INTERNATIONAL JOURNAL OF NATURAL AND APPLIED SCIENCES (IJNAS), VOL. 4, NOS.1& 2 (2009); P. 87 – 92, 2 FIGS.

A survey of mobile agent systems and paradigm

E. E. Williams*[1], D. E. Asuquo[2], and E. N. Effanga[3]

ABSTRACT

With the passage of time, computers have become more and more a part of everyday life as they become smaller and more powerful. Computer systems and users are becoming increasingly interactive and the internet, in particular, is a viable platform for interactive computing. The internet, no doubt, holds huge mass of information. Unfortunately, access to "desired" information is often times difficult due to poor human-computer interaction. One of the best means for getting services or finding information of a particular interest on a network requires the combined use of two softwares, mobile agents and an interface, to connect users and resources in a transparent, open, and scalable way. Mobile agents are autonomous programs that move on the network on behalf of their owners while searching for information. On the other hand, interfaces are dividing lines between systems. Together they make the usage of computer systems, efficient, supportive and pleasant. This work surveys the state of the art of mobile agents.

.

INTRODUCTION

As the Internet constantly expands, the amount of available on-line information expands as well (Kotz and Gray, 1999). One problem that needs solution is how users can efficiently find, gather, and retrieve this information. One possible solution to help in this situation consists of the use of 'mobile agents' - autonomous programs that move about the network on behalf of their owners while searching for information or even negotiating with other agents. Mobile Agents (MA) are processes (i.e., executing programs) that can migrate from one machine of a system to another machine (usually in the same system) in order to satisfy requests made by their clients (Johansen *et al*, 1995). The term '*agent*' was coined in the sphere of Artificial Intelligence (AI) although in more recent years the term has spread into the computing mainstream to become a significant and generic computing technology. In this context similar definitions of agents are presented in Wooldridge and Jennings (1995) as a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives." Subsequent interest in agent technology has exploded and agents for all purposes have been developed, including e-technologies, data-mining (Mitkas et al, 2003), control systems (Wooldridge, 2002) and many others. Each agent acts autonomously and independent of others, although in multiagent systems such agents can collaborate to perform actions beyond the scope of their individual capabilities. This can be seen in a number of scenarios presented by Sheldon et. al. (2004), with reference to command and control systems, data fusion and data analysis amongst others.

Basically, a mobile agent executes on a machine that hopefully provides the resource or service that it needs to perform its job. If the machine does not contain the needed resource/service, or if the mobile agent requires a different resource/service on another machine, the state information of the mobile agent is somehow saved, transfer of the mobile agent to the machine containing the necessary resource/service is initiated, and the mobile agent resumes execution at the new machine.

CHARACTERISTICS OF MOBILE AGENTS

A number of characteristics have been identified in Object Management Group (OMG, 2000). These include:

- **Mobility:** the core property in a mobile agent concept whereby the agent has the ability to migrate or transport itself from one node to another within the same environment or from node to another node in a different environment autonomously.
- **Proxy:** MAs may act on behalf of someone or for the benefit of some entities (e.g. software systems). In order to act on behalf of others, mobile agents must have at least a minimal degree of autonomy.
- **Proactive:** MA should be a goal-oriented entity, and take an initiative in responding to an environment.
- **Intelligent:** MAs may have certain degree of intelligence, base on knowledge in order to act efficiency.
- **Coordinative:** MAs should be able to perform data transfer activities in sharing with other agents within the given environment.

*Corresponding Author: edemwilliam@yahoo.com

Manuscript received by the Editor August 4, 2008; revised manuscript accepted July 21, 2009.

[1]Department of Mathematics/Statistic & Computer. Science, University of Calabar, Calabar, Nigeria
[2]Department of Mathematics/Statistics/Computer Science ,University of Uyo, Uyo, Nigeria
[3] Federal Ministry of Internal Affairs, Nigerian Prisons Service, Cross River State Command, PMB 1055, Calabar.

- **Learning:** refers to a mobile agent's ability in gaining information about the current environment, which will help MAs to modify its behaviour.
- **Cooperative:** MAs should be able to coordinate with other agents to achieve a common purpose.
- **Ragged:** MAs should be able to deal with errors when encountered and during their occurrence.

Researchers have also attempted to classify the types of the agent based on the above attributes. In many cases, it is not necessary to have an agent with all the above properties. This will be determined by the nature of the task which the agent should achieve. For example, Microsoft's software agent does not necessarily need to be mobile and cooperative but it must be smart and proactive. "This agent proposed to enhance the user interface of their applications with interactive personalities in the form of animated characters. These characters can move freely within the computer Clients display, speak aloud (and by displaying text onscreen), and even listen for spoken voice commands". For this work the mobility factor must exist as the main feature alongside one or many of the previous attributes depends on the need of that characteristic.

WHY MOBILE AGENT TECHNOLOGY

The nature of mobile agents can lead to many benefits. MAs can be used in areas, such as network management, mobile computing, information management, web services, remote software management and others (Lange & Oshima, 1999). Mobile agents can increase performance in these areas by achieving the following tasks:

- **Reduce the network load**, MAs allow users to package and migrate their operations to be carried out locally. This is useful where huge volumes of data are required to be processed; the data will be manipulated locally rather than transferred over the network. In other words, move the computation to the data rather than the data to the computation.
- **Overcome network latency**, MAs can offer great opportunities to respond to real-time systems where delay is not acceptable due to its mobile and adaptive properties.
- **Encapsulates protocols**, in distributed systems where many different protocols may be used to communicate among the network nodes, MAs can overcome this problem by moving to remote hosts to establish channels based on proprietary protocols. Execute asynchronously and synchronously, in a mobile computing arena where network connection is expensive (small bandwidth). Some tasks require open connection between mobile devices and fixed networks; this might not be economical and practical. Therefore, MAs can solve this problem by embedding a mobile task into the mobile agent and dispatch the mobile agent to the destination and disconnect the link between the mobile device and the fixed network. MAs now can operate asynchronously and autonomously. However, when the task is finished, MAs can reconnect and return to the mobile device with the results (Lange & Oshima, 1998).
- **Heterogeneous**, computer networks are heterogeneous in nature from software and hardware perspectives. MAs can work in this kind of environment perfectly due to their independence property. MAs are only depending on their execution environment.
- **Robust and fault-tolerant**, the events-based model makes MAs react dynamically to any critical situation within the environment. This makes it easier to build robust distributed systems based on mobile agent technology.

This emerging technology attracts a lot of interest from the fields of distributed systems, information retrieval, electronic commerce and artificial intelligence. The emergence of Java, with its support for mobile code, led to heightened research activity in this area. Java is the language of choice for mobile agent systems such as Concordia, Mole, JADE, Odyssey, Aglets, Tracy and Voyager. Java also supports development of mobile agents that are tightly integrated with the Web (Wong *et al*, 1999).

In the field of information retrieval, the different component requirements of mobile agent technology plays a prominent role. The agent's user interface in particular is used for displaying the status of the platform, in order to have a view of the agents/containers on the platform. Such interface with its characteristics Agent Transfer Protocol (ATP) serves as a medium for enhancing agent mobility. Generally, interfaces have ways of giving input, based on which the response is created dynamically. It may be seen as a form of software which accepts input from humans. A simple example is a search engine such as Google: first the user types in search words and sends it to the server. The server does a database lookup and returns matching data although the internal structure of the search engine may be very complex.

To be useful, an agent needs to be mobile, interacting with its host system (local or remote) so as to access or retrieve information. Basic requirements for use in agent technology include mobile agent code, agent executing environment (or agent platform) and the mobile agent creator, usually the agent's Graphical User Interface (GUI) at users' end (Aneiba and Rees, 2005). This User Interface (UI) is a tool meant for the management and control of mobile agents in terms of agent mobility. But the real concern, apart from movement of agents, would also be the format of displaying to users the information retrieved from the server. The agent's GUI lacks this functionality and moreover they tend to use up more system resources like memory.

Considering the use of mobile agent in web application, a more robust user interface is needed to process information on the web as well as improve the mode of displaying retrieved information.

THE INTERNET AND INTERACTIVE COMPUTING

Computer networks consist of computing nodes connected via communication channels. The development of computer networks has been driven by the need to exchange information between computing nodes or between nodes and shared resources like servers and I/O devices (Rothermel *et al*, 1999). The Internet (network of computers) is a huge mass of information. The development of the Web was driven by a desire for a simple and effective method of searching through this wealth of information for particular ideas or interests. Without doubt, the explosive growth of the Internet has been driven by the enormous popularity of the World Wide Web (WWW). The Web's continuing success is attributable to the simplicity with which it provides users an ability to locate and retrieve dispersed information from within the Internet. A major part of this success is due to the effectiveness of the hypertext transfer protocol (HTTP) through the Web’s architecture and format of storing/referencing information.

The computer systems are fast becoming interactive. What users get out of a computer system depends much on how well they are able to understand what they see or hear within the system. Moreso, when resources are becoming too numerous to handle. A means of improving interaction between the user and the computer is through an interface (Preece, 1994). Berners-Lee (2005) defined Unified Resource Locator (URL) and Unified Resource Identificator (URI) as addressable standards that describe resources on the web. In standard web operation, the browser sends a request containing the resource URI to the server. The server then interprets the URI and provides appropriate response. In a simple case, the URI defines a path to the desired HTML file. In many cases, the URIs does not actually refer to any distinct file but to a program creating the desired response on-the-fly utilizing databases. On the other hand, the URI is an important tool for finding information, especially by means of saving the addresses for later use.

Kotz and Gray (1999) had envisaged that within few years, nearly all major Internet sites will be capable of hosting or willing to host some form of mobile code or mobile agents. The network and computer technology, coupled with the exponential growth of the services and information available on the Internet, will soon bring us to a point where hundreds of thousands of people will have fast, pervasive access to a phenomenal amount of information from anywhere and everywhere. Mobile code, and in particular mobile agents, will be an essential tool allowing such access.

Mobile agent technology has been employed in many areas from network management tasks to information management. Dasgupta et al (1999) has produced several mobile agent applications that can be applied in electronic trading. For example, a mobile agent application is dealing with information retrieval in the m-commerce field has been developed. MAs have been used in order to retrieve information regarding the cheapest offer for a particular item over the Internet. In this application, MA has been issued by a Personal Digital Assistant (PDA) user using an application installed in the PDA. MA roams the Internet leaping from one node to another seeking and comparing the prices. When the mobile agent finds the cheapest offer, it will return to the source node with the desired results.

The use of mobile agents in the wireless environment means that the application should support disconnected mode. For example, when the mobile agent is sent from a PDA to the network, the PDA could be disconnected and later when the connection is re-established the mobile agent with the desired results can migrate back to the source (Wang *et al*, 2003). This feature gives the mobile agent a significant advantage over other communication paradigms (e.g. Client-Server, RPC). Further developments have been done using mobile agent technology on some application domains such as network management, data warehouses, software updates, and information management tasks such as searching for information, information filtering, information monitoring and wireless network tasks.

MOBILE AGENT PARADIGM

Mobile agents have been developed as an extension to and replacement of the client-server model (Gray, 1995). In the client-server model, a server is a machine that provides some service (or a set of services) and a client (most often another machine) makes request from those services. Communication between client and server is usually through message passing. So, when a client needs a particular service, it usually sends a request message to a server that contains the needed service. A limitation of the client-server model is that the client is limited to the operations provided at the server (Gray, 1995). If a client needs a service that a particular server does not provide, the client must find a server that can satisfy the request by sending out messages to all servers. This clearly is an inefficient use of network bandwidth. Also, this severely limits network scalability since managing and updating these servers would prove prohibitive.

Mobile agents are programs that can migrate from host to host in a network, at times and to places of their own choice. The mobile agent approach shown in figure 1 is a relatively new paradigm in the distributed systems environment (Kotz and Gray, 1999). The agents migrate from client to server in a network where the state of the running program is saved, transported to the new host, and restored, allowing the program to continue where it left off. Mobile agent systems differ from process-migration systems in that the agents move when they choose, typically through a “jump" or “go" statement, whereas in a process-migration system the system decides when and where to move the running process (typically to balance CPU load). Mobile agents differ from “applets”, which are programs downloaded as the result of a user action, then executed from beginning to end on one host.

The term ‘*agent*’ was coined in the sphere of Artificial Intelligence (AI) although in more recent years the term has spread into the computing mainstream to become a significant and generic computing technology. In this context similar definitions of agents are presented in Wooldridge and Jennings (1995) as a computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives.” Subsequent interest in agent technology has exploded and agents for all purposes have been developed, including e-technologies, data-mining (Mitkas *et al*, 2003), control systems (Wooldridge, 2002) and many others. Each agent acts autonomously and independent to others although in multi-agent systems such agents can collaborate to perform actions beyond the scope of their individual capabilities. This can be seen in a number of scenarios presented by Sheldon *et al* (2004) with reference to command and control systems, data fusion and data analysis amongst others.

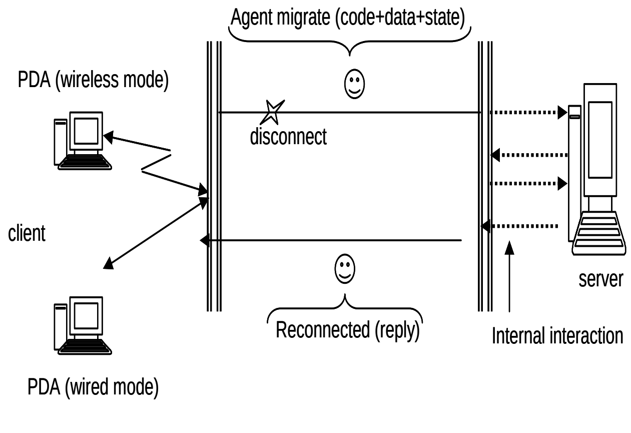

Fig. 1. Mobile agent paradigm

Several types of agent exist such as collaborative agents, interface agents, mobile agents, information agents, reactive agents, hybrid agents and smart agents. These types of agent may either be static or mobile in nature and built based on their purpose. Static agents tend to be resource-intensive and possess some form of intelligence or decision support, the focus is cooperation whereby multiple static agents combine to perform a task. Mobile agents on the other hand are lightweight and compact and are designed to migrate from host to host. The rationale behind such agents is the ability to perform tasks at source via executing the mobile-code, this allows for minimal bandwidth usage in resource-intensive computations and potential performance balancing.

Mobile agents are one form of mobile code. In its simplest form, the concept of mobile code involves the dynamic installation of code on a remote host. In Web applications, applets and servlets are a common form of mobile code. The mobile code concept also appears in “remote evaluation” systems (Stamos and Gifford, 1990), which extend the notion of remote procedure call to transport the procedure to the server along with the call. Many researchers extend the mobile-code concept to “mobile objects,” in which an object (code and data) are moved from one host to another. The mobile-agent abstraction extends this notion further, by moving code, data, and a thread from one host to another. A mobile agent runs in one location, moves (with its state) to another host, and continues at that host. While mobile code and mobile objects are normally moved by an external entity, mobile agents usually have migration autonomy.

Traditionally, distributive applications have relied on the client-server paradigm, in which client and server processes communicate either through message passing or Remote Procedural Call (RPC). This communications model is usually synchronous, i.e., the client suspends itself after sending a request to the server, waiting for the result of the call. An alternative architecture called Remote Evaluation (REV) was proposed by Stamos and Gifford (1990) in Karnik and Tripathi (1998). In REV, the client, instead of remote procedure, sends its own procedure code to a server, and request the server to execute it and return the result.

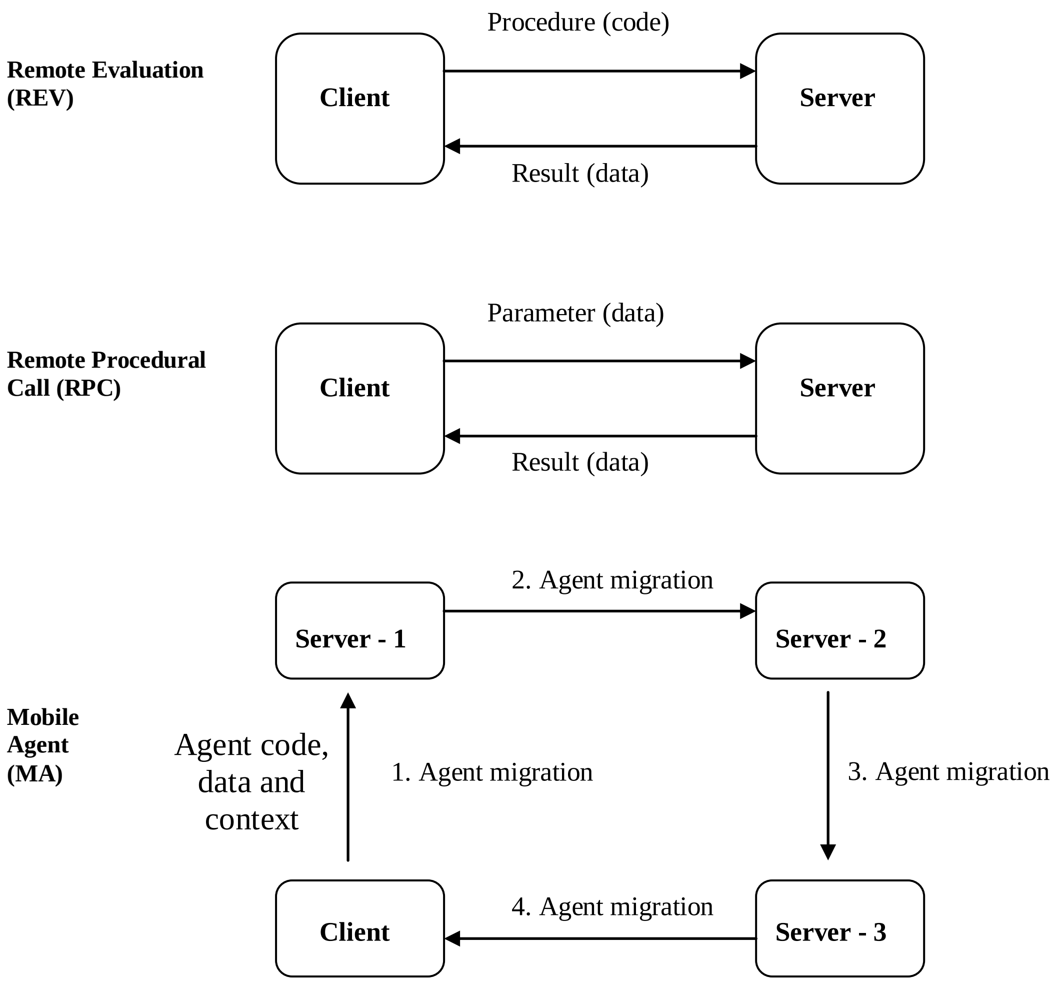

Fig. 2. Evolution of the mobile agent paradigm

The mobile agent paradigm has evolved from these antecedents. The diagram in figure 2 shows how it differs from RPC and REV. In RPC, data is transmitted between the client and server in both directions. In REV, code is sent from the client to the server, and data is returned. In contrast, a mobile agent is a program (encapsulating code, data and context) sent by a client to a server. Unlike a procedural call, it does not have to return its result to the client. It could migrate back to the client if appropriate. It thus has more autonomy than a simple procedure call.

Telescript (White, 1995), which was developed by General Magic in the early 1990s, was the first system expressly designed to support mobile agents in commercial applications. It was followed by several research systems such as Tacoma and Agent Tcl, in which agents are written using script languages. The emergence of Java, with its support for mobile code, led to heightened research activity in this area. Aglets, Voyager, Concordia and Ajanta are examples of Java-based mobile agent systems.

CONCLUSION

The nature of mobile agents can lead to many benefits. MAs can be used in areas, such as network management, mobile computing, information management, web services, remote software management and others as summarized by (Lange and Oshima, 1999). They can increase performance in these areas by reducing network load (Bigus and Bigus, 1998), overcoming network latency, encapsulating protocols in heterogeneous distributed systems and making them robust and fault-tolerant, executing asynchronously and synchronously, operate asynchronously and autonomously (Lange and Oshima, 1998).

REFERENCES

Aneiba, A. and Rees, S. J.(2005).Mobile agents technology and mobility.URL :http://www.soc.staffs.ac.uk/aa11/Mobile_Agents_Tech_and_mobility.pdf

Berners-Lee, T. (2005). Uniform resource identifier (URI): Generic Syntax. Retrieved March 12, 2008 from ftp://ftp.rfc-editor.org/in-notes/rfc3986.txt.

Bigus, P. and Bigus, J. (1998). *Constructing Intelligent Agents with Java*. John Wiley and Sons Ltd, England. 135pp.

Dasgupta, P., Narasimhan, N., Moser, E. and Melliar-Smith, P. M. (1999). Mobile agents for networked electronic trading, IEEE Concurrency.

Gray, R. S. (1995). Agent Tcl: A transportable agent system. *Proceedings of the CIKM workshop on Intelligent Information Agents*. Fourth International Conference on Information and Knowledge Management (CIKM 95), Baltimore, Maryland.

Gray, R., Kotz, D. and Peterson, R.A. (2001). Mobile-agent versus client/server performance: scalability in an information-retrieval task. *Technical Report*, TR2001-386. Retrieved July 20, 2005 from ftp://ftp.cs.dartmouth.edu/TR/TR2001-386.ps.Z

Johansen, D., Renesse, R. V. and Schneider, F. B. (1995). Operating system support for mobile agents, *Proc. of the 5th IEEE* HOTOS Workshop, Orcas Island, USA.

Karnik, N. M. and Tripathi, A. R. (1998). Design issues in mobile agent programming systems. IEEE Concurrency, 6(6): 52–61.

Kotz, D. and Gray, R. S. (1999). Mobile agents and the future of the internet, ACM Operating Systems Review 33(3) :7-13.

Lange, D. and Oshima, M. (1998). Programming and deploying Java mobile agents with aglets. Addison-Wesley,

Lange, D., and Oshima, M. (1999). Seven good reasons for mobile agents.Communications of the ACM, No. 43(3) : 88–99.

Mitkas, P., Symeonidis, A., Kehagis, D and Athanasiadis, I. (2003). Application of data mining and intelligent agent technologies to concurrent engineering. In: Proceedings of the 10th International Conference on Concurrent Engineering (CE-2003), Madera, Portugal.

Morreale, P. (1998). Agents on the move. Spectrum, IEEE, Volume. 35, No. 4: 34-41.OMG (2000). Agent Technology. Green paper, Technical Report ec#2000-03-01. Object Management Group.URL: http://www.jamesodell.com/ec2000-08-01.pdf

Preece, J. (1994). *Human-Computer Interaction*. Addison-Wesley, New York.

Rothermel, K., Hohl, F. and Radouniklis, N. (1999). Mobile agent systems: what is missing? Institute for Parallel and Distributed High-Performance Systems (IPVR) Breitwiesenstraße 20-22, D-70565 Stuttgart, Germany. Sheldon, F., Potok, T. and Kavi, K. (2004). Multi-agent system: case studies in command and control, information fusion and data management. *Journal of Informatica.* 28:1-12

Stamos, J. W. and Gifford, D.K. (1990). *Remote Evaluation*. ACM Transaction on Programming Languages and Systems, 12: 4.

Wang, A., Sørensen, C. and Indal, E. (2003). A mobile agent architecture for heterogeneous Devices. URL: http://www.idi.ntnu.no/grupper/su/publ/alfw/woc2003-mobile-agent.pdf

White, J. E. (1995). Telescript Technology. Mobile Agents, General magic white paper, URL: http://www.genmagic.com/.

Wong, D., Paciorek, N. and Moore, D. (1999). Java-based mobile agents. Communications of the ACM,42(3) :92-105.

Wooldridge, M. (2002). An introduction to multi-agent systems, John Wiley and Sons Ltd., England,691pp.

Wooldridge, M. and Jennings, N. R. (1995). Intelligent agents: Theory and Practice, *Knowledge Eng. Rev.*, 10: 2.