



Next Generation in Computing with Agent Oriented Distributed System: Protocols and Features

RAHUL SINGH CHOWHAN^{*1} and PURVA DAYYA²

¹Agriculture University, Jodhpur, India.

²MPUAT, Udaipur, India.

Abstract

Distributed structure of programming and computing has evolved in all facets of development and spawned a new era of distributed computing. In these structures, the isolated processing nodes have been networked to form a powerful processing and communication device. Now, they need an integrated infrastructure for communication that is capable of providing and developing strong and secure mobility features. The mobile agent technology is becoming a promising solution for widely distributed systems like the Internet. Mobile agent based infrastructure is capable of forming migratory environment efficiently using the network resources. The mobile agent is software code capable of moving closer to the required resource in contrast to systems that incorporate the mobility of non-executable code only. It is also required for various agent-oriented languages to support architectural heterogeneity and less or no dependency on the previous system once an agent is transmitted to another system in the network. This paper reviews the architectural components that exist and required for agent-based systems. It also reviews various software components of universal mobile agent system and explores some aspects of migration and intercommunication of agent-oriented approaches.



Article History

Received: 09 May 2018

Accepted: 29 May 2018

Keywords

Agent oriented languages,
Mobile objects,
Distributed programming,
Dynamic itinerary,
On-the-fly execution,
Agent migration.


Introduction

Mobile agents are collaboration of programmable code and variables which can pass on from one system to other on its own within a network environment. The agent performs processing at each host and decides when and where to move next and how it moves. The movements of mobile agent are

evolved from remote procedure call (RPC) methods. In RPC communication methodology a programme calls for a remote method situated within another computer without actually going through the depths of network. Any mobile agent can travel across the network by saving its execution or data state on the current host and then migrate the save state to next

CONTACT Rahul Singh Chowhan  word2rahul@gmail.com  Agriculture University, Jodhpur, India.

© 2018 The Author(s). Published by Techno Research Publishers

This is an  Open Access article licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License (<https://creativecommons.org/licenses/by-nc-sa/4.0/>), which permits unrestricted NonCommercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

To link to this article: <http://dx.doi.org/10.13005/ojcs11.02.08>

host in the network. Another way to achieve this is to move through data duplication by copying and cloning itself¹.

A mobile agent can be termed as a particular formation of mobile code in domain of code mobility. They are useful to implement applications that are capable to work in distributed environment since with their active participation they can move on from one machine to other in their period of execution. There are two types of mobile agent and they are classified as per their dynamic itinerary planning or pre-stored IP-addresses. The mobile agent selects the path depending on its current network conditions². There are two fundamental kinds of mobile agents itinerary planning as mentioned under:

- Predefined itinerary based mobile agent- fixed travel plan.
- Dynamic itinerary based mobile agent- migration path on the fly.

Mobile agent comes under the umbrella of software agent technology though they need not require offering intelligent functionalities like interactive to underlying environment, decision making, competing or collaborative methods. These are the supportive and integrated capabilities of software agent technology. Mobile agents do not preoccupy or make intensive use of computational ability or processing power of resources such as RAM, file processors and network connections. This environment is a buildup of distributed mobile agent systems that can intelligently make decision or simply distribute the task across multiple nodes³. This review paper contributed in various aspects of agent oriented programming capable of serving numerous benefits like mobility, remote execution, autonomous control, goal oriented working etc over existing traditional frameworks. In this paper, various basic needs for execution of mobile agents, required programming languages for development of agents and many other fundamental to advanced features and solutions that mobile agent technology promises to resolve as it evolve to become the most prominent featured standard.

Platform for Execution of Mobile Agent

The mobile agent execution platform comprises of two fundamental things: mobile agents and agent

runtime environment/ agency. For the purpose of migration over connected nodes agents need to initiate a request/call mechanism then execution environment is responsible to transfer that particular agent on an execution environment of next hop⁴. The execution environment is basically the software that executes as a middleware above the underlying operating system. This system in reality takes helps from interpreters and virtual machines to execute agent codes. Sometimes in this concept of marshaling is also used, for which the idea came into existence from stub and skeleton that are used in Remote Procedure Calling mechanism. The marshaling allows a server procedure to be called on client that is altogether a different machine. In mobile agent technology the concept of marshaling is called agent marshaling as the incoming and outgoing communication is based on agents and not just on message passing mechanism⁵.

Agent Marshaling

The process of marshaling involves acquisition of data and transforming them in suitable format that is understandable by current host. Data items like class objects, multiple values associated with a agent code cannot be propagated across the network. For the purpose of transmission it is necessary to convert them into a pre-specified format of data representation like in binary form or stream of bits. The agent migrates to accomplish a task/goal or simply for monitoring of available resources⁶. During its migration it may suspends/deactivates its execution in current environment, save its current data state or execution state and restarts or resumes its execution from the previous point of suspension. The migration helps agents to move close to the host with information resources. This way they introduces a layer of abstraction because for the user it may seems like agent is accessing the information from local resources but in reality it has migrated to another host in the network and communicating back to the home server⁷. Based on this fact the migration of agent can happen in two ways:

Strong migration

This is the capability of mobile agent to transfer itself along with its business logic, execution state and data state. In this the execution state includes PC (Program Counter) and Register Values from stack frame of a program. The data state contains

local variables that are used in business logic, which also migrates with the agent. In strong migration procedure global stack variables also moves which enables new mobile agent execution environment to understand and allow incoming mobile agent to execute.

Weak Migrations

This is the capability of mobile agent to carry the business logic and partially executed state. The states carried during weak migrations are the variable in the stack frame.

Weak migration is an integral part of strong migration because in weak migration only state of current execution moves while in strong migration both execution state and data state get marshaled and transmitted over a network. This often leads to increased processing and lots of computation intensive things. The strong migration moves the business logic to be present at source and destination. It helps the agent in resuming of a mobile agent migrated process on other network hosts⁸.

Features Within an Agent Environment

Agents are allowed on network nodes by use of simple mobile agent server on each and every node. This agent server captures the various states of agent during its migration, registers a new agent in its registry and also provides the security at some extent. The handling of mobile agent depending upon the client features like thin or thick client it maintains the various life-cycle methods of the various incoming and outgoing agents⁹.

Mobile Agent Languages

Mobile agents showcase a novice paradigm for network computing and parallel processing. Mobile agents are the programming entities and that's why developers need programming languages for defining mobile agent. There is wide range of programming languages in which mobile agents can be developed and each of it requires their own agent server compatible to understand that language. Some agent programming languages are confined to research systems under supervised network conditions while others are used in commercial systems. General programming languages like Java can also be used to develop mobile agents by extending special libraries in development environment. Java possess

most of required features that helps and needed in development of mobile agents like multithreading, intermediate byte codes, extended support of communication libraries, and many more. This has allowed developers to come up with Java based environment for development of mobile agents like IBM Aglets, Odyssey, Voyager etc. There exists a few programming languages specially developed for agents like Telescript and Obliq¹⁰. Programming languages allowing development of mobile agents must possess features as under:

- They should enable programs to be marshaled into data and vice versa.
- They should download code from remote computers and link it to run time.
- They should support mobility as an integral feature.
- They must ensure the security of entities generated within the environment
- They should also support the platform independency to allows agents to cover inter platform network.
- They should provide inheritance and polymorphism mechanism to allow agent to either compete or collaborate for a specified goal or assigned task
- They must provide parallel execution for running of various agents in multi agent system.
- They should allow the agent migration on the network nodes for execution of tasks or monitoring purpose.
- They need to represent object communication which enables mobile agents to interact in a multi-agent environment.
- It must allow the similar semantic treatment of remote object as it is for local object on home server.

Agent Execution Management

Mobile agents require an execution environment to run and execute. All hosts willing to accept the agent needs to configure a mobile agent server, also called as agency or home server, for successful execution of mobile agents on them. This process is analogous to execution of processes on operating system. In the same way agency acts as operating environment for the type of mobile agents it is dealing with. In general, mobile agents are developed in various language so

different execution environment is required based on the language used for development of agents¹¹. Mobile agents will be allowed to execute on only those hosts in network that are capable of running the appropriate agent server for a particular type of agents. There is a need of such agent servers that can involve and allow the agents from different platforms developed in different language. The multi support of programming language within an agent execution environment is more needed in the multi agent system where multiple agents can interact, compete or collaborate for the goal assigned to them. So for successful execution of any agent, the agent server must support the semantics and syntaxes of that language in which agents are developed¹².

- Execution Environment is responsible for management and execution of agents.
- It makes it possible for agents to be available on the machine.
- It provides agent with standalone platform for independent execution.
- Agent is capable of accessing the functionalities served by the current execution system.
- It avails the execution runtime for mobile agent to maintain various execution states.
- This must incorporate a procedure to send and receive agents to and from the agent server.
- It also ensures the inter-agent, inter-platform and agent-to-platform communication for delegation of tasks.
- It must provide a security mechanism to prevent malicious agent from execution on the current host.
- Avails with API based communication mechanism for various transfer requests.

Inter agent communication

The agents are capable of communicating and interacting with each other being on same or other machine. An agent can also interact with agent in other hosts by asking/scheduling for a meeting on its execution environment or agency. It is the responsibility of agent server to intimate all or individual agent about the call of meeting method by some other agent residing on other host. The agents can reply by sending rejection or acceptance

of meeting method call based on their competence and requirement¹³. It is an inbuilt feature of any agent programming language allowing development of mobile agent to allow them communicated with each other whether remotely or locally. The general programming languages like Java support the communication facility by providing the proxy objects. The proxy objects have the definitions for life cycle methods like cloning, deactivating, suspension, and retraction etc. for the Aglets¹⁴. These methods basically represent the communication methods to allow the any general programming language to add up the features of agent programming languages. They provide agent communication mechanism like:

- Method invocation.
- Stream based communication.
- Publish or subscribe based event passing.

Security

Security is the crucial part of any network system as the intercommunication happens frequently among the network nodes which may make security breaches and leaks. For most of mobile agent system security is an integral component of agency or agent server but few agents are dependent on security features provided by programming language. The security manager available with the agency allows or disallows the outside access and nature of access of environment. This can be built within the language that a object will be permitted with special permission over the network or not. The various agent server serves with different security mechanism like Agent TCL provides administrator based security policies to interpret security checks, Telescript uses authority property which is need to be proved when agent migrates between regions, Java based agents use digital signature for authentication and many more¹⁵. There exists three major issues while dealing with security of mobile agents:

- Protecting host from malicious mobile agents
- Protecting mobile agents from malicious hosts.
- Protecting mobile agents from other mobile agents

There are few possible solutions:

- Allowing the agents based on security control mechanism while discarding or restricting the unauthorized access to resource without the server permission.
- Provide authentication mechanism by use of digital signature. They allow execution system to acknowledge only those agents that possess verified certificate or signature. Agents may also contain the history of unauthenticated or authenticated machines they have migrated previously.
- Encryption of an agent and its sensitive data on the agent server may also help to reduce security risks.
- Agent server may allow an isolated execution of incoming mobile agents from other hosts.
- Mobile agents must not be allowed to access the technical details of host and resource information until it is specified with proper credentials.
- Monitoring or control mobile agents may be granted with open privileges to migrate or monitor the network hosts for any malicious activities.
- Some agents may get denied with access or execution if their authentication fails.
- The source of mobile agent may be verified based on the certificate or signature it carries to other hosts.

Mobility and Distribution

Code mobility structures for development of distributed applications can be categorized as client/server (CS), remote evaluation (REV), code-on-demand (COD) and mobile agent (MA) methods:

- The customer server approach is broadly utilized as a part of conventional and modern dispersed frameworks. For instance the information code and execution stay settled at machine A while machine B asks for service from the server with same information contentions of the demand, the code and remaining information to give the administration are inhabitant inside machine B, as reaction machine B gives service asked by getting to computational assets on it, later on agent returns the execution state to

machine A.

- Agents based approach expect that the code and information facilitated by machine A gets migrated and relocates the information and code it needs on machine B, where code is executed.
- In remote evaluation it is assumed that actual business logic lies with the client whereas the computational resource is available with the server. The business logic is sent to server for execution for which in return server gives computed results.
- Code-on-Demand as the name states where data and code are called on client from only when needed. The complete execution and processing of business logic takes place on the processing elements of client. Examples are Java applets¹⁶.

Advantages of Mobile Agent

There are several advantages of mobile agent to develop and deploy multiple services not only in distributed environment but also in smart environments:

Minimized Cost of Communication

Agent oriented programming allows remote communication to appear as if it is happening locally. This creates a new level of abstraction and transparency for the user as the capability of mobile agents to move near to resourceful station and execute locally is completely hidden from the home server on which the agent has spawned its execution¹⁷.

Asynchronous Performance

An agent once migrated to the destination doesn't need to interact with its source computer. Though advanced agent structures provides the backtrack algorithm that can keep intimating the source about the current status of agent in a confined itinerary. Though it becomes difficult in case of huge inter-networks like Internet. This facility of an agent allows remote processing even after the source is shut down.

Dynamic deployment of software

Mobile agents can decide destinations', codes and data that can be dynamically deployed there only while they are needed. It's done in computers whose

computational resources are limited and are useful in smart environments.

Direct manipulation

They are locally executed on the computers it visits. By installing real time system based mobile agents network congestions can be delayed. Upcoming mobile agent systems are capable of dividing their loads to other agents reducing the network load while carrying various saved states of objects. Also various system implements the recovery mechanism which sends the data after a time interval to the home server making the light weight migration for mobile agents¹⁸.

Mobile Agent Application Areas

Mobile agent technology is not just a canonical name for future distributed technologies but it is highly capable of supporting the traditional applications as well as distributed applications. Typical applications for mobile agent system may contain the following:

Remote IR

Most data has been managed in text format on dispersed machines. They require remote searching and indexing to open, read and write files. Mobile agents execute locally on the destination host. Communication among agents and agencies may be reduced as agent has the business logic to filter information that their users need from databases¹⁹.

Network management

With mobility feature mobile agents can navigate to remote hosts and observe local resources. Adoption of agents in existing technologies simply eradicates the need of administration for mentoring multiple network activities like upgrading and many others. Network management has several advantages:

- Transfer of mobile agent on source generates less network traffic than the transfer of raw data because the code comparatively less in size than the processed information.
- It access functions of devices on the node easily it's locally executed on the node.

Computing Mobility

For processing of various tasks mobile agents make use of nodes local capabilities and resources. If an agent is willing to migrate to another node it must

first send request to local agency. The agency grants the condition for mobile agent to successfully migrate over the networking nodes. The system can automatically store the moving agent a queue if the destination cannot be reached²⁰.

Software Testing

Mobile agents can be used in development phase of software operations and mobile computing setting. There are different testing methodologies for software like the code running on machines with mobility is called flying emulator. This kind of code executing within a machine depends on its application logic and services with a local area network.

Active networking

Mobile agent can be termed as an active packet in the terminology of active networks²¹. Basically two methodologies can be used for implementation of active networks:

- The first is active packet methodology which changes the destination address of host within the packets existing in current architecture
- The another is active node approach which allows novice rules being dynamically executed at intermediary and terminal nodes²².

Active Documents

Mobile code technology is broadly utilized as a part of module modules for rich web applications (RIA) in web programs for instance java applet; this gives us intuitive client encounters in light of virtual machines. Portable specialist based modules for RIA can normally convey the two codes and state at customer PCs for instance mobi Doc which fabricates versatile parts reports where archives can be powerfully made out of versatile operator based segments.

Cloud Computing

Load balancing is a process of relocation mechanism that allows appropriate migration of processes loaded on single node. The use of versatile operator advances the use of load balancing that helps in migration of stack frame of program. In a disseminated framework for instance distributed computing framework, PCs have a tendency to be various and their computational burdens are unique. PCs may likewise be powerfully added to

or expelled from the framework. Since versatile specialists can move to different PCs, errands that are actualized as portable operators can be migrated at reasonable PCs whose processors can execute the assignments²³.

Operational Ambient Computing Using Mobile Agents

This mobile agent's application is useful in building and operational ambient computing environment.

Middleware for Ambient Computing

This comprises of machines having constrained assets, for example, confined levels of CPU and measure of memory space. Mobile agents help in preserving this constrained asset since agent should be available at the machine. Ambient computing environment are utilized as social infrastructure which underpins enormous clients and additionally it can bolster a decentralized framework for ambient computing environment²⁴.

Context-Aware Mobile Agent

Some exploration projects also endeavor to relocate agents among connected nodes as per genuine changes in surroundings. Fractional and partial framework supports encasing of mobile agent and gives an extension between development of physical entities and development of mobile agents to help and explain the elements utilizing current context. For instance RFID innovation which ties physical elements with mobile agents and navigate agents to stationary or portable clients close to the spots to which the agents are located.

Personal Assistant Agents

Agents are frequently used to aid true and additionally cyberspaces where their help ought to be given in customized frame as indicated by their client's prerequisites²⁵. Mobile Agents can keep up client references and experiences inside them and relocate

at adjacent machines to cooperate with clients immediately amongst client and agents.

Conclusion

Agent oriented programming is becoming a promising paradigm in computing, mobile systems and distributed systems. are only a usage strategies utilized as a part of the advancement and task of disseminated frameworks including savvy condition as other programming operators, including multi-specialists which are not objectives themselves but rather devices for displaying and dealing with our social orders and frameworks. Consequently future portable specialists will be utilized as fundamental advances in dispersed figuring or surrounding processing. Encompassing processing situations end up social foundation in building and urban areas and with speculation of versatile operators they can go about as helpful apparatus for building and working huge scale surrounding figuring conditions which comprises of huge clients and various gadgets. This new face of object oriented programming is supporting the transition of objects becoming agents and agents becoming smart agents with the development of new communication standards and ubiquitous execution platforms.

Acknowledgemnt

I would like to thank my wife Purva Dayya (Ph.D. Scholar, MPUAT, Udaipur) for her cooperation and insightfulness that greatly assisted this research work to reach its completion. It gives me immense pleasure to extend my sincere thanks to all academic friends and lecturers who have supported me in finishing this refined work. They are (alphabetical order) Ashish Sharma, Shilpa Kumari, Shivam Lohiya and Vitthal Patnecha. Although, I am responsible for any errors in this work and should not disparage the technical rapport and stature of these esteemed persons.

References

1. Shigeki Shiokawa, "Performance analysis for use of mobile agent in wireless multihop networks." In 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), pp.827-832. DOI:10.1109/ICUFN.2016.7537153, IEEE, 2016.
2. S. Sobitha Ahila, and K. L. Shunmuganathan., "Overview of mobile agent security issues—Solutions." In Information Communication and Embedded Systems (ICICES), 2014

- International Conference on, pp. 1-6. IEEE, 2014.
3. H. M. Eldegwi, M. B. Badawy, and Hamdy M. Kelash., "Building a Secure Decentralized Energy System with Remote Monitoring Using Mobile Agents." In 2015 Fifth International Conference on e-Learning (econf), pp. 263-268. DOI: 10.1109/ECONF.2015.80, IEEE, 2015.
 4. Zhixin. Tie, "A Mobile Agent-Based System for Server Resource Monitoring." *Cybernetics and Information Technologies*, Volume 13, Issue 4, Pages 104–117, ISSN (Print) 1314-4081, DOI: 10.2478/cait-2013-0057, December 2013.
 5. Rahul Singh Chowhan and Rajesh Purohit, "Study of mobile agent server architectures for homogeneous and heterogeneous distributed systems." *International Journal of Computer Applications*. 2016; 156, no. 4: 32-37.
 6. Rahul Singh Chowhan, Amit Mishra, and Ajay Mathur, "Aglet and kerrighed as a tool for load balancing and scheduling in distributed environment." In Recent Advances and Innovations in Engineering (ICRAIE), 2016 International Conference on, pp. 1-6. IEEE, 2016.
 7. Tina Setter, Andrea Gasparri, and Magnus Egerstedt. "Trust-based interactions in teams of mobile agents." In 2016 American Control Conference (ACC), pp. 6158-6163. DOI: 10.1109/ACC.2016.7526637, IEEE, 2016.
 8. Youssef M. Essa, Gamal Attiya, and Ayman El-Sayed. "Mobile agent based new framework for improving big data analysis." In Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on, pp. 381-386. DOI: 10.1109/CLOUDCOM-ASIA.2013.75, IEEE, 2013.
 9. Gaoyun Chen, Jun Lu, Jian Huang, and Zexu Wu. "SaaS-the mobile agent based service for cloud computing in internet environment." In 2010 Sixth International Conference on Natural Computation, vol. 6, pp. 2935-2939. IEEE, 2010.
 10. Lingaraj. K, Rajashree V. Biradar, and V. C. Patil. "A survey on mobile-agent middleware design principals and itinerary planning algorithms." In 2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), pp. 749-753. DOI: 10.1109/ICATCCT.2015.7456983, IEEE, 2015.
 11. Jeffrey T. McDonald, Alec Yasinsac, and Willard C. Thompson III., "Mobile agent data integrity using multi-agent architecture" AIR FORCE INST OF TECH WRIGHT-PATTERSONAFB OH, 2004.
 12. Akinyokun Oluwole Charles, Ekuewa J.B., and Arekete S. A., "Development of agent-based system for monitoring software resources in a network environment." *Artificial Intelligence Research* 3, DOI: 10.5430/air.v3n3p62, no. 3 (2014): p62
 13. H. M. Eldegwi, M. B. Badawy, and Hamdy M. Kelash., "Building a Secure Decentralized Energy System with Remote Monitoring Using Mobile Agents." In 2015 Fifth International Conference on e-Learning (econf), pp. 263-268. DOI: 10.1109/ECONF.2015.80, IEEE, 2015.
 14. Mi Young Jo and Keecheon Kim. "A research on the regional routing scheme based mobile agent for SDN." In 2016 International Conference on Information Networking (ICOIN), pp. 211-213. DOI: 10.1109/ICOIN.2016.7427116, IEEE, 2016.
 15. Michail Fragkakis, and Nikolaos Alexandris. "Comparing the trust and security models of mobile agents." Information Assurance and Security, 2007. IAS 2007. Third International Symposium on. IEEE, 2007.
 16. Imran Khan, Mohammad Nauman, Masoom Alam, and Furqan Aziz. "SAuthMash: mobile agent based self authorization in mashups." In Proceedings of the 7th International Conference on Frontiers of Information Technology, p. 41. ACM, 2009.
 17. Marthie Schoeman, and Elsabé Cloete. "Architectural components for the efficient design of mobile agent systems." In Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology, pp. 48-58. South African Institute for Computer Scientists and Information Technologists. ACM, 2003.
 18. Masayuki Higashino, Shin Osaki, Shinya

- Otagaki, Kenichi Takahashi, Takao Kawamura, and Kazunori Sugahara. "Debugging Mobile Agent Systems." In Proceedings of International Conference on Information Integration and Web-based Applications & Services, p. 667. ACM, 2013.
19. Ko-Hsin Cindy Wang, "Massively multi-agent pathfinding made tractable, efficient, and with completeness guarantees." In The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 3, pp. 1343-1344. International Foundation for Autonomous Agents and Multiagent Systems, ACM, 2011.
20. Rui A. Martins, Manuel E. Correia, and Alexandre B. Augusto. "A literature review of security mechanisms employed by mobile agents." In Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on, pp. 1-4. IEEE, 2012.
21. Mohamed A. Madkour, Fathy E. Eassa, Abdullah M. Ali, and Noor U. Qayyum. "Securing Mobile-Agent-Based Systems Against Malicious Hosts'." *World Applied Sciences Journal*. 2014; **29**, no. 2: 287-297.
22. Youssef M. Essa, Gamal Attiya, and Ayman El-Sayed. "Mobile agent based new framework for improving big data analysis." In Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference on, pp. 381-386. DOI: 10.1109/CLOUDCOM-ASIA.2013.75, IEEE, 2013.
23. Gaoyun Chen, Jun Lu, Jian Huang, and Zexu Wu. "SaaS-the mobile agent based service for cloud computing in internet environment." In 2010 Sixth International Conference on *Natural Computation*, vol. 6, pp. 2935-2939. IEEE, 2010.
24. Behrooz Safarinejadian, and Kazem Hasanpour., "Distributed Data Clustering Using Mobile Agents and EM Algorithm." DOI: 10.1109/JSYST.2014.2337114, IEEE Systems Journal 10, no. 1 (2016): 281-289.
25. Chowhan R.S., "Mobile Agent Based Workflow Management System (MAWFMS) for Information Flow and Business Processes", DOI:10.13005/ojcst11.01.08, *Orient. J. Comp. Sci. and Technol.* 2018; **11**(1):40-49.