Embedded Mobile Agent (EMA) for Distributed Information Retrieval

Oguntunde, B.O
Department of Computer Science
Redeemer's University,
Ede, Osun State, Nigeria

Osofisan A.O
Department of Computer Science
University of Ibadan, Ibadan,
Oyo State, Nigeria.

Aderounmu, G.A
Department of Computer Science and Engineering
Obafemi Awolowo university,
Ile-ife, Osun State, Nigeria

***Abstract*— Mobile agent paradigm has been recognised as a viable approach for building distributed applications. Mobile agents migrate through the network, execute asynchronously and autonomously, conserve bandwidth, achieve better load balancing, adapt dynamically to changes in their environment, are robust and fault tolerant. Existing agents run and execute on agent platforms also called, the Mobile Agent System (MAS), which provides run-time execution and support facilitites for mobile agent to accomplish it tasks. These MASs from different vendors are different in laguage, design,and implementation and are not interoperateable, this impedes the achievement of the full potentials of mobile agent paradigm. This work is aimed at providing a robust structure for deploying mobile agents so they can execute independent of the MAS. We propose a lightweight agent to run in the kernel mode of the operating system as an operating system service, giving an impression of the agent directly communicating with the operating systems.**

.Keywords- embedded mobile agent, operating system service, lightweight agent, agent platform.

I. INTRODUCTION *(HEADING 1)*

Mobile agent paradigm has been recognized as a viable tool and a promising approach for building distributed applications [1, 2, 3] and a lot of research has been done, nevertheless, it is still a promising area of research, because, a lot of its many potentials are yet to be exploited. Agents solve complex software problems in distributed environments where protocols, operating systems, hardware and runtime environments are heterogeneous. Mobile agent has been defined as a computer entity capable of reasoning, use the network infrastructure to run in another remote site, search and gather the results, cooperate with other sites and return to its home site after completing the assigned tasks [4]. Mobile agents system provides infrastructure for executing automous agents and also migrate them between computers connected by a network.

Mobile agent paradigm was proposed as an alternative to client server [1], it offers flexibility on the reliance on network connection [2]. Once launched, can be disconnected, it keeps performing its tasks and can be reconnected to receive the result at a later time [5]. Mobile agents have the potentials to improve the speed and efficiency of computation by moving computation to data [6], thus eliminating unnecessary and massive data transfer over the network. According to [7], they are viable tools when information needed is vast and widely distributed, and in application or service that needs to learn and improve over time.

Mobile agent technology consists of mobile agents and mobile agent middleware also called platform (MAS). Mobile agent platform is a distributed execution environment for mobile agents [8], MAS provides services and primitives that help in the use, implementation and execution of system development using mobile agent paradigm [9]. Mobile agents run and execute on mobile agent platforms that provide run-time execution for mobile agents. The platforms are installed on the computers in the systems on which the agents are expected to run which consumes memory, increases access time and prevents other tasks from being run on the computer. There are many different agent platforms developed to support agent applications [10, 11, 12] and these platforms are not interoperateable, i.e an agent built on one platform cannot run on another platform. Mobile agents are naturally heterogeneous and should not be limited by agent platforms. This is one of the issues impeding the global acceptance of mobile agent paradigm as the absolute solution for the next generation distributed systems. There is therefore, a need for a unified system to run and execute mobile agents regardless of the design, platforms and vendor. Efforts made by the Foundation for Intelligent and Physical Agent (FIPA) to achieve interoperability among agents and Mobile Agent System Interoperability Framework (MASIF) to achieve interoperability among MASs [13] are yet to be fully achieved [14]. Furthermore, mobile agents have enormous potentials that are yet to be discovered, research in the past has focused mainly on the application of this technology to solve complex problems or improve certain solutions to sophisticated software problems, but little is being done on the improvement of the technology itself. Improving the mobile agent technology is the major focus of this work.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

In retrospect, mobile agent has been applied in many areas of research such as, information retrieval and management [15, 16], electronic commerce [17, 18], grid job scheduling [19], expert finding [20], network management [1, 21], traffic detection and management [22], examination system [23], supply chain management [24] and many more. A lot of issues arose with the use of mobile agents among which are security, complexity and lack of standard [25, 26]. The complexity and sophistication naturally led to many attempts to simplify and extend agent functionality, thus attention shifted to providing necessary security for mobile agents, agent platforms and hosts on which they execute [27]. The versatility of mobile agent paradigm also increased research interest in enhancing mobile agents in the area of agent communication [28] and agent structure [12], in order to extend their functionalities. It is on this note that we attempt to enhance the mode of deployment of mobile agents in order to make them execute without going through the agent platform. This work, attempts to eliminate agent platforms and make agent run as part of the hosts' operating system.

This work presents an embedded mobile agent that offers the possibility of mobile agents interacting directly with the operating systems on the host computers. The existing mobile agents require agent platforms to be previously installed on the computers on which they are to run; this platform needs to be explicitly initiated before receiving and providing runtime execution for incoming mobile agents. This work takes advantage of the fact that all computers run an operating system and attempts to make agents part of the operating system. The kernel mode of the operating system (specifically Windows Operating System) is extended with a static agent as a service in the kernel mode. The focus of this research work is to eliminate the agent platforms and make mobile agents run as part of Windows O/S in form of operating system service.

II. WINDOWS OPERATING SYSTEM SERVICE

Windows Operating system provides a way of making certain programs available to run as part of the operating system in the form of operating system service. Operating system Service is a long running executable program that runs in its own windows session, without user's intervention, this is similar to daemon in Unix operating systems [29].

The existing mobile agent for information retrieval model consists of mobile agents that execute on agent platforms previously installed on the computer machine. The platform is installed in memory on top of the operating system running on the host, this obviously consumes memory, and increases access time. The framework provided in this work includes a light weight agent embedded into the kernel mode of the operating systems to free memory and reduce access time. Figure 1 presents the proposed embedded agent as Windows XP operating system service. A light weight static agent that receives and executes mobile agents was embedded into the kernel mode of the Windows XP operating system as part of the executive services.

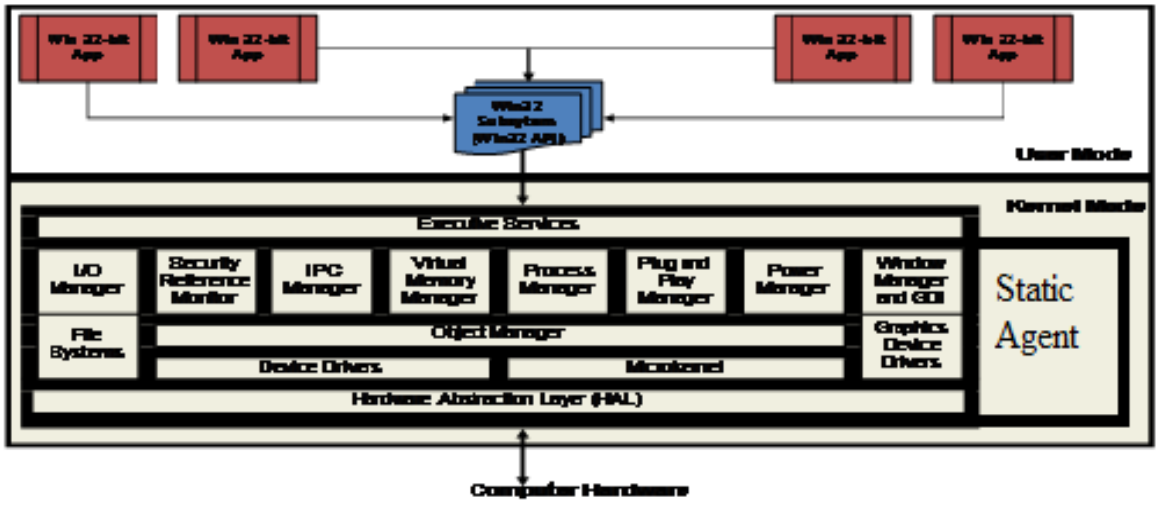

Figure 1: embedded agent as Windows XP operating system service (Adapted from 30)

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

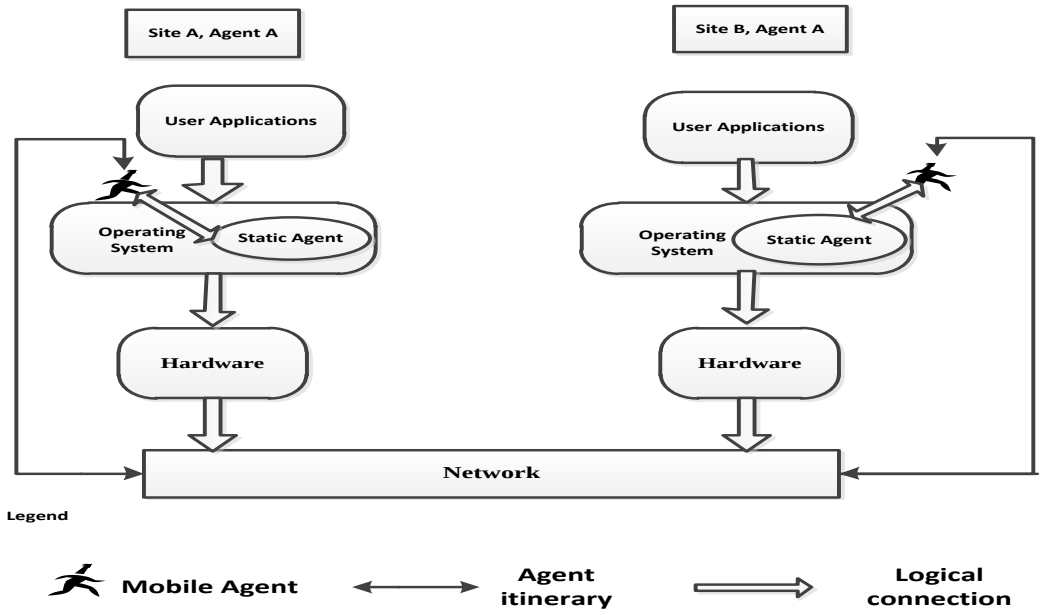

Figure 3: the conceptual architecture of the proposed system.

The embedded agent was designed using the layered architecture, such as the three layer architecture designed by [24]. Each layer represents a particular function. A mobile agent contains code, state information and attributes. The attributes of mobile agent include its name which is unique for identification, the authority or the owner of the agent and the agent system type. Code contains the logic of the agent, i.e. code defines the behavior or the required tasks of the agent, same type of agents use same code. Code in an object oriented context means the class code necessary for an agent execution [31]. Data corresponds to the value of the agent's instance variables and include information about the mobile agent such as its launcher, movement history, resource requirements and authentication keys for use by the infrastructure; these are referred to as the initial data. The data also include results of the mobile agent's tasks on different nodes visited, referred to as the generated or received data.

The enhancement is a static agent embedded and made to run in the kernel mode of the operating system as an Operating System Service. The static agent receives and provides execution environment for visiting mobile agents as depicted in figure 2, this gives the impression of mobile agents communicating directly with the operating system. The target operating system is the Windows OS (Windows XP, Windows vista and Windows 7). The mobile agent class has attributes execute state, communicate, record and the list of tasks to perform. Figure 2 shows the conceptual architecture of the proposed system.

III. IMPLEMENTATION

To create a customised service involves setting up the inheritance and other infrastructure elements. The static agent class inherits from the ServiceBase class and a main method defines the service to run, the following implements the static agent

```
public class StaticAgent implements Runnable{
    private Socket insoc;
    private ObjectInputStream ois;
    public static int AGENTS_PORT = 4999;
    public StaticAgent(Socket insoc) throws IOException
    {
        this.insoc = insoc;
```


http://sites.google.com/site/ijcsis/
ISSN 1947-5500

```
        // ……..;
    }
    public void run() {
        try{
                MobileAgent magent =
(MobileAgent) ois.readObject();
//                ……..
        System.out.println(ex);
             }finally{
                try{
            }catch(Exception x){}
         }
    }.
```

We evaluate the proposed Embedded Mobile Agent (EMA) architecture in distributed information retrieval environment, Let's consider the situation at the meteorological agency with headquarters in Abuja, Nigeria and three other branches each representing one geopolitical zone, say, Lagos for southwest, Port-Harcourt for Eastern Zone and Kano for the northern Zone out of the six geopolitical zones in Nigeria. Each branch agency runs an embedded agent as part of their operating system, provides execution environment with heterogeneous hardware configurations and versions of Windows operating system (Windows XP, Windows Vista and Windows 7).

using the case of retrieving weather information. The information was stored in databases that are distributed geographically and connected by a network. Four hypothetical locations were chosen and connected together by a network as depicted by figure 3, the Mobile Agent was sent from one location given the itinerary and it visited the other nodes collected the required information and returned to the origin. On each host, the static agent was installed and ran continuously, without users intervention, it is lunched automatically once the computer boots. The static agent listens to the port for incoming mobile agents, negotiates passage to the destination host, validates and authenticates the incoming agent, launches the received mobile agent and provides runtime execution for the agent. The mobile agent migrates through the network, negotiates access with the static agent on remote host, downloads the required information and adds it to its bag. It then determines the next node to visit, initiates a move to the next node, returns to the origin with the result of the search and disposes itself.

The director at the headquarters in Abuja (origin) requires the weather conditions (temperature and atmospheric conditions) from the different stations in each zone.
The EMA is dispatched from the origin (Abuja) given the itinerary (names or IP addresses) of the nodes to visits. The EMA migrates to first node in its itinerary, retrieves the required information and migrates to the next location in its itinerary, repeats the same process and to the next performs its function and later returns to the origin with the result of the search. To retrieve information on each host, the EMA performs SQL queries on relational databases which contain records in a table with structure described by figure 4.

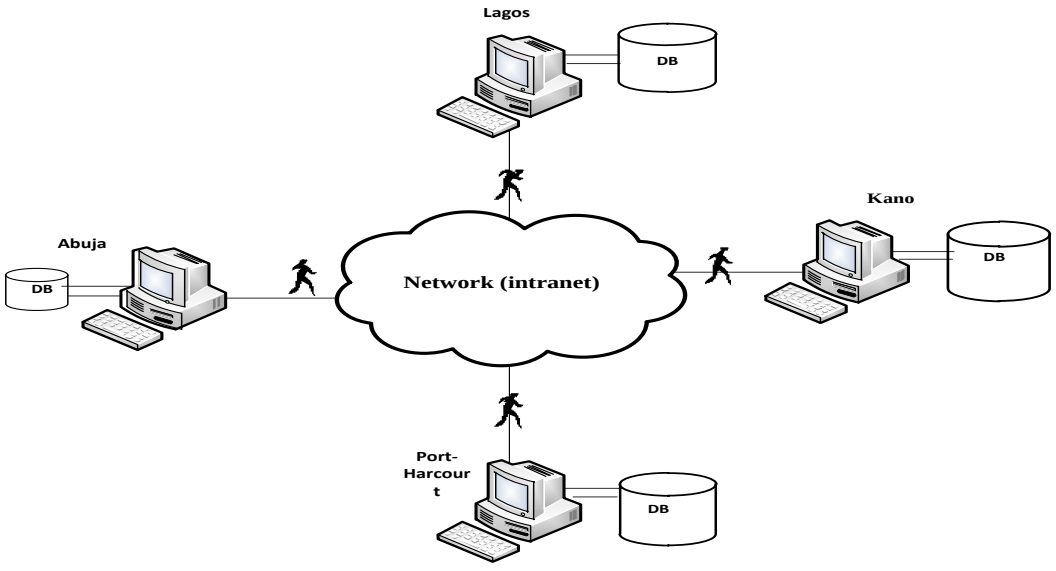

Figure 3: Overall architecture of the proposed system

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

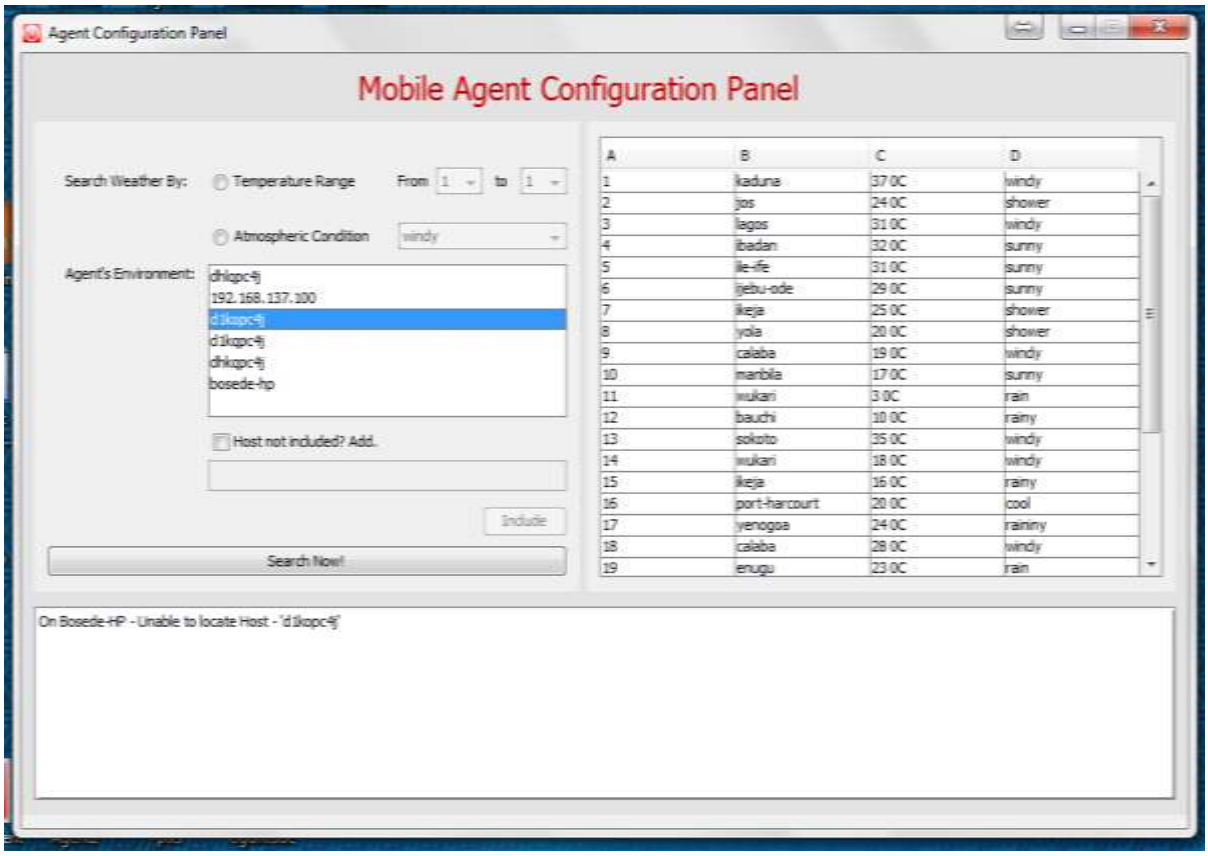

Figure 4: Search sample with the EMA

Experimental results

EMA was deployed into an existing local area network and its performance was compared with that of existing Java Agent Development Framework (JADE). The EMA proves to be a more efficient and autonomous scheme with a high level of flexibility compared to JADE. It reduces memory consumption, reduces access time, is robust and fault tolerant.

Conclusion

In this work, we present Embedded Mobile Agent (EMA) architecture, based on the common mobile agent structure with an additional feature added. The system provided by [3] also focussed on enhancing the structure of mobile agent to improve future performance while executing on agent platform. Whereas, the system presented in this work was designed to be embedded in the Windows Operating System in the form of Windows service to enable mobile agents directly interacts with the Operating System. The implementation of the Embedded Mobile Agent was achieved and deployed into an existing Local Area Network to retrieve information from remote distributed databases. The potentials of mobile agents can be extended by eliminating the barrier placed by agent platforms.

In the area of future research, implementation with other operating systems such as Unix or any of its flavours will be investigated and the EMA will be applied to more sophisticated distributed operations. In addition, we intend to provide adequate security for the Embedded Mobile Agent.

References

[1] G.A. Aderounmu, "Development of an intelligent mobile agent for computer network performance management," Ph.D dissertation, Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria, 2001

[2] G.A. Aderounmu, B.O. Oyatokun and M.O. Adigun. Remote Method Invocation and Mobile Agent: a Comparative Analysis. *Issues in Informing Science and Information Technology,* vol. 3, 2006, Available: http://informingscience.org/proceedings.INSITE2006/IISTAder188.pdf

[3] S. Gabriel and I. P. Claudiu, "A Proposal for an Enhanced Mobile Agent Architecture (EMA)". *Annals of the University of Craiova, Mathematics and Computer Science Series. 37(1): 71 – 79, 2009*

[4] O. Abdelkader, "Mobile Agent-Based Applications: A Survey," *International Journal of Computer Science and Network Security, 331 – 339, 2009.*

[5] J. Dale and D.C. DeRoure, "A Mobile Agent Architecture for Distributed Information Management", *in proc. 1997 International workshop on the virtual Multicomputer, 1997.*

[6] G.P. Picco,. "Understanding code mobility". Technical report, Dipartimento di Elettronica Informazione, Politecnico di Milano, 2005. [date of last access: 18 June, 2013]

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

[7] I. Sridhar, and J. Vikram, "Designing Distributed Applications using Mobile Agent," in pro*c. 2001 International Conference on High Performance Computing, 2001. Hyderabad, India.*

[8] S. Dilyana and G. Petya, "Building Distributed Applications with Java Mobile Agent", presented at the International workshop NGNT, 2002.

[9] S. S. F. Roberto, "The Mobile Agents Paradigm," research paper, department of Information and Computer Science, University of Califonia, Irvine, 2001.

[10] A. Syed, D. John and Y. Pavana, "A survey of mobile agent systems," Student Report, Department of Computer Science and Engineering, University of California San Diego, 2000. (Date of last access: 03 July, 2013 available cseweb.ucsd.edu/classes/sp00/cse221/reports/dat-yal-and.pdf).

[11] G. J. van 't Noordende, B. J. Overeinder, R. J Timmer, F. M. T. Brazier and A. S. Tanenbaum, 2009. "*Constructing secure mobile agent systems using the agent operating system"*, International Journal of Intelligent Information and Database Systems, Vol. 3, No. 4,

[12] Bellifemine, F.L, Greenwood D and Caire G. 2007. *Developing Multi-agent systems with JADE.* John Wiley & Sons Ltd, England.

[13] Roth V., Pinsdof U. and Binder W. Mobile agent interoperability revisited. Available: citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.1814&rep=rep1&type=pdf

[14] Clark K. L and Lazarou V. S. 1997. A Multi-Agent System for Distributed Information retrieval on the World Wide Web.

[15] Htoon H. and Mie M. T. T. 2008. Mobile Agent for Distributed Information Retrieval System. *Proc. 2008 ECTI-CON.*

[16] Brian B., Robert G., Katsuhiro M., David k., George C. and Daneila R. 1999. Mobile Agent in Distributed Information Retrieval. Technical report submitted to Thayer School of Engineering, department of Computer Science Dartmouth College Hanover, new Hampshire. Firstname.lastname@dartmouth.edu.

[17] Christoph B., Volker R. and Ralph M. 2002. Perspectives on Electronic Commerce with Mobile Agents.

[18] Enock M. 2005. A Development of Resource/Commander Agents Used in AgentTeamwork Grid Computing Middleware.

[19] Iyilade, J.O, Aderounmu G.A and Adigun, M.O. 2005. An Agent-based Approach for finding a Supervisor in an Academic Environment. *Proceedings of the 3rd International Conference on Education and Information Systems: Technologies and Application. Orlando, Florida, USA, 363–381.*

[20] Bohoris C. (2003). Network Performance Management Using Mobile Software Agents. Ph. D thesis, University of Surrey, UK.

[21] Bo Chen, Henry H. C. and Joe P. 2009. Integrating mobile agent technology with multi-agent systems for distributed traffic detection and management systems. Transportation Research Part C, pp 1-10.

[22] Gawali, R.D and Meshram, B.B. 2009. Agent-Based Autonomous Examination Systems. *Proceedings of the Intelligent Agent and Multi-agent systems, 2009, (IAMA 2009) International Conference.*

[23] Wenjuan W., Tong Li, Weidong Z. and Weihui D. 2009. Mobile agent system for supply chain management. *Proceedings of the second symposium of International Computer Science and Computational technology (ISCSCT'09), Huangshan, P.R China, 525-528.*

[24] Fortino G. and Russo W. 2003. High-level interoperability between java-based mobile agent systems. A report of the project ' Giovane Ricercatore 2003', University of Calabria.

[25] Tudor M., Bogdan D., Mihaela D. and Ioan S.. A Framework of Reusable Structures for Mobile agent Development. *Proceedings of the 8th IEEE international Conference on Intelligent Engineering Systems (INES '04)*, Cluj-Napoca, 2004.

[26] Biermann E. 2004. A Framework for the Protection of Mobile Agents Against Malicious Hosts. A Ph.D thesis, University of South Africa, South Africa.

[27] Priya, B.G; Suba, S.; Bensal, T.; and Boominathan, P. 2009. Enhanced Communication Scheme for Mobile Agent, *Proceedings of the Intelligent Agent and Multi-agent systems, 2009, (IAMA 2009) International Conference.*

[28] Wikipedia 2013. Windows Service. Available: www.en.wikipedia.org/wiki/Windows_service. [date of las access: 24 July, 2013]

[29] WIN133. (2009). Introduction to Windows XP architecture. Available: www.warrenworks.com/csc_110/SupportFiles/w2karchitecture.ppt. [Retrieved on 20 *September*, 2012.

[30] Lange D. B. And Oshima M. 1999. Seven Good Reasons for Mobile Agents. *Communication of the ACM, .42(3).*

AUTHORS PROFILE

Oguntunde, B.O holds a B.Tech (Hons) degree in Computer Engineering from LAUTECH, Ogbomosho, Nigeria, in 2000. She obtained her M.Sc and PhD in computer Science from the University of Ibadan. She is currently teaches at the Redeemer's University, Nigeria. Her research interests are in the areas of computer communication and deployment of mobile agent software management of heterogeneous computer network.

Osofisan, A.O. obtained a B.Sc (Hons) degree in Computer Science from the Obafemi Awolowo University, Ile Ife, M.Sc from Georgia Tech. and PhD from Obafemi Awolowo University. She is currently a Professor and the director of Business School, The University of Ibadan where she also lectures in the department of Computer science. Her areas of specializations are data communications, data warehousing and data mining. She has many articles in these areas at both local and international level to her credit.

Aderounmu G.A. obtained a obtained a B.Sc. (Hons) degree in Computer Engineering from the Obafemi Awolowo University, Ile-Ife, Nigeria, in 1991. He obtained his M.Sc and Ph.D in Computer Science from the same University in 1997 and 2001 respectively. He lectures in the department of Computer Science and Engineering; he is presently a professor and the director of the Information Technology and Communication Unit (INTECU) of the same University. His areas of specializations are design, analysis, and simulation of ATM networks with respect to switching, protocol, and buffer management and mobile agent software development. He has many articles at both local and international level to his credit.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500