Volume 2, No. 11, November 2011

Journal of Global Research in Computer Science

ISSN-2229-371X

RESEARCH PAPER

Available Online at www.jgrcs.info

ANALYSIS OF MOBILE AGENT

Lect. Prakash V. Rajguru*[1,] Dr. Sushant B.Deshmukh[2]
[*1] Department of Computer Science, Adarsh College, Hingoli (Maharashtra), India
prakash_rajgure@yahoo.com, rajguru_phd@yahoo.com
[2] Post Graduate Department of Physics & Computer Science, R.G.Bagdia Arts, S.B.Lakhotiya Commerce. & R.Bezonji Science College, Jalna (Maharashtra), India
sushant.D69@gmail.com

Abstract: Mobile agents are considered a very interesting technology to develop applications for mobile, pervasive, and distributed computing. Thus, they present a combination of unique features, such as their autonomy and capability to move to remote computers to process data there and save remote communications. Many mobile agent platforms have been developed since the late nineties. While some of them have been outdated, others continue releasing new versions that fix bugs detected or offer new interesting features. Moreover, other new platforms have appeared in the last few years. So, a common problem when one wants to benefit from mobile agent technology to develop distributed applications is the decision about which platform to use. In this research paper, we provide an up-to-date evaluation of existing Mobile Agent platforms.

***Keyword*:** Mobile Agent, Mobile code, Distributed, Interactions.

INTRODUCTION

In the past few years there has been an growth of interest in mobile agent technology and several platforms have been developed. Some of them have only been used for research purposes while others have been deployed as commercial products. The society is now looking for applications where these platforms can be effectively used. Some comparisons about the functionality of some mobile agent systems have been presented in this research paper

A Mobile Agent is an emerging technology that is gaining momentum in the field of distributed computing. The use of mobile agents can bring some interesting advantages when compared with traditional client/server solutions, it can reduce the traffic in the network, it can provide more scalability, it allows the use of disconnected computing and it provides more flexibility in the development and maintenance of the applications. In the latest years, several commercial implementations of mobile agent systems have been presented in the market

DEFINITION OF MOBILE AGENTS

a. ***The IBM Agent*** "Intelligent agents are software entities that carry out some set of operations on behalf of a user or another program with some degree of independence or autonomy, and in so doing, employ some knowledge or representation of the user's goals or desires

b. ***The Maes Agent*** [Maes 1995, page 108] "Autonomous agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed."

c. ***General Definition of an Agent:*** "an abstract or physical autonomous entity which performs a given task using information gleaned from its environment to act in a suitable manner so as to complete the task successfully. The agent should be able to adapt itself based on changes occurring in its environment, so that a change in circumstances will still yield the intended result."

d. An agent can also be described as an active object with the ability to perceive, to reason and to act. Moreover, we assume that agents possess their own knowledge and can communicate with each other.

e. This concept of agents can be extended to the concept of intelligent agents, which have the following additional properties. Autonomy: An agent does not depend on the actions of an external entity (human or other). It can act in an autonomous way and has the control of its behavior, its state and its actions. Mobility: An agent can move and decide to join or not a system or an exchange platform. Pro-activity: An agent is able to make decisions and to take initiatives. It works towards personal goals, has its own behavior depending on its individual objectives. It can achieve complex actions, which can be divided into sub-tasks organized according to its behavior. Sociability: Agents can interact and communicate with others. Reactivity: An agent has it's own view of the world and it can react to the changes in the environment. Temporal continuity: Agents are continuously running processes.

AGENT CLASSIFICATIONS

JADE:

Java Agent Development Framework is developed by *Telecom Italia Lab* since July 1998, was released as *open source* in February 2000 (last version: JADE 3.4.1, November 2006). It is a very popular *FIPA-compliant* agent platform. An agent is composed of different concurrent (and non-preemptive) *behaviors*, which can be added dynamically. Among the benefits, we could indicate that there is a wide variety of tools provided (e.g., for remote management and monitoring of agents, and to track interchanged messages) and it can be integrated with different software such as *Jess*1. The JADE built-in *Agent Mobility Service* supports mobility among containers within the same *JADE platform* (similar to the idea of *region* in Grasshopper and SPRINGS), and researchers at the *Autonomous University of Barcelona* provide an *Inter-Platform Mobility Service* Proxies do not exist; instead, an agent searches the current location of its target by querying the *AMS* (*Agent Management System*), according to the FIPA specifications.

AGENT TCL / D'AGENTS:

Agent TCL also called as D'Agents is a mobile agent system created at Dartmouth College to address the weaknesses of existing mobile agent systems, such as insufficient security mechanisms, support for only specific and complex languages, difficult or nonexistent communication between agents, and inadequate migration facilities. The architecture of Agent TCL is based on the server model of Telescript and supports a modified version of the Tool Command Language (TCL) as its high-level scripting language implementation (support for Java and Scheme is being added in D'Agents).

There are two types of agents: those that move from machine to machine-accessing resources, and those that remain on the machine. Security in Agent TCL is provided in various capacities. To protect migrating agents and to provide authentication (e.g., to verify the identity of an agent's owner). Agent TCL uses Pretty Good Privacy (PGP) for its digital signatures and encryption. To protect resources, a resource manager assigns each agent a set of access permissions. So, when an agent tries to access a resource, the request is sent to the resource manager that checks the agent's access permissions with the resource. If the agent does not have the proper permission, it is denied access to the resource. To prevent agents from performing malicious acts, each interpreter is extended to include a security module that prevents such acts. Agent TCL has been used in both information-management and information retrieval applications.

TRYLLIAN:

Tryllian is developed by the homonym company in 2001 (last version: 3.2.0, released as *open source* in November 2005), is based on a *sensing-reasoning action* mechanism. It allows programmers to define a reactive (based on incoming messages) and proactive behavior of agents. Tryllian proposes a *task based* programming model and communication among agents is achieved through *message passing* and in accordance with the *FIPA* standard. It also provides a *persistency* service. The main disadvantage of Tryllian is that it does not offer location transparency. Finally, it does not offer facilities for synchronous communication or conventional method invocation.

ARA:

Ara is a platform for the portable and secures execution of mobile agents in heterogeneous networks. Mobile agents in this sense are programs with the ability to change their host machine during execution while preserving their internal state. This enables them to handle interactions locally which otherwise had to be performed remotely. A mobile agent in Ara is a programmable to move at its own choice and without interfering with its execution, utilizing various established programming languages and the platform provides facilities for access to system resources and agent communication under the characteristic security and portability requirements for mobile agents in heterogeneous networks. The application focus of Ara is on weak-connection/high-volume systems such as wirelessly or intermittently connected computers, or globally distributed large databases. Such environments with intrinsic restrictions regarding the ratio of bandwidth/connectivity vs. data volume seem particularly well suited for mobile agent applications. An important area of application for mobile agents in Ara is in mobile computing. The Ara software currently runs on various types of UNIX operating systems

IKV++ 'S GRASSHOPPER:

Grasshopper is a platform developed by GMD Focus and IKV++. Grasshopper is a mobile agent development and runtime platform, which is built on top of a distributed processing environment. It is completely implemented in Java, based on the Java 2 specification. It has been designed in conformance with the Object Management Group's Mobile Agent System Interoperability Facility (MASIF), and the platform can, furthermore, be enhanced with an add-on, which is compliant with the specification of the Foundation for Intelligent Physical Agents (FIPA). Grasshopper supports several transport protocols by the use of an internal ORB: a proprietary protocol based on TCT/IP, Java RMI, CORBA IIOP, and MAF IIOP. The platform provides comprehensive support for security, communication and persistency

CONCORDIA:

Concordia is a full-featured framework developed at Mitsubishi Electric Information Technology Center America's (MEITCA) Horizon Systems Laboratory. It provides for the development and management of network-efficient mobile agent applications for accessing information anytime, anywhere, and on both wire-based and wireless device supporting Java. The applications move around network machines running Concordia to access services such as databases and those provided by other agents. At the highest level, a Concordia system consists of a Java Virtual Machine (JVM), a Concordia Server running on a machine in a network, and a mobile agent running in the system. The component responsible for agent mobility is the Conduit Server. When an agent wants to initiate its transfer to another machine, it invokes the methods provided by the Conduit Server. Concordia's security model provides support for two types of protection: protection of agents from being tampered with, and protection of server resources from unauthorized access. To protect agents during transfer, Concordia uses encryption. To protect resources on each server, Concordia relies on its Security Manager component to manage resource protection.

MOLE:

Mole is the first Mobile Agent System that has been developed in the Java language. The first version has been finished in 1995, and since then Mole has been constantly improved. Mole provides a stable environment for the development and usage of mobile agents in the area of distributed applications. In Mole system, agent model based on Agents and places. Each Agent's identifier is created at the creating of each agent, which uniquely identifies that agent globally. There is Strong Migration and Weak Migration. In Strong Migration, the underlying system captures the underlying agent's entire state (execution state and data) and transfers it together with the code to a new location where the state of the agent is restored. Week Migration, which only transfer the data, state of the agent. The size of the transferred state information can be limited even more by letting the programmer select the variables making up the agent state. Mole uses Week Migration scheme because one of the goals of Mole to run on any machine having a VM. And a normal java VM doesn't support capture of threads, which would have been required for the Strong Migration scheme.

TACOMA:

An *agent* in TACOMA is a piece of code that can be installed and executed on a remote computer. Such an *agent* may explicitly migrate to other hosts in the network during execution. The TACOMA project focuses on operating system support for agents and how agents can be used to solve problems traditionally addressed by other distributed computing paradigms, e.g. the client/server model.

A TACOMA agent executing on one host moves to another host by using TCP to communicate with TACOMA software at the destination host. Distributed applications have already been implemented using TACOMA to gather and visualize Arctic weather data to provide matching between service providers and potential clients, to communicate and interact with users and to manage software installation in a network.

VOYAGER:

Voyager is purely java agent-enhanced Object Request Broker (ORB) created by ObjectSpace Company. Goals of this product to provide programmer to create state of the art distributed programs quickly and easily while providing a lot of flexibility and extensibility for the products that are being created with the voyager system. Voyager supports RMI, DCOM, and CORBA architecture to provide stationary client server applications, which makes this system very flexible. This is a 100% pure java based system. One of the great advantages of this system is that it supports both traditional client server architecture and agent-based architecture.

Voyager system includes a flexible security framework, lightweight security implementation, support for secure network communications via SSL adapters, and firewall tunneling using HTTP or the industry-standard SOCKS protocol.

AGLETS SDK:

The Aglets Software Developer Kit (ASDK) was developed at the IBM Research Laboratory in Japan. The first version was released in 1996. ASDK requires the JDK 1.1 or higher to be installed. The migration of Aglets is based on a proprietary Agent Transfer Protocol (ATP). The ASDK run-time consists of the Aglets server and a visual agent manager, called Tahiti. There is an additional module of software, called Fiji, that allows the installation of an Aglets server on a HTTP-browser. The ASDK provides a modular structure and an easy-to-use API for the programming of Aglets. This platform has extensive support for security and agent communication and provides an excellent package of documentation.

TRACY:

Tracy is developed at the *University of Jena* in Germany (last version: 1.0.1-52, April 2005), has a *plugin-oriented* architecture. Plugins are software components that can be added dynamically to a running *agency* (the context where agents execute), if required, in order to provide high-level services (e.g., inter-agent communication, migration, security, etc.). Therefore, Tracy agencies are lightweight (a service is not loaded if it is not required) and extensible (new plugins can be developed and added to support new services). The platform offers several *migration strategies* for agents. A key disadvantage of Tracy is that it does not support remote communications between agents: an agent must travel to the agency where another agent is running in order to communicate with it.

SPRINGS:

SPRINGS is developed recently by the *Distributed Information Systems Group* at the *University of Zaragoza* in Spain, focuses on scalability and reliability in scenarios with a moderate and high number of mobile agents. The features of other popular platforms, such as Voyager and Grasshopper, have inspired its development. Similarly to other platforms (e.g., Grasshopper and JADE), it proposes a hierarchical architecture based on *regions*. It provides full *location transparency*: a) for movements (the programmer does not need to specify network addresses but just the name of the destination and b) for calls, through the use of *dynamic proxies*. Moreover, it attempts to minimize the *live lock* problem that can arise when agents move quickly. The main disadvantage of SPRINGS is perhaps that it does not support agent communication using the standard FIPA. Also, it does not provide sophisticated security mechanisms. Despite it is easy to use, it does not offer any graphical tool to the user. Finally, as it is a new platform, there is little documentation available about it.

PIAX:

PIAX (P2P Interactive Agent eXtensions) is an open source framework that integrates Mobile Agents Paradigm and P2P structured overlay network. Using P2P and mobile agent advantages, PIAX allows building a scalable and efficient federated system in a large-scale distributed environment where various kinds of data and processes are located in each device.

SMARD:

Secure Mobile Agent Rapid Development is a development environment oriented programming of on mobile agents based applications. Programmers can design, build and launch mobile agents using a graphical interface. The developed applications can be run on JADE platforms

FMA:

Float's Mobile Agent is free mobile agent platforms which provide phone editing tool allowing users to easily manage all of the personal data stored in their phones, via a number of different connections methods. FMA also allows management of Phonebook (SIM and memory), SMS, Profiles, and Files stored on the phone

SENSORWARE:

Sensorware is an implementation of mobile agent environment for wireless sensor network. Sensorware scripting language is based on Tcl , and Scripts can move their code and data from node to node, autonomously. The distributed algorithms are realized as control scripts that are autonomously replicated or migrated in the “proper” sensor.

MAF:

Mobile Agent Framework is based Python research prototypes which provide a set of primitives to facilitate the development of distributed mobile agent. MAF should provide a mechanism to be able to incorporate and integrate effortlessly with a variety of "foreign agents" written in other languages such as C and C++.

JAMES:

The James platform has been developed by the University of Coimbra (Portugal) in cooperation with Siemens S.A. This platform is mainly oriented for telecommunications and network management. James is not a commercial product. It requires JDK 1.1.1 or higher to execute. The main strengths of this platform include the following features:

Efficient code migration, support for fault tolerance, integration with SNMP, mechanisms for resource control, flexible code upgrading, disconnected computing and agent management. The James platform has been enhanced with a set of mechanisms to optimize the migration of mobile agents, including caching techniques, protocol optimizations, recycling of threads and sockets.

ODYSSEY:

Odyssey is a Java-based mobile agent system from General Magic. It requires JDK1.1 or higher to execute. The platform has a transport-independent API that work with Java RMI, IIOP and DCOM. Odyssey provides the basic functionality and a small set of features.

JUMPING BEANS:

Ad Astra, Silicon Valley Company, commercially distributes the Jumping Beans platform. It requires JDK 1.1.2 or higher to execute. The main strengths of this platform include the support for security, agent management, easy integration with existing environments and a small footprint. However, this platform uses a client/server approach for the agent migration: if an agent wants to migrate between two Agencies it has to go first to the Agent Manager. This approach may represent a point of bottleneck in large-scale applications.

CONCLUSION

In this research paper, we survey & studies several Mobile Agent Systems, all the systems in this paper focused mainly on the environment provided by the system for agents, the mechanisms for agent mobility, agent communication, and language support. The use of Mobile Agents appears to offer certain advantages for client-server computing but as we’ve noted in the above systems, it also raises some difficult issues with respect to efficiency, flexibility and security. These issues have an effect on an agent’s ability of mobility. Many important issues such as how agents determine the available resources/services on a machine it transferred to (i.e., resource discovery), mobile agent system-to-OS interaction, the use of persistent storage, and support for failure were either briefly discussed. Mobile Agents need more applications that take advantage of the characteristics of mobile agents since there is no single alternative to all of the functionality supported by a Mobile Agent Framework. A potential application for mobile agents would involve the use of the Internet and the many uses of the Internet. Solutions to the security and virus problems in Mobile Agents could also result in new and successful methods of Client-Server interaction in network services.

REFERENCES

[1]. David M. Chess, Colin G. Harrison, and Aaron Kershenbaum. “Mobile Agents: Are they a good idea?”, IBM Research Report.

[2]. Dag Johansen, Robert van Renesse and Fred B. Schneider, “Operating System Support for Mobile Agents”, Department of Computer Science, Cornell University, USA.

[3]. David Kotz and Robert S. Gray. “Mobile Agents and the Future of the Internet”, ACM Operating Systems Review

[4]. David Wong, Noemi Paciorek, Tom Walsh, Joe DiCelie, Mike Young, and Bill Peet. “Concordia: An Infrastructure for Collaborating Mobile Agents”, Mitsubishi Electric ITA, Horizon Systems Laboratory.

[5]. “Overview of voyager”, Graham Glass, CTO ObjectSpace

[6]. “Voyager Core Package”

[7]. “Mole- concepts of Mole Agents Systems”, J. Baumann, F.Hohl, Dr. K. Rothermel, M. Straßer

[8]. Grasshopper Programmer’s Guide, *URL: http://www.grasshopper.de.*

[9]. JADE-Board. Java Agent DEvelopment Framework. http://jade.tilab.com/ [2006].

[10]. IEEE Foundation for Intelligent Physical Agents (FIPA). Agent Communication Language Specifications. http://www.fipa.org/repository/aclspecs.html [2006].

[11]. Object Management Group. Mobile Agent System Interoperability Facilities Specification. http://www.omg.org/docs/orbos/97–10-05.pdf [2006].

[12]. S. Ilarri, R. Trillo, and E. Mena, “SPRINGS: A scalable platform for highly mobile agents in distributed computing environments,” in *Fourth International WWMM 2006 Workshop on Mobile Distributed Computing (MDC’06), Niagara Falls/Buffalo, New York, USA.* IEEE Computer Society, June 2006.

[13]. The University of Kaiserslautern. MACE—Mobile Agent Code Environment. http://wwwagss.informatik.uni-kl.de/Projekte/Ara/mace.html [2006].

[14]. Sudmann NP, Johansen D. Adding mobility to non-mobile Web robots. *Proceedings of the IEEE ICDCS’00 Workshop on Knowledge Discovery and Data Mining in the World-Wide Web*. IEEE: Piscataway, NJ, 2000; 73–79.

[15]. IEEE Foundation for Intelligent Physical Agents (FIPA). Publicly Available Agent Platform Implementations. http://www.fipa.org/resources/livesystems.html#zeus [2006].

[16]. K. Burbeck, D. Garpe, and S. Nadjm-Tehrani, “Scale-up and performance studies of three agent platforms,” in

International Conference on Performance, Computing, and Communications (IPCCC'04), Phoenix, Arizona, USA. IEEE Computer Society, April 2004, pp. 857– 863.

[17]. R. Leszczyna, "Evaluation of agent platforms," European Commission, Joint Research Centre, Institute for the Protection and Security of the Citizen, Ispra, Italy, Tech. Rep., June 2004.

[18]. Baumann J, Hohl F, Rothermel K, Strasser M, Theilmann W. MOLE: A mobile agent system. *Software—Practice and Experience* 2002; **32**(6):575–603.

[19]. "What TACOMA Taught Us." Dag Johansen, Fred B. Schneider, and Robbert van Renesse.

[20]. "Mobile Agent Computing". A White Paper. Mitsubishi Electric ITA, Horizon Systems Laboratory, January 1998.

[21]. Robert S. Gray and David Kotz and George Cybenko and Daniela Rus. "D'Agents: Security in a multiple-language, mobile-agent system", Mobile Agents and Security, Lecture Notes in Computer Science, Springer-Verlag, 1998.

[22]. "The Architecture of the Ara Platform for Mobile Agents", Holger Peine and Torsten Stolpmann. In proceedings of the first International Workshop on Mobile Agents, Berlin, Germany, 1997.

[23]. "An Introduction to Mobile Agent Programming And The Ara System", Holger Peine. Technical report ZRI-Report 1/97, Department of Computer Science, University of Kaiserslautern, Germany, 1997.

[24]. "TACOMA- Fundamental Abstractions supporting Agent Computing in a Distributed Environment" Nils Peter Sudmann, Department of Computer Science, University of Tromsø, Norway, November 1996.

Short Bio Data for the Author

RAJGURU PRAKASH VITHOBA received the M.C.M.from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad and M.Phil. Degree in Computer Science from Algappa University Karikudi in 2005 and 2008 respectively. From 2005 he is working as Lecturer of Computer Science in Adarsh Education Society's, Hingoli. He is doing Ph.D. He attended many National and International Conferences, Workshops and Seminars. He is having five International & three National Publication in Standard International Journals. His areas of interest are Mobile Agent, Computing Educational Evaluation, Human Computer Interaction, Green Computing, and ICT.

Dr. DESHMUKH SUSHANT BAPURAO received the M.Sc Physics with Specialization in Advanced Electronics from Pune University Pune in 1992 and Ph.D (Microwaves) from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad in June 1996. He has working as Lecturer from last 11 year and have Nine-year Research experience. He is having Eleven National & International Research Publication. He is also Life Member of Indian Physics Association, Physical Society of India. He has work on Two Research Project Funded by UGC. He is Recognized Research Guide of S.R.T.M University, Nanded & J.J.T University Rajasthan. His Research areas of interest are Dielectric Spectroscopy, Finite Element Method, Quantum Chemistry, Mobile Agent, Computing Educational Evaluation, ICT.