of the inclusion of the controller in the model. The chart for this experiment is in (Figure 3).

The characteristics of Mobile Agent in a Protected Environment are:

a. Autonomist: the ability of the agent to execute without the need for human interaction. This feature does not prevent intermittent interaction that might be required from time to time.
b. Intelligence: the ability of the agent to learn, and adapt over time. Learning is crucial for intelligent mobile systems and enables them to adapt their behavior accordingly.
c. Communicative: an agent should have the ability to communicate with other agents for the purpose of exchanging data. This communication should be regulated and monitored some how to prevent security breaches.
d. Goal Oriented: The mobile agent should have been oriented to a achieve a goal. This goal is explicitly stated in its internal plan of action.
e. Mobility: Mobile agent can decide to migrate to a different machine or network while maintaining their persistence (consistent internal state over time)
f. Perceiving: A mobile agent should perceive its surrounding environment and react or response accordingly. Sometimes agents should not just react; they may take active steps to change that environment according to their desire.

VII. CONCLUSION

In this research, we have presented the operability of a mobile agent in a protected network. The major idea behind the model is to reduce the size of the mobile agent. The model allows the mobile agents to visit all the available places during the migration. During migration, some nodes may refuse the mobile agents because their sizes are large and not acceptable to store them. The model can help in this situation. Two experiments have been performed and according to the result, the model proved its efficiency in reducing the mobile agent size. The Analysis from the experiments shows that the mobile agent model performs well in a protected network environment.

REFERENCES

[1] Harrison, C.G., Chess, D.M. & Kershenbaum, K. (1995). *Mobile Agents: Are they a good idea*? IBM Research Report 19887, IBM Research Division, 1995.

[2] Imianvan, A.A. (2008). "*Development of a mobile agent for evaluating the use of bandwidth in a computer network*." A Ph.D. Thesis in the Department of Computer Science, Federal The university of Technology, Akure, Nigeria,

[3] Lange, D. & Oshima, M. (1998). *Programming and Deploying Java Mobile Agents with Aglets*, Addison

[4] Ching-hang, F., Parr, G. & Morrow, P. (2011). *Security Schemes for a Mobile Agent Based Network and System Management Framework*. Journal of Network & Systems Management. Vol. 19, Issue 2, pp.230-256
Wesley, ISBN: 0 201 32582 9.

[5] Mohammad, S.H., Md. Abdul Mukit. & Md. Abu Naser Bikas, (2012) "*An Implementation of Intrusion Detection System using Genetic Algorithm"* International Journal Of Network Security & Its Applications

[6] Muhammad, A., & Shibli, S.M. (2008). "*Intrusion Detection and Prevention System Using Secure Mobile Agents*" IEEE International Conference On Security And Cryptography, (pp. 76-82), Porto Portugal.

[7] Shiv, S.S., Nitin, G., Saugata, G. & Saurabh, C. (2011). "*A Survey on Mobile Agent based Intrusion Detection System*" International Symposium on Devices MEMS, Intelligent Systems & Communication (ISDMISC) 2011, Proceedings published by International Journal of Computer Applications (IJCA).

[8] Qiang, C. & Marshall, A. (2004). *Network Management Performance Analysis and Scalability Tests: SNMP vs COBRA*. IEEE/IFIP Network Operations and Management Symposium, NOMS, South Korea

[9] Ding, J., Peter, J.C., Dianxiang, X., Hudong, H. & Deng, Y. (2010). "*A Formal Model-based Approach for Developing and Interoperable Mobile Agent System*", Multiagent and Grid Systems- An International Journal (IOS Press): pp. 401-412.

[10] Cassel, L.N., Patridge, C. & Westcott, J. (1989). "*Network Management Architecture and Protocols: Problems and Approaches*". IEEE Journal on Selected Areas in Communications, Vol 7, No. 7, pp. 1104-1114.

[11] Allan, L. & Karen, F. (1993). "*Network Management: A Practical Perspective*." Addison Wesley.

[12] Zhang, J. (2011). *A mobile Agent-based Tool Supporting Web Services Testing*. Wireless personal communications. Vol. 56, Issue 1, pp.147-172.

[13] Baldi, M., Gai, S. & Picco, G.P. (1997). *Exploiting Code Mobility in Decentralized and Flexible Network Management*. In *Proceedings of the Workshop on Mobile Agents (MA'97) – LNCS 1219*. pp. 13–26.

[14] Vinoski, S. (1997). *CORBA Overview: CORBA: Integrating Diverse Applications within Distributed Environments*, IEEE Commun. Mag., Vol. 14, No. 2.

[15] White, J. (1996). *Mobile Agents White Paper*, Available at: http:/citeseer.ist.psu.edu/white96mobile.html [Accessed 17/05/2017]

[16] Dipanjan, C. & Harry, C. (1999). "*Service Discovery in The Future for Mobile Commerce*", Article, ACM Crossroads, November 1999.

[17] Harry, C., Tim, F., Anupam, J., Dipanjan, C. & Liang, X. (2000). "*Service Discovery in the Future Electronic Market*": Article, Workshop on Knowledge-based Electronic Markets, AAAI-2000.

[18] Olga, V.R., Vladimir, K., Anupam, J., Tim, F. & Yelena, Y. (2001). "*Agents2go: An Infrastructure for Location-Dependent Service Discovery in The Mobile Electronic Commerce Environment"*: In Proceedings, ACM Mobile Commerce Workshop.

[19] Youyong, Z., Tin, F., Li, D., Harry, C. & Rong, P. (2003). "*Using Semantic Web Technology in Multi-Agent Systems: A Case Study in the TAGA Trading Agent Environment*", Proceedings of the 5th International Conference on Electronic Commerce, pp.1-7

[20] Yang, S., Qinpei, Z. & Qin, L. (2015). *Secure mobile Agent in eCommerce with Forward-Secure Detachable Digital Signatures*. ETRI Journal. Vol. 37, Issue 2, pp.1-12.
(IJNSA), Vol.4, No.2.

[21] Maes, P. (1994). *Agents that Reduce Work and Information Overload*. In: Communications of the ACM, 37(7), pp. 31-40.

[22] Kotz, D., Gray, R. & Rus, D. (2002). *Future Directions for Mobile Research*, Available at http//dsonline.computer.org/0208/f/kot.htm

[23] Harry, C. & Tim, F. (2002). "*Beyond Distributed AI, Agent Teamwork in Ubiquitous Computing*": Article, Workshop on Ubiquitous Agents on Embedded, Wearable, and Mobile Devices, AAMAS-2002.

[24] Lee, S. & Kim, K. (2012). *Mobile Agent based Framework for mobile ubiquitous application development*. Telecommunication Systems. Vol 51 Issue 2/3, pp.137-146.

[25] Krishman, I. & Zimmer, W. (1991). *Integrated Network Management*, No. II. (ISBN 0-444-89028-9). Elsevier Science Publishing Company, Incorporated.

[26] Chess, D., Grosof, B. & Harrison, J. (1995). "*Itinerant Agents for Mobile Computing*." Technical Report RC20010, IBM Research.

[27] Yashpal, S., Kapil, G. & Niranjan, S. (2012). *Dimensions and Issues of Mobile Agent Technology*. International Journal of Artificial Intelligence & Applications. Vol. 3, No 5, pp.51-61.

[28] Cai, T., Gloor, P.A. & Nog, S. (1996). *Dartflow: A workflow management system on the web using transportable agents.* Technical Report TR96-283, Department of Computer Science, Dartmouth College, Hanover, NH 03755.

[29] Ranganathan, M., Acharya, A., Sharma, S. & Saltz, J. (1996). "*Network aware Mobile Programs*", Dept. of Computer Science Tech. Report, University of Maryland, College Park.

[30] Bigus, P. & Bigus, J. (1998). *Constructing Intelligent Agents with Java*, John Wiley and Sons Ltd. England, ISBN: 0471-19135-3

[31] Dasgupta, P., Narasimhan., N., Moser, L.E. & Melliar-Smith, P.M. (1999). *Mobile Agents for Networked Electronic Trading*, 1999 IEEE

[32] Zhang, Q., Mu., Y., Zhang. & Deng, R.H. (2011). *Secure mobile Agents with controlled resources*. Concurrency & Computation: Practice & Experience. Vol.23, Issue 12, pp.1348– 1366.

[33] Sedgewick, R. & Wayne, K. (2011). *Algorithms*. Fourth edition. Addison-Wesley Professional, Massachusetts

[34] Jiang, Y., Liu, Y., Huang, W. & Huang, L. (2014). *Performance analysis of a Mobile Agent prototype system based on VIRGO P2P Protocol*. Concurrency & Computation: Practice & Experience. Vol.26, Issue 2, pp.447-461.

[35] Naylor, M., Buchman, W.J. & Scott, A. V. (2000). "*Enhancing Network Management Using Mobile Agents*," IEEE Transactions on Knowledge and Data Engineering, Vol. 12, No. 5, pp. 818-826.

[36] Bieszczad, A. (1998). *Mobile Agents for Network Management*. IEEE Communication Surveys http//www.coms oc.org Vol. 1, No 1.

[37] Satoh, I. (2003). *A Testing Framework for Mobile Computing Software*. IEEE Transaction on Software Engineering, Vol. 29, No 12 pp.1112 -1121

[38] Reddy, P. M. (2002). "*Mobile Agents: Intelligent Assistant on the Internet*", Resonance, 19pp.

[39] Singh, Y., Gulati, K. & Niranjan, S. (2012). "*Dimension and Issues of Mobile Agent Technology", International Journal of Artificial Intelligence & Applications* (IJAIA), Vol. 3, No. 5, pp.11

[40] Ebietomere, E.P. & Ekuobase, G.O. (2014). *Issues on Mobile Agent Technology Adoption*. African Journal of Computing and ICTs. Vol 7. No 1, pp 21-32

[41] Tarig, M.A. (2007). Increasing Mobile Agent Performance by using Free Areas Mechanism. Journal of Object Technology, Vol.6, No 4 pp. 125-140

[42] Akinyokun, O.C., Ekuewa, J.B. & Arekete, S.A. (2014). *Development of Agent-based system for monitoring software resources in a network environment*. Artificial Intelligence Research. Vol 3, No 3, pp. 62 - 74

[43] Amosa, B.M.G., Sobowale A.A., Adepoju Temilola, Hammed, M.A., and Onyeka Ndidi. (2017). An Agent Based System for Monitoring Loan Defaulters in Commercial Banks. *Academic Journal of Science. Vol., 7 No2 pp. 45-54.*

[1] [44] Strasser, M., Baumann, J. & Hole, F. (1997). *Mole: A Java Based Mobile Agent System*, Proceedings of Workshop on Mobile Object Systems, Lecture Notes in Computer Science (LNCS), Vol. 1222, Springer.