Intelligent versus Malicious Agent: A Comparative Study

Dost Muhammad Khan[1], Nawaz Mohamudally[2]

[1]Assistant Professor, Department of Computer Science & IT, The Islamia University of Bahawalpur, PAKISTAN & PhD Student, School of Innovative Technologies & Engineering, University of Technology Mauritius (UTM), MAURITIUS

[2]Associate Professor, & Consultancy & Technology Transfer Centre, Manager, University of Technology, MAURITIUS (UTM)

Abstract
The agent can be categorized into two types; one is known as an intelligent agent and the second is malicious agent, called computer viruses. These types of agents have some common characteristics; both are intelligent by nature and both have the properties reproduction and transfer from one host to another host in the network, called transferability. The fundamental difference between these agents is that the intelligent agent uses deliberate transport infrastructure while the malicious agent hijacks the resources. The intelligent agents are used in a distributed environment because they are not cumbersome for the network traffic; they overcome network latency, operate in heterogeneous environment and possess fault-tolerant behavior, on the other hand the malicious agents are inefficient and wasteful for computer network resources. This is a discussion paper; we present a comparative study of intelligent and malicious agent and also draw a comparison between an intelligent agent and a distributed object.

Keywords: *Intelligent Agent, Malicious Agent, Distributed Objects*

1. Introduction

This is mobile era and the current global affairs in the mobile communications area with the advent of series of emerging technologies and paradigms have changed the computing model from a one to many to a many to many. This new area also has created a new paradigm called intelligent agent. The concept of intelligent agent is over a decade old, the technology proves not to be perfect yet. Many researchers are now developing methods for improving the technology, with more standardization, better programming environments, as well as design patterns that many allow intelligent agent to be used in products. It is obvious that the more an application gets intelligent, the more it also gets unpredictable, dangerous and uncontrollable and some time becomes a malicious, which is harmful for the resources of the computer. The intelligent agent in most aspects is considered to be new a programming paradigm. There are not many applications out there that use intelligent agents, so it is hard for this technology to become widely adopted. Most work had been around the agent's framework instead of developing the real applications. An agent's envisioned autonomous behavior, involving collaboration with other agents at various network locations, creates a dynamic environment that requires new design methodologies and modeling tools to properly formulate and construct agent-based system. There are many agent frameworks that exist today namely, ABLE, Agent Builder, Aglets, JADE, JATLite, Kaariboga and many more. There must be standardization on some specific execution environments and the format on how mobile agents should be encoded in terms of code and state, which allows agent to work with other agents. An agent is either considered as a distributed object or it is malicious instead of intelligent [16][17][18][19]. This paper envisages on a comparative study of intelligent and malicious agent.

The rest of the paper is organized as: Section 2 is an overview of agents; intelligent and malicious agent, section 3 is about the results and discussion whereas conclusion is drawn in section 4.

2. Overview of Agents

An agent can either be intelligent or malicious. The malicious agent is called a computer virus. In this section we will discuss in detail these two categories of agent i.e. intelligent and malicious agent.

2.1. Intelligent Agent

According to Ted Seller, IBM Almaden Research Center, an agent is a software thing that knows how to do things that you could probably do yourself if you had the time. According to G.W.Lecky & Thompson, an agent is a piece of software 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 software should be able to adapt itself based on changes occurring

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

in its environment, so that a change in circumstances will still yield the intended results. In other words we can say an agent is a piece of program which takes action in pursuit of a goal on the basis of knowledge, information and reasoning. The knowledge representation, reasoning and learning topics of Artificial Intelligence are required for the intelligence of an agent. An agent is like an object which has independent thread of control and can be initiated. Every agent looks like an object, when an agent is implemented in its frame work, it works as an independent thread and can be initiated from the server for execution according to its execution plan. This makes an agent different from an ordinary object. The intelligent agent is commonly referred as mobile intelligent agent or simply an agent. The term 'intelligent agent' is derived from two different disciplines, term 'agent' is created from Artificial Intelligence and 'code mobility' is defined from the distributed systems [1][2][3]. An intelligent agent can be divided into a weak and a strong notations [13]. Table 1shows the properties for both the notations.

Table 1. Notations of an Agent

Weak notation	***Strong notation***
Autonomy	Mobility
Social ability	Benevolence
Reactivity & Proactivity	Rationality
Temporal continuity	Adaptivity
Goal oriented	Collaboration

The properties of strong notation are an agent is mobile, adaptive, collaborative and rational. On the other hand, the properties of weak notation are an agent is autonomous, proactive and goal oriented. The intelligence refers to the ability of the agent to capture and apply domain specific knowledge and processing to solve problems. An intelligent agent uses knowledge, information and reasoning to take reasonable actions in pursuit of a goal. It must be able to recognize events, determine the meaning of those events and then take actions on behalf of a user. The fundamental element of intelligent behaviour is the ability to adopt or learn from experience. An intelligent and autonomous agent has properties like perception, reasoning and action as shown in figure 1 [13].

Figure 1 Properties of an Agent

The agent perceives the state of its environment, integrates the perception in its knowledge base that is used to derive the next action which is then executed. This generic cycle is a useful abstraction as it provides a black-box view on the agent and encapsulates specific aspects [13]. Figure 2 depicts the life cycle of an agent.

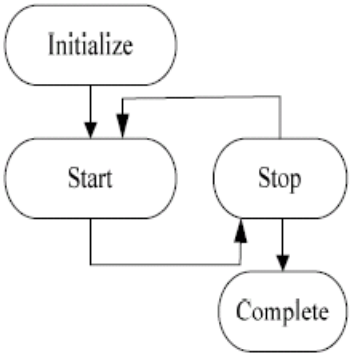

Figure 2 The Life Cycle of an Agent

An agent initializes itself in the first state, whereas in the second state the agent starts to operate; the next state is the stop and agent may start again depending upon environment and tasks that it tried to accomplish. The agent reaches to its complete state after completing all the tasks. This makes the life cycle of an agent. A sample of a program is given below:

```
public class agentClass extends Kaariboga {
  public agentClass(String name)  {
     super("Agent Name_"+name);
  }
  public void run()
  {
         // this is the main method of an agent;
         // here different classes are instantiated;  }
}
```

In this sample the class 'agentClass' extends the class 'Kaariboga', which is a super class and is the basic requirement of each developed agent under this framework. The constructor 'agentClass' is used to give the name the agent, followed by different 'methods', the required method is 'public void run ()', where different classes are instantiated.

An intelligent agent is program that can be dispatched from one computer and delivered to a remote computer for execution. It is capable of autonomous actions in pursuit of a specific goal. The autonomy in agent implies that the software agent has the ability to perform its tasks without supervision, or at least with minimum supervision, in which it will be a semiautonomous software agent. Its autonomy distinguishes it from general software programs. The agent is capable of to perform concurrency, client-server asynchrony, reduce network usage (bandwidth and frequency), installation of client-specific interfaces and dynamic interface upgrades. The client becomes 'smart', an Internet-connected device that allows the user's local applications to interact with server-based applications through the use of web services. It supports work offline. It can be deployed and updated in real time over the

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

network from a centralized server. It supports multiple platforms and languages because of web services. It runs on any device that has Internet connectivity, including desktop, workstations, note books, PDAs and mobile phones [9][10][11][12]. The basic working of a mobile agent is depicted in figure 3.

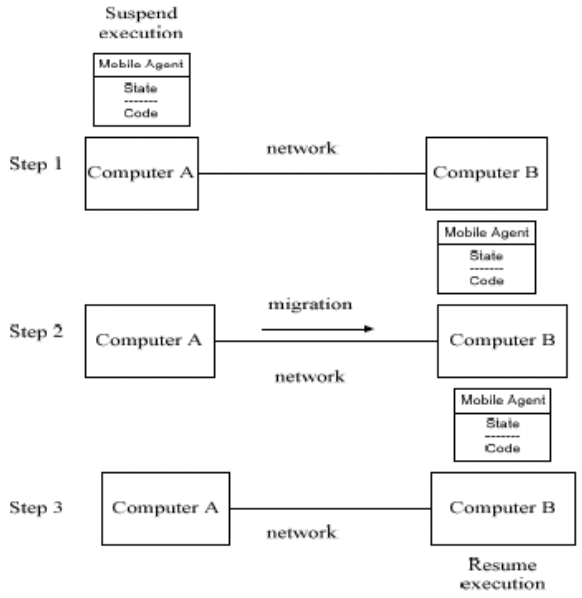

Figure 3 Function of Mobile Intelligent Agent

The explanation of figure 5 is that Computer A and Computer B are connected via a network. In step 1 a mobile agent is going to be dispatched from Computer A towards Computer B. In the mean time Computer A will suspend its execution. Step 2 shows this mobile Agent is now on network with its state and code. In step 3 this mobile Agent will reach to its destination, computer B, which will resume its execution.

The following are areas but not limited to where mobile intelligent agents can be applied:

i. Distributed Computing: Mobile agents can be applied in a network using free resources for their own computations.
ii. Collecting Data: A mobile agent travels around the net. On each computer it processes the data and sends the results back to the central server.
iii. Software Distribution and Maintenance: Mobile agents could be used to distribute software in a network environment or to do maintenance tasks.
iv. Mobile agents and Bluetooth: Bluetooth is a technology for short range radio communication. Originally, the companies Nokia and Ericsson came up with the idea. Bluetooth has a nominal range of 10 m and 100 m with increased power. The applications for mobile agents are myriads.
v. Mobile agents as Pets: Mobile agents are the ideal pets. Imagine something like creatures. What if you could have some pets wandering around the internet, choosing where they want to go, leaving you if you don't care about them or coming to you if you handle them nicely? People would buy such things won't they?
vi. Mobile agents and offline tasks: Mobile agents could be used for offline tasks such as, an agent is sent out into the internet to do some task, and the agent performs its task while the home computer is offline and returns with its results. Mobile agents could be used to simulate a factory, machines in factory are agent driven, agents provide realistic data for a simulation, e.g. uptimes and efficiencies, and simulation results are used to improve real performance or to plan better production lines [4][6][7][8][15].

Many researchers are now developing methods for improving the technology, with more standardization and better programming environments that may allow mobile agents to be used in products. It is obvious that the more an application gets intelligent, the more it also gets unpredictable and uncontrollable. The main drawback of mobile agents is the security risk involved in using them [5][14]. The table 2 shows strengths and weaknesses of agent technology.

Table 2. Strengths and Weaknesses of an Agent

Strengths	***Weakness***
Overcoming Network Latency	Security
Reducing Network traffic	Performance
Asynchronous Execution and Autonomy	Lack of Applications
Operating in Heterogeneous Environments	Limited Exposure
Robust and Fault-tolerant Behavior	Standardization

2.2. Malicious Agent

A malicious agent called a computer virus is a malicious computer program that infects host systems and replicates itself to other host systems. A computer virus has been defined as a set of computer instructions that reproduces itself and it may attach to other executable code, usually the code is a short program that may either embed in other code or stand on its own. In essence, this computer

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

program is designed to infect some aspect of the host computer and then copy itself as much and as often as it has the chance. A computer virus is a kind of malicious agent written intentionally to enter into computer without the user's permission or knowledge, with an ability to replicate itself, thus continuing to spread. Some viruses do little but replicate others can cause severe harm or adversely effect program and performance of the system. A virus should never be assumed harmless and left on a system [20][21]. The most common types of malicious agent are:

i. Resident Viruses: This type of virus permanently residents in the RAM memory, from there it can interrupt and control all of the operations of the system, such as , corrupting files and programs that are opened, closed, copied, renamed etc. The examples of resident viruses are: Randex, CMJ, Meve, and MrKlunky [20][21].

ii. Direct Action Viruses: This type of virus replicates and when it executes take action i.e. when a specific condition is met, the virus will go into action and infect files in the directory or folder that it is in and in directories that are specified in the AUTOEXEC.BAT file PATH. The batch file is always located in the root directory of the hard disk and carries out certain operations when the computer is booted [20][21].

iii. Overwrite Viruses: This type of virus deletes the information in the files that it infects; rendering them partially or totally useless once they have been infected. The only way to clean a file infected by an overwrite virus is to delete the file completely, thus losing the original content. The examples overwrite viruses are: Trj.Reboot, Trivial.88.Dare and many more [20][21].

iv. Boot Virus: This type of virus affects the boot sector of a floppy or hard disk which is an essential part of a disk, where information about the disk itself is stored along with the bootable program for the computer from the disk. In order to protect from the boot viruses, it is the best way to ensure that floppy disks are write-protected and never start the computer with an unknown floppy disk in the disk drive. The examples of boot viruses are: Polyboot.B, AntiEXE and many more [20][21].

v. Macro Virus: This type of virus infects files that are created using certain applications or programs that contain macros. The macros make it possible to automate series of operations so that they are performed as a single action, and hence save the user to carry them out one by one. The examples of macro virus are: Relax Melissa.A, Bablas, and O97M/Y2K and many more [20][21].

vi. Directory Virus: The directory viruses change the paths of a file which indicate the location. Executing a program or a file with the extension .EXE or .COM, infected by a virus, you are unknowingly running the virus program, the original file and program is moved by the virus. It is impossible to locate the original file if it is infected by directory virus [20][21].

vii. Polymorphic Virus: The polymorphic viruses encrypt or encode themselves in a different way by using different algorithms and encryption keys every time they infect a system. This makes it impossible for anti-viruses to find them using string or signature searches because they are different in each encryption and also enables them to create a large number of copies of themselves. The examples of polymorphic virus are: Elkern, Marburg, Satan Bug, and Tuareg and many more [20][21].

viii. Companion Viruses: Companion viruses are file infectors, resident or direct action viruses. They are known as companion viruses because once they get into the system they "accompany" the other files that already exist. In other words, in order to carry out their infection routines, companion viruses can wait in memory until a program is run i.e. resident viruses or act immediately by making copies of themselves i.e. direct action viruses. The examples of companion viruses are: Stator, Asimov.1539, and Terrax.1069 and many more [20][21].

ix. FAT Virus: The file allocation table or FAT is the part of a disk used to connect information and is a vital part of the normal functioning of the computer. The attack of such types of viruses is dangerous, by preventing access to certain sections of the disk where important files are stored. The damages caused can result in information losses from individual files or even entire directories [20][21].

x. File Infectors: This type of virus infects programs or executable files with an .EXE or .COM extension. When one of these programs is run, directly or indirectly, the virus is activated, producing its damaging effects. The majority of existing viruses belongs to this category, and can be classified through their actions [20][21].

xi. Worms: A worm is a program very similar to a virus; it has the ability to self-replicate, and can lead to negative effects on the system. If worms are detected in a system then it is possible to remove them through anti-virus programs. The examples of worms are: PSWBugbear.B, Lovgate.F, Trile.C, Sobig.D, Mapson and many more [20][21].

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

xii. Trojans or Trojan Horses: Trojans or Trojan horses are another type of malicious code, which unlike viruses do not reproduce by infecting other files, nor do they self-replicate like worms [20][21].

xiii. Logic Bombs: They are not considered viruses because they do not replicate. They are not even programs in their own right but rather camouflaged segments of other programs. Their objective is to destroy data on the computer after certain conditions. Logic bombs go undetected until launched, and the results can be destructive [20][21].

3. Results and Discussion

We can simulate attack of malicious agent on a small, isolated network by using an intelligent agent. A malicious agent has two common characteristics; first, at least a partially automated capability to reproduce and second, a method of transfer which is dependent on its ability to attach itself to other computer entities like programs, disk sectors, data files, etc. that moves between these systems. The abilities to automatically reproduce and move to other hosts are indigenous properties of all malicious agents, called computer viruses. This ability to transfer and reproduce manifests as the rapid spread of a malicious agent through a computer network. A malicious agent tends to be inefficient and wasteful for the host computer and network resources. The properties of reproduction and transferability are not unique in malicious agents, the intelligent agents are also capable of to be reproduced and transferred. An intelligent agent can be defined as an entity that can receive, percepts through sensors from its environment, and perform actions on its environment through affecters. An autonomous agent, a kind of intelligent agent, senses its environment and acts in pursuit of its own agenda and so as to affect what it senses in the future. A mobile agent, another kind of intelligent agent, is capable of transporting itself from one machine to another. Therefore, the combination of these agents called mobile autonomous agent (simply intelligent agent) and is capable of behaving much like a malicious agent called computer virus because it possesses the capability to autonomously reproduce and transfer itself to other machines. A malicious agent i.e. virus can be disguised as an intelligent agent and distributed in the network causing damage to the host machines that execute the agent. Can we think of intelligent agents as a kind of malicious agents i.e. viruses? The answer is “No”. The major and fundamental difference is, the intelligent agents use deliberate transport infrastructure on the other hands the malicious agents i.e. computer viruses hijack some bug or feature of existing distributed system like email and ftp.

Table 3. Comparison of Intelligent and Malicious Agent

Property	*Intelligent Agent*	*Malicious Agent*
Communication	Asynchronous	Both
Processing Power	More Autonomous	Not Capable
Network support	Heterogeneous	It can work in any type of the network
Network Management	Reduce Network traffic and latency	Not Capable
Transport Protocol	TCP	Unknown
Network Packet size	Code and execution state can be moved around network. (only code in case of weak mobility)	The full code (Only full text of a program)
Network Monitoring	It gives flexibility to analyze the managed nodes locally	Not Capable
Type of the Client	Smart	Not Capable

A Sample piece of Code	1. public class agentClass extends Kaariboga{
	2. public agentClass(String name) {
	3. super("Agent Name_"+name); }
	4. public void onArrival() {
	5. // display the message; }
	6. public void onDestroy() throws ArrayIndexOutOfBoundsException {
	7. // display the message; }
	8. public void run() {
	9. sum = a + b;
	10. System.out.println (“The sum is:” + sum);}}
		No Specified program code

Frameworks	Yes, Intelligent agent always uses and send through a specific framework	No, Malicious agent does not use any framework. It spread through email or ftp.
Name	Every Agent must has a unique name	No name for any malicious agent, every time it changes its name
User’s Authentication	Requires the authentication of the user	Enters without the user’s permission
Usages	Agents are used to collect data from distributed location, software distribution and maintenance etc.	There is no usage of malicious agents

In an intelligent agent, the client becomes smart; which supports work offline, can be deployed and updated in real time over the network from a centralized server, supports multiple platforms and languages because of web services and runs on any device that has Internet connectivity, including desktop, workstations, note books, PDAs and mobile phones. The codes to add two integer numbers are shown as an example in table 3, an agent in ‘Kaariboga Java based Agent framework’. Every agent has its unique name and is always sent through a specific framework. On the other hand, the malicious agent has no specific program code, its transport protocol is unknown, it can work both synchronous and asynchronous modes of communication, it can work in any type of network either heterogonous or homogenous and its purpose is only to damage the computer resources. There is no framework and no name for malicious agent, every time it changes its name. Furthermore, intelligent agent requires the authentication of the user while the malicious agent enters without the permission of the user.

Because the intelligent agent is distributed in nature therefore, it is considered as a distributed object i.e. intelligent agent is a distributed object with a different name like Microsoft’s objects, OLE, COM, DCOM, ActiveX and many more. We believe that if we are talking about a single agent then it looks like a distributed object. We examine a distributed object and an intelligent agent and draw a comparison between them as shown in table 4.

Table 4. A Comparison of an Intelligent Agent and a Distributed Object

Distributed Object	***Intelligent Agent***
It invokes methods	It has conversation

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

Object wait what action for the actions	Agents decide what action will be taken
It is a small piece of program	It is a collection of programs
It is only for achieving a single application goal	They are programs which co-operate with each other to achieve a single application goal.
It does not initiate actions for its own.	It performs autonomous action
They have fixed roles	Agents can change roles dynamically as the application runs.
Objects are used by other to perform actions	Agents can say 'no' when requested to perform an action

In a distributed object, the objects are defined through the data and the behavior i.e. the methods are required to implement an object. The interactions between objects are explicitly defined; the sequence of methods calls is spelled out in detail. An object is a software entity that encapsulates state i.e. the data and behavior i.e. the function and exposes a set of methods or procedures to manipulate that state. Objects are used by objects to perform action. Objects do not initiate actions of their own preference. In an intelligent agent, we have a collection of software entities that autonomously perform actions. Each agent has a more complex internal state than an object, but more importantly, they have internal goals, where an agent can decide what to do and when to do it. An agent can say 'no' when requested to perform an action but this is not the case in distributed object. Distributed object has fixed roles but the agent can change roles dynamically as application runs. In a distributed object, every object is contributing through a small piece of function to achieve a single goal. In an agent, a collection of goal-oriented software entities cooperate to achieve a single goal. Distributed object involves methods while the agent has conversations. Distributed object is data packets with buttons waiting to be pushed while the agent actively decides what buttons to push.

4. Conclusion

In this paper, we discuss intelligent and malicious agent and draw a comparison between these types of agents. The intelligent agent is always sent through a specific framework while the malicious agent spreads through 'ftp' or 'email' and never uses any framework. The user's authentication is required for an intelligent agent to enter into the computer but the malicious agent enters without the permission of user. We also draw a comparison between and intelligent agent and a distributed object. The intelligent agent performs autonomous action, changes its role dynamically as the application runs while the distributed object has the fixed roles and does not initiate actions of its own. The intelligent agent can say 'no' when requested to perform an action while the distributed object does not has this capability. We conclude here security, performance, lack of applications, limited exposure and standardization are the major weaknesses that intelligent agent is not yet popular enough. It is an era of the people where many users can interact with many intelligent devices at the same time, i.e. one can say it is a many-to-many communication where intelligent agent can play important role in the development of these applications due to their support of asynchronous communication, autonomous and heterogeneous behavior, reduce network traffic and latency and smart clients.

Acknowledgement

The authors are thankful to The Islamia University of Bahawalpur, Pakistan for providing financial assistance to carry out this research activity under HEC project 6467/F – II.

References

[1] Hermans, Bjorn., "*Intelligent Software Agents on the Internet: an inventory of Currently offered functionality in the information society & a prediction of (near-) future developments"*, The Netherlands, 9th July, 1996.
[2] Kiniry, J., and Zimmerman, Daniel, "A Hands-On look at Java Mobile Agents", *IEEE, 1997*.
[3] Robert, S.G., "*Mobile Agents: Overcoming Early Hype and a Bad Name"*, Dartmouth College, Thayer School of Engineering 8000 Cummings Hall, Hanover, New Hampshire, 2004
[4]Lang, D.B., "Mobile Agents: Environments, Technologies, and Applications", *Proceedings of the Third International Conference on the practical Application of Intelligent Agents and Multi-Agent Technology*, 1998.
[5]Ichiro, S., "Selection of Mobile Agents", *24th IEEE, ICDCS 2004*.
[6]Sutandiyo, W., Chhetri, M. B., Krishnaswamy, Shonali., and Loke, S.Wai., "Experiences with Software Engineering of Mobile Agent Applications", *Australian Software Engineering Conference (ASWEC) 2004*.
[7]Wayne, Jansen., Peter, Mell., Tom, Karygiannis., and Don, Marks, "Applying Mobile Agents to Intrusion Detection and Response", *National Institute of Standards and Technology Computer Security Division NIST* Interim Report (IR) – 6416 October 1999
[8]Suna, Alexandru., Fallah-Seghrouchni, Amal El., and Klein, Giles, "Using Mobile Agents for resource sharing", *IEEE/WIC/ACM, IAT 2004*.
[9]Suna, Alexandru., and Fallah-Seghrouchni, Amal El., "A Programming language for Autonomous and Mobile Agents", *IEEE/WIC, IAT 2003*.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

[10]Wang, Ji., Shen, Rui., and Zhu, Hong, "Scenario Mechanism in Agent-Oriented Programming", *IEEE, APSEC 2004*.
[11]Suna, Alexandru., and Fallah-Seghrouchni, Amal El., "Programming Mobile Intelligent Agents: an Operational Semantics", *IEEE/WIC/ACM, IAT 2004*.
[12]Tripathi, A.R., Ahmed, T., and Karnik, N.M., "Experiences and future challenges in mobile agent programming", *Microprocessors and Microsystems*, Vol 25. March 2003, pp.121, 129. , 2003.
[13]Mohammadian (ed), Masoud., "*Intelligent Agents for Data Mining and Information Retrieval*", ISBN: 1591401941 Idea Group Publishing., 2004.
[14]Gray, R., Cybenko, G., Kotz, D., and Rus, D., "*Mobile Agents: Motivations and State of the Art"*, Handbook of Agent Technology, AAAI/MIT Press, 2002.
[15] Mangina, Eleni., "Intelligent Agent-Based Monitoring Platform for Applications in Engineering", International Journal of Computer Science & Applications Vol. 2, No. 1, pp. 38 – 48, Technomathematics Research Foundation, 2005.
[16] Bigus, J.P., "*Constructing Intelligent Agents using JAVA – 2nd ed"*. ISBN: 0- 71-39601-X, 2001.
[17]Sundsted, T., "*Agents on the move, Bolster your client apps by adding agent mobility"*, URL:http://www.javaworld.com/javaworld/jw-07-1998/jw-07-howto.html, 1998.
[18]Ichiro, S., "Reusable Mobile Agents for Cluster Computing", *National Institute of Informatics*, URL http://www.cs.hku.hk/cluster2003/presentation/technical/4B-1.pdf, 2003.
[19] URL: www.agentlab.de/aose.html, web site visited in 2011.
[20] URL: http://hubpages.com/hub/Types-Of-Computer-Viruses, web site visited in 2011.
[21] URL: http://www.buzzle.com/articles/different-types-of-computer-viruses.html, web site visited in 2011.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.