



Journal Homepage: - www.journalijar.com
**INTERNATIONAL JOURNAL OF
 ADVANCED RESEARCH (IJAR)**

Article DOI: 10.21474/IJAR01/8090
 DOI URL: <http://dx.doi.org/10.21474/IJAR01/8090>



RESEARCH ARTICLE

COMPARATIVE STUDY OF JADE AND SPADE MULTI AGENT SYSTEM.

Geetha Radhakrishnan¹, Chithambaram.V², and Shunmuganathan K.L.²

1. Asst. Prof., Kerala Agricultural University, Kerala.
2. Principal, Dhanalakshmi College Of Engineering, Chennai.
3. Research Centre, Dhanalakshmi College of Engineering, Chennai.

Manuscript Info

Manuscript History

Received: 12 September 2018
 Final Accepted: 14 October 2018
 Published: November 2018

Keywords:

Spade, Jade, Multi-Agents.

Abstract

The efficient storing, processing and management of voluminous data has become possible with the advent of big data processing techniques and cloud computing. However, as data moves away from a secured environment, into the virtual cloud space, its vulnerability proportionately increases. Safeguarding data when shifting, processing or storing in the cloud has proven to be extremely challenging and there is very little clarity in this regard. This paper aims to improve security of data in a cloud environment through the introduction of efficient agent based security prototype. Portable, mobile, communicating and migrating agents are hosted between user and system, between systems and between platforms. This paper focuses on the performance evaluation of co-operating Multi-Agent Systems on SPADE and JADE platforms.

Copy Right, IJAR, 2018,. All rights reserved.

Introduction:-

Hackers are always one step ahead of any new defence mechanism in the cyber security system, always devising new methods to gain unauthorised access into protected data [1]. Malware are generated with only slight variations, and so network threats and malignant software have increased to the proportions of big data [2], [3]. Under such circumstances, providing security from the clouds is more practical instead of facilitating defence for each threat within the host [4], [5]. It is also more essential and rational to deploy agents with learning abilities into this environment to preclude continuous human intermediation [6]. We propose a multi-agent based system to resolve security issues using agents developed using SPADE and JADE platforms. In this paper, we strive to explain our preference in choosing these two platforms in agent development.

The rest of this paper is divided as follows. In Section 2 we begin our discussion for agent based cloud security by analysing relevant works done in the area of multi-agent systems, FIPA and SPADE. These studies introduce us to the purpose of multi-agents and how they have been used in various sectors. We also take a look at FIPA standards and innumerable efforts to improve these guidelines. The prevailing agent platforms along with their various drawbacks are also explained. Section 3 presents the architecture of SPADE and JADE platforms. It discusses the step by step development of agents using SPADE and JADE, and gives an outline of the execution of these recommended agents. Section 4 examines the observations and provides the system evaluation. Section 5 discusses the inferences, implications and prospective improvement areas to augment this archetype.

Corresponding Author:- Geetha Radhakrishnan.

Address:- Asst. Prof., Kerala Agricultural University, Kerala.

Background

Multi-agent systems have found use in varying fields through their logical, technical, practical, efficient methods to decipher and resolve challenges [7] – [12]. Along with the ability to interact with each other and with their alternative actions [13], they also have self-healing protection systems [14]. This society of intelligent agent systems capable of adjusting to situational variations, interact and exchange knowledge to achieve a common goal without external interventions, is a Multi-Agent System (MAS) [10]. MAS may contain a combination of humans and agents. Agents in a MAS are autonomous and have only a local view of the system. Also, no agent in the MAS is assigned with governing or supervisory authority.

Implementations of agents are based on Foundation for Intelligent Physical Agents (FIPA) standards [15]. FIPA is the organization that establishes standards for computer software, interacting agents and agent-based systems. According to FIPA, an agent is a computational process that implements autonomous functionality of an application. Agents are autonomous, which means that they are independent, self-governing and have control over their own actions. They are reactive and respond to variations in the environment. They learn from previous experiences and make impromptu tactical adjustments and take initiatives to achieve their goals as they are goal-directed, thereby displaying their pro-activity and flexibility [16]. Agents are social and interact with both humans and other agents. They are also mobile and are able to move with ease from one machine to the other. Their robust nature ensures continuity and vigorous fault recovery to realize goal achievements [17].

The FIPA Agent Platform (AP) follows the FIPA reference model. This comprises of Agent Management System (AMS), Directory Facilitator (DF), Agent Communication Channel (ACC) and the Internal Platform Message Transport (IPMT). The DF provides an indexed almanac to the services provided by other agents registers and deregisters the agents. AMS also registers and deregisters agents, provides an alphabetically filed directory of agent descriptions, supervises and controls access to the AP and delivers life-cycle management facilities. The ACC provides inter-agent communication both internally and traversing through platforms. IPMT provides message forwarding services within a platform [18], [19].

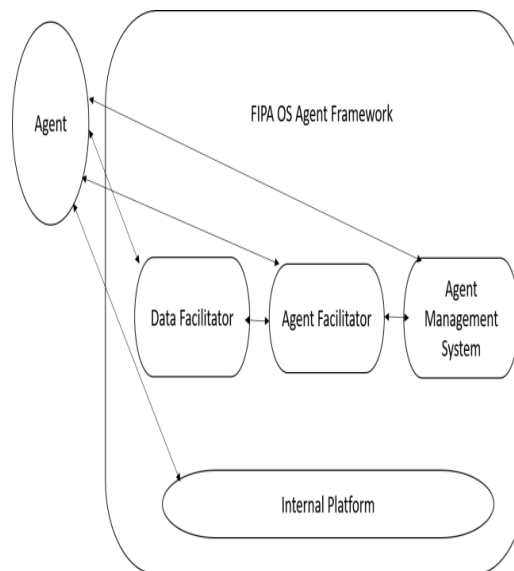


Fig.1:-FIPA OS Agent Platform

Agents conforming to FIPA, benchmarks are developed using FIPA Open Source (FIPA-OS) and is shown in figure 1. FIPA OS features include maintenance of agent shells belonging to various categories, message and conversation management, dynamic alignments to support multiple IPMTs, diagnostics and virtualization tools, abstract interfaces and software designs [20]. The agents communicate with each other using FIPA Agent Communication Language. Conversation, ACL message, syntax and semantics are the four FIPA ACL components. The syntax of the messages is domain independent and need not be defined for all domains addressed.

The semantics/ontologies aid in the definition of names and data types to be used in messages and their meanings. Messages in the colloquy are explicitly interpreted by all involved participants [21]. FIPA standards has played a major role in the development of Multi Agent Systems MAS [22], [23].

Several agent development platforms have evolved and many more are being conceived. Selection of the ideal agent to suit our purposes is an important factor in agent development. Based on the surveys of agent platforms [24] - [27], we understand that most platforms are Java based. Of these very few have operating systems and are still active. It is clear that only 5 of them follow FIPA standards and an even lesser number has got good agent communication language.

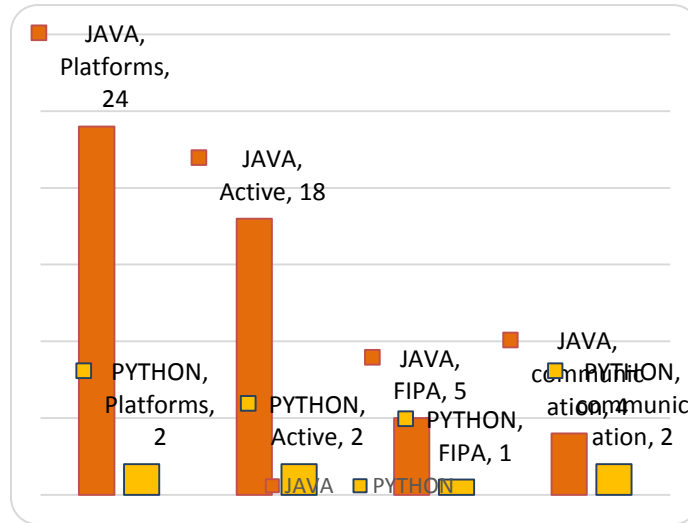


Fig. 2:-Agent Platforms features using Java and Python

Of the agent platforms that were reviewed, Java Agent Development Environment (JADE) and Smart Python Agent Development Environment were found to be ideally suited for agent development.

SPADE

Smart Python multi-Agent Development Environment is a Multiagent and Organizations Platform which is scripted using Python [28] - [30]. It is the pioneer agent platform based on XMPP/Jabber technology and propounds services that expedite the design of multi-agents that can communicate with each other through a communication protocol like FIPA-ACL.

The agents developed using SPADE run on distributed machines and are capable of communicating with each other. Each agent when created is assigned with a Global Unique Identifier (GUID) and a Transport Address (TA) so as to enable registration with the white paper services. APIs are provided with the task to create, suspend, resume, freeze, migrate, clone and kill agents.

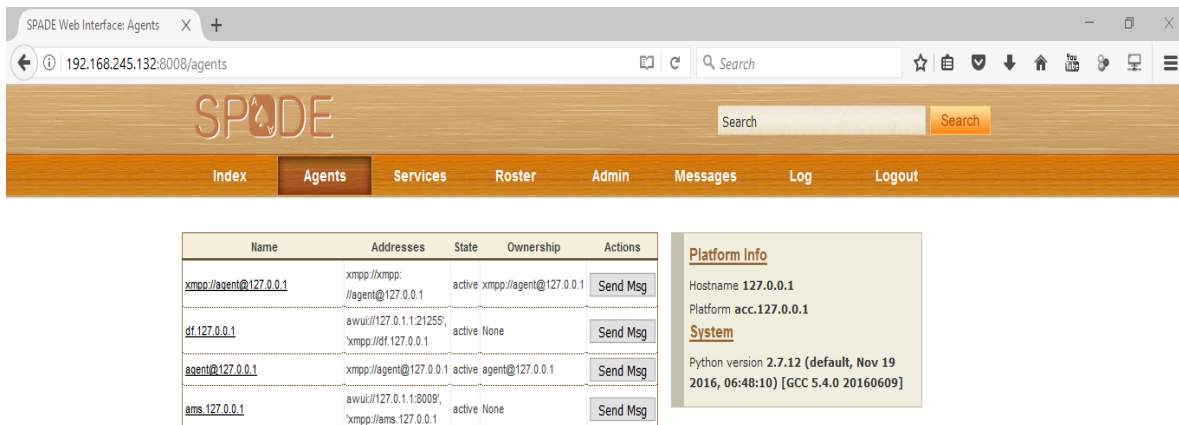


Fig. 3:-SPADE Agent Platforms after login

SPADE application comprises of agents with unique names, that reside over platforms made of one or more containers. Containers can be executed in a distributed manner on various hosts. The Main Container is the first container that is to be launched. It hosts the Agent Management System (AMS) that handles the agents and Directory Facilitator (DF), which facilitates the white pages and default yellow pages of the system, supervises the container table, which is the registry of object references and transport addresses of all the container nodes and manages the Global Agent Descriptor Table (GADT), a registry of all agents in the platform.

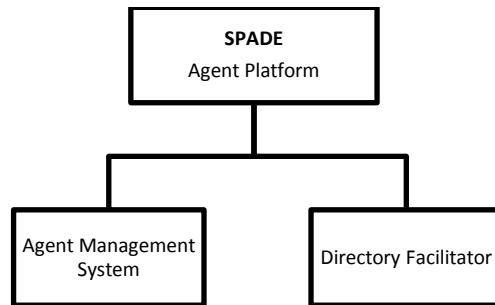


Fig. 4:-SPADE architecture

SPADE agent comprises of a procedure for connecting to the platform, message disseminator and several behaviours to which the messages have been transmitted. Each agent sets up a platform link using a unique Jabber Identifier (JID) and valid platform. The JID or name of the agent, comprises of a username and server domain separated by @ e.g. agentname@127.0.0.1

The address of the agent is the Agent Communication Channel (ACC) and is as follows:
e.g. xmpp://agentname.127.0.0.1

Platform Connection: SPADE agent has in-built jabber technology and so, when an agent is connected to the SPADE platform, it gets automatically registered, and remains active, persistently communicating with the agent platform.

Message Disseminator: This is the component that ensures delivery of a message to the correct recipient or agent.

Agent behavior: This is the task executed by the agent repeatedly. An agent can have several behaviours and each behavior has a message template attached to it. This template determines the kind of message that a behavior should receive. There are some pre-defined behavior types postulated by SPADE and these aid in the implementation of the tasks assigned to an agent. They are:

1. Cyclic and Periodic behavior that are used for execution of repetitive tasks
2. One-Shot and Time-Out behavior help when informal tasks are to be performed
3. Finite State Machine is used when complex behaviors are to be developed and
4. Event behavior responds to the agent

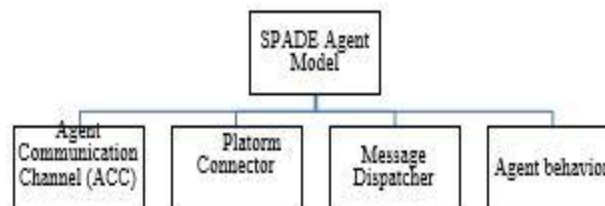


Fig. 5 SPADE Agent Model

The communication is established as defined by FIPA ACL through asynchronous message delivery. The SPADE ACL message includes the sender, the receiver and the actual communication which declares the objective of the sender. Agents also communicate through the Jabber technology.

Jade

Java Agent Development Environment is a robust and efficient agent platform, developed using Java language, and follows the FIPA standards. Due to its JAVA base, it can effortlessly network with Java implementations and has the ability to be distributed across machines running on different Operating Systems. JADE includes an active run-time

environment where agents can live, a library of classes that programmers have to develop, a graphical tools suite for administering and monitoring the running agents [31].

Active Jade *environments* are called containers. Each container can have several agents. A set of active agents is called a platform. Each JADE agent has a unique name for its identification. The main container in JADE has three automatically created agents called

RMA:

the Remote Management Agent which handles the GUI interface

AMS:

the Agent Management Service, which keeps of all agents and JADE programs

DF:

the Directory Facility, which advertises agent services

The Main Container must be started first. A central Java RMI registry, keeps a record of all active containers to ensure effortless agent communication.

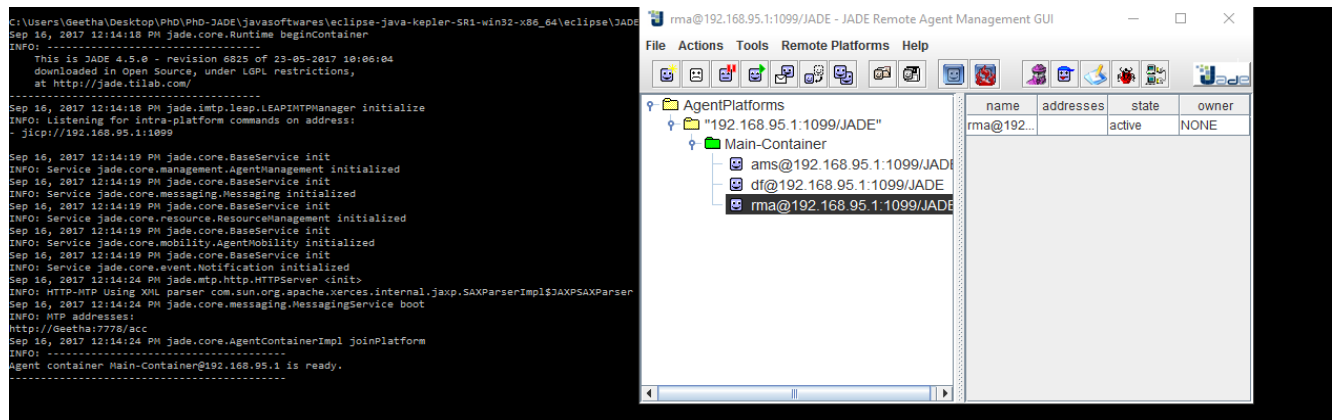


Fig. 6:-Depicting the Main Container and default agents in JADE

Communication between agents takes place using the Agent Communication Language (ACL). Each ACL message has the following fields:

1. Message sender
2. Message recipient
3. Message intention/performative
4. Message content
5. Language of the content
6. Ontology

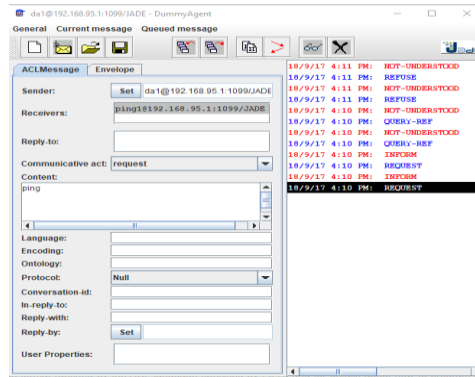


Fig. 7:-Communication between agents in JADE

Proposed System

Agents have been proposed for uninterrupted security service on the cloud. It has been proposed to set up Sentinel agents between localhost and cloud, between platforms and also between virtual machines. Each Sentinel Agent System (SAS) would comprise of an Authentication Agent (AuthA), File Uploading Agent (FUA) and

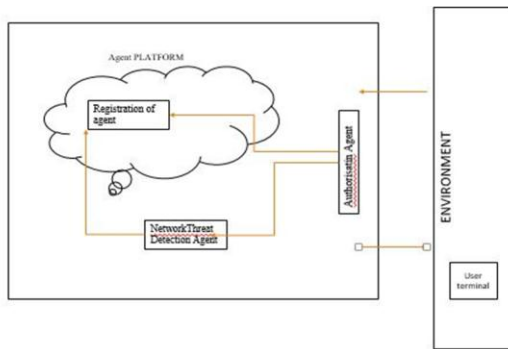


Fig. 8 :-Proposed Sentinel MAS

Agents were developed in SPADE and JADE and registered with the corresponding AMS. Messages were sent between agents. At first both platforms were hosted on the same system and then messages were sent between agents of the same platform. It was found that initially the processing time taken to complete the job was about the same. However, as the number of messages increased, JADE took slightly more time to complete the task. This was because messages were passed from one agent to the next only after a complete thread was formed between all the agents. Waiting time between agents increased slightly as none of the agents were ready to accept messages unless all of the agents were ready to accept messages. However, in SPADE it was found that as each agent completed its task, it became available for accepting the next task and so jobs were done asynchronously but were completed slightly faster. The rate of completion of jobs slightly increased with an increase in the number of messages.

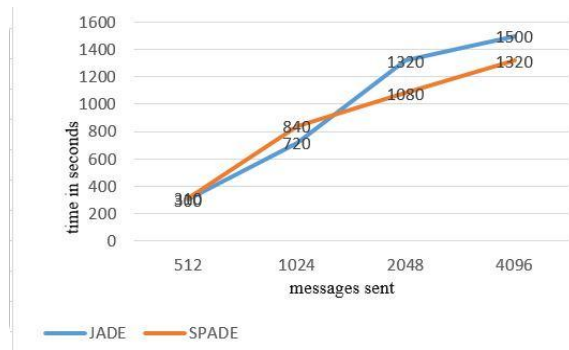


Fig. 9:-Rate of messages being processed

It was also observed that as the messages being transferred increased, CPU performance was affected. The percentage of CPU usage in both JADE and SPADE increased. However, in JADE the percentage of CPU usage was comparatively lesser than in SPADE as the number of messages transmitted increased.

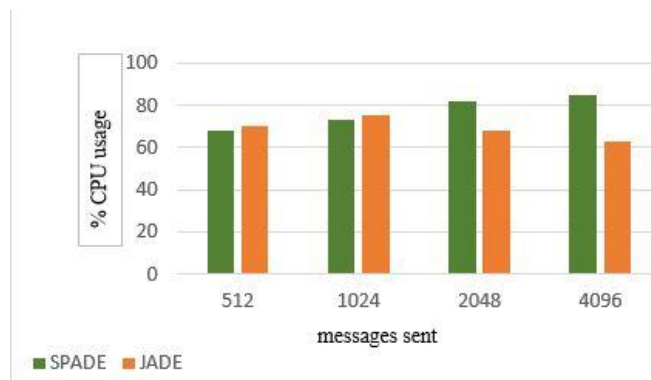


Fig. 10 :-CPU usage in JADE and SPADE

Conclusion:-

This paper has focused on the performance evaluation of co-operating multi agent systems on SPADE and JADE platforms after careful review with other agents. A multi-agent based system was also proposed to resolve security issues for agents developed using SPADE and JADE platforms. Both platforms identify network threats with the aid of agents, and makes suggestions to reduce them. SPADE agent system can utilise CPU optimally for communication and threat detection when compared to JADE. However, malware threat detection has not been implemented. The work will be proceeding further with the development of agents for malware threat detection using ML techniques.

References:-

1. Lori M. Kaufman, Data Security in the World of Cloud Computing, IEEE Computer and Reliability Societies, July/August 2009, pp. 61-64.
2. Huaglory/Tianfield, Security Issues in Cloud Computing, 2012 IEEE International Conference on Systems, Man and Cybernetics, October 14-17, 2012, COEX, Seoul, Korea
3. MirceaGeorgacu, Natalia Suicimezov, Issues Regarding Security Principles In Cloud Computing, USV Anal of Economics and Public administration, Volume 12, Issue 2(16), 2012
4. Diogo A.B. Fernandes, Liliana F. B. Soares, Joao V. Gomes, Marlo M. Freire, Pedro R.M. Inacio, Security Issues in Cloud Environments: A Survey, published online: Springer, Special issue paper, September 2013
5. DimitriosZissis, DimitriosLekkas, Addressing Cloud Computing Security Issues, Future Generation Computer Systems, 28(2012), pp. 583-592
6. Dominic Greenwood, M. Lyell, A. Mallya, H. Suguri, The IEEE FIPA Approach to Integrating Software Agents and Web Services. In Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 07), pages 978-81-904262-7-5 (RPS) 2007 IFAAMAS.

7. Derek McColl, Ming Hou, Kevin Heffner, Simon Banbury, An Agent-based Software Architecture to Aid Human Operators during UAS Target Engagement, 2017 IEEE International Conference on Systems, Man and Cybernetics, Canada, October 2017
8. Divya, K.L. Shunmuganathan. "A Role of Multi-Agents in Intelligent Query Processing: Deploying Co-operative Agent using FIPA –OS in Cloud". *IEEE Conference* 2016
9. Frank Doelitzscher, Christoph Reich, Martin Knahl, Alexander Passfall, Nathan Clarke, An Agent Based Business Aware Incident Detection System for Cloud Environments, *Journal of Cloud Computing: Advances, Systems and Applications*, Springer Open Journal, 2012, pp 1-9
10. Omar Lopez Ortega, Felix Castro Espinoza, Obed Perez-Cortes, An Intelligent Multiagent System to Create and Classify Fractal Music, Springer-Verlag GmbH Austria, January 2018
11. Jing Xie, Chen-Ching Liu, Multi-agent Systems and their Applications, *Journal of International Council on Electrical Engineering*, January 2017, Volume 7, Issue 1, pp 188-197
12. I.E. Foukarakis, A.I. Kostaridis, C.G. Biniaris, D.I. Kaklamani, I.S. Venieris, *Webmages: An Agent Platform Based on Web Services*
13. T. Ishida, H. Yukoi, Y. Kakazu, "Self-organized norms of behaviour under interactions of selfish agents", *Systems, Man and Cybernetics*, 1999. IEEE SMC '99 Conference Proceedings and IEEE Xplore August 2002.
14. Su Seng, K.K. Li, W.L. Chan, ZengXiangjun, DuanXianzhong, "Agent-based Self-healing Protection System", *IEEE transactions on Power Delivery*, Volume 21, Issue 02, April 2006
15. Wiebe Van Der and Michael Woolridge. *Multi-Agent Systems. Handbook of Knowledge Representation. Elsevier B.V. 2007.M.*
16. LiviuPenait, Sean Luke, *Co-operative Multi-Agent Learning: The State of the Art*, Springer Science + Business Media, Netherlands: *Autonomous Agents and Multi-Agent Systems*, 2005, Volume 11, pp 387-434
17. P. Charlton, R. Cattoni, *Evaluating The Deployment Of FIPA Standards When Developing Application Services*, *International Journal of Pattern Recognition and Artificial Intelligence*, Volume 15, Issue 03, May 2001
18. The FIPA Specifications www.fipa.org
19. Stefan Poslad, Phil Buckle, Rob Hadingham. *The FIPA OS Agent Platform: Open Source for Open Standards*, <http://fipa-os.sourceforge.net/docs/papers/FIPAOS.pdf>
20. FIPA (2002a) FIPA Abstract Architecture Specification, SC 00001L. <http://www.fipa.org/specs/fipa00001/SC00001L.pdf>
21. FIPA (2002b) FIPA, SC00067F. <http://www.fipa.org/specs/fipa00067/SC00067F.pdf>
22. P. Charlton, R. Cattoni, A. Potrich, E. Mamdani, *Evaluating The FIPA Standards And Their Role In Achieving Cooperation In Multi-Agent Systems*, IEEE Xplore August 2002
23. Nicholas R. Jennings, Katia Sycara, Michael Wooldridge, *A Roadmap of Agent Research and Development*, Springer: *Autonomous Agents and Multi-Agent Systems*, 1998, Volume1, Issue1, pp 7-38
24. Michael Luck, Peter McBurney, Chris Preist, *Agent Technology: Enabling Next generation Computing ,A Roadmap for Agent based Computing*, Agent Link, Version 1.0
25. KalliopiKravari, Nick Bassiliades, *A Survey of Agent Platforms*, *Journal of Artificial Societies and Agent Simulations*, January 2015, Volume 18, Issue 1
26. Juan M. Alberola, Jose M. Such, Ana Garcia Fornes, Augustin Espinosa, VicentBotti, *A Performance Evaluation of Three Multiagent Platform*, June 2010, Springer Science+Business Media B.V., pp 145-176
27. Florin Leon, MarcinPaprzycki, Maria Ganzha. *A Review of Agent Platforms*. Technical Report, ICT COST, Action ICI404, *Multi-Paradigm Modelling for Cyber Physical systems (MPM4CPS)*, November 2015.
28. The SPADE site <https://github.com/javipalanca/SPADE>
29. Gustavo Aranda, Javier Palance, Natalia Criado. *SPADE User's Manual for SPADE 2.0*.
30. SPADE Tool. <http://code.google.com/p/spade2/>
31. Fabio Bellifemine, Giovanni Caire, AgostinoPoggi, Giovanni Rimassa, *JADE: A Software Framework for Developing Multi-Agent Applications. Lessons Learned*, *Information and Software Technology*, Elsevier, October 2008, pp 10 - 21