ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 4, June 2012

Computer Assisted Testing and Evaluation System: Distance Evaluation Using Mobile Agent Technology

[1] Prakash V. Rajguru, [2]Dr. Sushant B. Deshmukh.

[1]Department of Computer Science & IT, Adarsh Arts, Commerce & Science College, Hingoli. (Maharashtra), India.
Email: prakash_rajgure@yahoo.com, Cell. 09822186940
[2]Post Graduate Department of Physics, R.G.Bagdia arts, S.B.Lakhotiya Commerce. & R.Bezonji Science College, Jalna (Maharashtra), India. Email: sushant.d69@gmail.com, Cell. 09422238695

Abstract: **The growth of Internet has led to new avenues for distance education. A crucial factor for the success of distance education is effective mechanisms for distance evaluation (DE). Existing Internet evaluation mechanisms, such as web based testing, rely principally on the client-server model. Such mechanisms usually do not scale well and also do not fully support features like: evaluation of subjective questions, delivery of dynamic content, and off-line examinations. These features are extremely desirable for distance evaluation and there is a need for alternate ways of designing such applications.**
We study the existing mobile agent frameworks to understand state of the art. We then use the mobile agent approach for designing, implementing and deploying a system for distance evaluation of students. We consider the entire examination process: (i) Paper setting, where the examiners spread over the internet collaborate to produce a question paper, (ii) Examination conduction, where the question papers are distributed and the answer papers are collected, and (iii) Answer-paper evaluation, result compilation and publishing.
We have designed and implemented, Computer Assisted Testing and Evaluation System in Distance Evaluation using Mobile Agent Technology. Mobile Agent in Distance Evaluation aims to map closely to real world examination scenarios and addresses the full scope of the examination process, viz., paper setting, distribution and testing, evaluation and result compilation. In this research we describe how to implement the Distance Evaluation using Mobile Agent Technology.

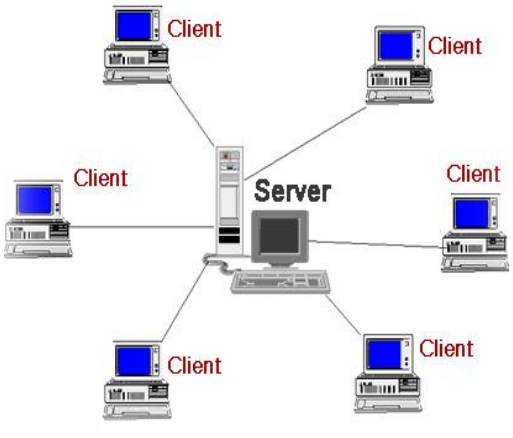
Keywords**: Internet, Client-Server Technology, Mobile Agent Technology, Distance Evaluation, and Agent Transfer Protocol.**

I. INTRODUCTION

As the Internet constantly expands, the amount of available on-line information expands as well. The issues of how to efficiently find gather, and retrieve this information has led to the research and development of systems and tools that attempt to provide a solution to this problem. These systems and tools are based on the use of mobile agents. Mobile agents are processes (i.e., executing programs) that can migrate from one machine of a system to another machine (usually in the same system) in order to satisfy requests made by their clients. They implement a computational metaphor that is analogous to how most people conduct businesses in their daily lives visit a place, use a service, and then move on. Basically, a mobile agent executes on a machine that hopefully provides the resource or service that it needs to perform its job. If the machine does not contain the needed resource/service, or if the mobile agent requires a different resource/service on another machine, the state information of the mobile agent is some how saved, transfer of the mobile agent to the machine containing the necessary resource/service is initiated, and the mobile agent resumes execution at the new machine. Advantages of using mobile agents include low network bandwidth since they only move when they need to move, continued execution even when disconnected from the network, ability to clone itself to perform parallel execution, easy implementation and deployment, and reliability. Mobile agents have been developed as an extension to and replacement of the client-server model. In the client server model, a server is a machine that provides some service (or set of services) and a client (most often another machine) makes requests for those services. Communication between a client and a server is usually through message passing. So, when a client needs a particular service, it usually sends a request message to a server that contains the needed service. A limitation of the client-server model is that the client is limited to the operations provided at the server. So, if a client needs a service that a particular server does not provide, the client must find a server that can satisfy the request by sending out messages to all servers. This clearly is an inefficient use of network bandwidth. Also, this severely limits network scalability since managing and updating these servers would prove prohibitive.

II. CLIENT-SERVER TECHNOLOGY

1. "A Client-Server network is a distributed network



which consists of one higher performance system, the server, and several mostly lower performance systems, the clients. The server is the central registering unit as well as the only provider of content and service. A client only requests content or the execution of services, without sharing any of its own resources."

2. "Client-Server architecture is network architecture in which each computer or process on the network is either a client or a server. Servers are powerful computers or processes dedicated to managing disk drives (file servers), printers (print servers), or network traffic (network servers). Clients are PCs or workstations on which users run applications. Clients rely on servers for resources, such as files, devices, and even processing power."

The most commonly used paradigm in constructing distributed systems is the Client- Server model. In this scheme clients request services or content from a server. The client and server require a known set of conventions before the can communicate. This set of conventions contains a protocol, which must be implemented at both ends of a connection.

ADVANTAGES AND DISADVANTAGES

The following advantages and disadvantages of the Client-Server architecture.

➢ **ADVANTAGES**

- Data management is much easier because the files are in one location. This allows fast backups and efficient error management. There are multiple levels of permissions, which can prevent users from doing damage to files.
- All the data are processed on the server, and only the results are returned to the client. This reduces the amount of network traffic between the server and the client machine, improving network performance.
- Thin client architectures allow a quick replacement of defect clients, because all data and applications are on the server.
- Divides Application Processing across multiple machines:
- Scales Horizontally – Multiple servers, each server having capabilities and processing power, can be added to distribute processing load.
- Scales Vertically - Can be moved to more powerful machines, such as minicomputer or a mainframe to take advantage of the larger system's performance
- Reduces Data Replication - Data stored on the servers instead of each client, reducing the amount of data replication for the application.

➢ **DISADVANTAGES**

- Client-Server-Systems are very expensive and need a lot of maintenance.
- The server constitutes a single point of failure. If failures on the server occur, it is possible that the system suffers heavy delay or completely breaks down, which can potentially block hundreds of clients from working with their data or their applications.
- As the number of simultaneous client requests to a given server increases, the server can become overloaded.

III. NEW TRENDS IN INTERNET APPLICATIONS

There are many trends in Internet technology and activity that encourage the use of mobile agents on the Internet. These trends are outlined and are briefly described below:

i. *Bandwidth***:** Internet access is broadening to the point where people will have a reasonable-speed access to the Internet. The Internet backbone has an enormous amount of bandwidth available.
ii. *Mobile devices***:** Internet users are mobile and therefore they need their Internet access to come with them by using portable computing devices. Everything from laptops or palmtops to car telephones to pagers can access the Internet.
iii. *Mobile users***:** Internet users have shown that they like to have access to everything from anywhere through the popularity of things like web-mail. Web terminals are becoming more and more popular.
iv. *Intranets:* Internal or private and smaller versions of the Internet are being used for information sharing within companies and corporations. Intranets are usually managed by a single organization and can make use of new technologies quickly since security within the intranet is of less concern.
v. *Information overload***:** The massive amount of information available on the Internet today is immeasurable. Users are easily overwhelmed by the sheer quantity of data that is at their disposal. Filtering technology, while still quite limited, can help reduce the stream of information to a given user to a tolerable level.
vi. *Customization***:** Site customization for individual users is possible through the Internet and can be provided on either the client or server side.
vii. *Proxies:* Third party proxies can provide site wide customization for one or more Internet services. They can be used to reduce information overload and customize service access.

IV. DISTANCE EVALUATION SYSTEM

Most of the present day Internet based evaluation is web-based and employs the client-server paradigm. It uses HTML-forms for user interface, with either common gateway interface (CGI)-scripts or java-servlets for back end processing. The students download the questionnaire as a web page and the answers are submitted back to the server. This is essentially the pull-model of distributing the information. The second Internet based model uses java-applets as the front-end for question paper. This too follows a similar mechanism as the previous case

except that using Java gives more flexibility to the examiner in choosing the type of content. With the need for providing multimedia content, multimedia support languages (e.g. flash scripting language) are too being used to provide front-ends.

- **Extending existing distance evaluations schemes:**

We will now highlight the extensions that are desirable in the distance evaluation systems:

i. *Push Model:* In some cases there is a need to send the question paper to the examinee at a time as decided by the examiner. Such a scenario also arises in a case where a number of students are to be evaluated simultaneously for the same set of questions. Most of the paper-based testing methods prevalent today follow this model.

ii. *Variety of delivered contents***:** The use of electronic media for information dissemination has made it possible to present the questions using dynamic content in form of audio, video-clips, or multimedia. It will be desirable to support such rich content in the question-paper.

iii. *Subjective questions*: The students may be required to provide answers that are objective, written text or involve some graphical schematics. All of these cannot be automatically evaluated and would require manual corrections. The present day on-line systems don't have a provision for these.

iv. *Off-line examinations*: The paradigm followed in these schemes is client-server and the students have to remain on-line for the duration of test. For remote interactions, this can be achieved either by opening a socket connection which remains alive during the entire duration of examination, or by opening a socket connection for every request by the client.

v. *Adaptive Questions*: It will be desirable to build adaptive tests wherein questions of various level of difficulty are offered to the candidates in dynamic order. This order is determined by the student's response to the previous set of questions.

V. MOBILE AGENT TECHNOLOGY

➢ *What is an agent?*

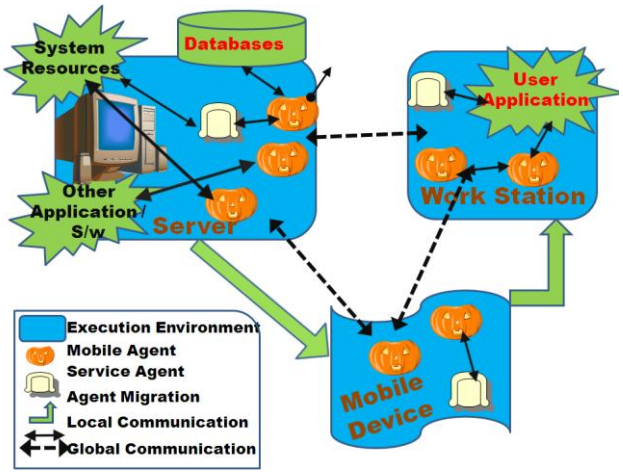
Workers involved in agent research have offered a variety of definitions, each hoping to explicate his or her use of the word "agent."

The IBM Agent [http://activist.gpl.ibm.com:81/WhitePaper/ptc2.htm] "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 users goals or desires."

The Brustoloni Agent [Brustoloni 1991, Franklin 1995, p. 265] "Autonomous agents are systems capable of autonomous, purposeful action in the real world."

Mobile Agent = State + Action + Mobility

The Agent term was already introduced into the computer science in the late 1970s within the artificial intelligence (AI) research (see Nwanda). Agent stands for autonomous intelligent behavior including the ability for communication, which is subsumed in a general introduction of the Agent term in the AI research by Wooldridge and Jennings. Accordingly, an Agent is thought of as a software



entity or component in the modern software engineering owning the following properties, described in a general overview by Bradshaw and Nwanda:

- *Autonomy***,** an Agent can act following self-constructed plans
- Ability for communication, an Agent can interact with other Agents
- *Reactivity*, an Agent can react on environment changes
- *Pro-activity,* regarding to the previous feature, an Agent can react on specific events by performing special standard or optional processes depending on the event
- *Adaptation,* an Agent can be configured to match special user experiences or to solve problems in a specific way
- *Intelligence*, an Agent own on the one side decision-making abilities and on the other side learning algorithms for learning about its user's behavior.

Harrison. Et al. & Braun highlights some desirable characteristics of mobile agents:

- Mobile Agents work in large and heterogeneous networks, where no assumptions about reliability and security of the involved platforms can be made.
- The migration is controlled by the programmer, which means that the program itself and not the operating system decide when and where to go.

- The execution of the program is not location-transparent, but the program moves to a specific server for using services, which are only provided at this platform.
- *Mobility:* Mobility describes the ability of a Mobile Agent to move through a heterogeneous electronic network and deciding autonomously when and where to go. A Mobile Agent migrates for example from a platform A to another platform B to access services of applications or to retrieve data from a database, which are only offered at the target platform of its migration. However, the graphic should emphasize another feature of the Mobile Agent Approach. These Agents can only live in a special software environment; the environment offers two important communication capabilities. External applications are able to interact with Mobile Agents via the local Mobile Agent environment and vice versa.
- *Persistence:* Once a mobile a agent is launched, it should not be reliant on the system that launched it and should not be affected if that node fails. The concepts of an agent moving between network nodes give it the ability to 'survive' and to reach as many resources as possible. This is useful for mobile computer users due to the fact that they can log on, launch an agent, log off and check later on its progress
- *Peer-to-peer communication***:** A failure of the client-server paradigm is the inability of server to communicate with other severs. Mobile agents are considered to peer entities and as such can adopt whichever stance is most appropriate to their current needs. Fore example, when a mobile agent is interrogating a resource it takes the role of a client, when another mobile agent wishes to query it, then it becomes a server. This follows for great flexibility in dealing with network entities and distributed resources.

➢ **Benefits of Mobile Agent Technology**

Mobile agent technology promises to provide some very distinct advantages compared to the other approaches. Some of them are:

1) Overcoming Network Latency Because MAs execute locally, they can respond their environments faster. This is key requirement in some critical real-time systems.
2) Encapsulation of Protocols Upgrading protocols in a distributed system is a cumbersome task. MAs are able to move to remote hosts and establish 'channels' based on the new or proprietary protocols.
3) Disconnected Operations MAs can operate asynchronously and autonomously from the process that created them, after being dispatched. Mobile devices, which need continuous access of fixed network, often suffer from fragile and low bandwidth connects. In such cases they can embed their task in MAs, dispatch them, and then reconnect later to collect these agents.
4) Reduction in network traffic: MA's code is very often smaller than data that it processes, so the transfer of mobile agents to the sources of data creates less traffic than transferring the data.
5) Asynchronous autonomous interaction*:* Mobile agents can be delegated to perform certain tasks even if the delegating entity does not remain active. This makes it an attractive for mobile application and disconnected operations.
6) Interaction with real-time systems: Installing a mobile agent close to a real-time system may prevent delays caused by network congestion.
7) Efficiency savings: CPU consumption is limited, because a mobile agent execute only on one node at a time. Other nodes do not run an agent until needed.
8) Space savings: Resource consumption is limited, because a mobile agent resides only on one node at a time. In contrast, static multiple servers require duplication of functionality at every location. Mobile agents carry the functionality with them, so it does not have to be duplicated.
9) Support for heterogeneous environments: Mobile agents are separated from the hosts by the mobility framework. If the framework is in place, agents can target any system. The costs of running a Java Virtual Machine (JVM) on a device are decreasing
10) Online extensibility of services: Mobile agents can be used to extend capabilities of applications, for example, providing services. This allows for building systems that are extremely flexible
11) Convenient development paradigm: Creating distributed systems based on mobile agents is relatively easy. The difficult part is the mobility framework, but when it is in place, then creating applications is facilitated.
12) Easy software upgrades*:* A mobile agent can be exchanged virtually at will. In contrast, swapping functionality of servers is complicated.

VI. EXAMPLE OF MOBILE AGENT

- *JADE***:** JADE is free software and is distributed by TILab, open source software under the terms of the LGPL (Lesser General Public License Version 2). JADE has good GUI, accessible use, good documentation and high acceptance.
- *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.
- *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.

- *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.
- *VOYAGER:* Voyager is 100% java Agent-enhanced Object Request Broker (ORB) created by Object Space 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.
- *JACK***:** JACK Intelligent AgentTM is an Agent oriented development environment fully integrated with the Java programming language. JACK provides Agent-oriented extensions to the Java programming language.
- *GRASSHOPPER:* Grasshopper was developed by IKV++ in 1999 (last version: 2.2.4, January 2003), and then became part of the commercial Mobile, and today its development has probably been abandoned. It is an easy-to-use platform for mobile agents, compliant with the standards MASIF and FIPA (http://www.fipa.org/). A Grasshopper system can be composed of different regions. It provides agent developers with interesting features, including a graphical user interface to manage agents, agencies, and regions. By defining regions, the developer can benefit from dynamic proxies. The main disadvantage of Grasshopper is that it is not available anymore and new versions will not appear in the future. The region server could become a bottleneck, as it must update every proxy right before using it. A disconcerting feature of Grasshopper (stated in the manual) is that a call to an agent that is moving can end up executing on the copy of the agent at origin (which will be removed once the agent arrives at its destination). Finally, as in Aglets, the same predefined method is always executed after an agent's trip.
- *TRYLLIAN:* Tryllian (http://www.tryllian.org), 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 (based on heartbeats) 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 (the current location of the target agent of a message must be known in advance). In addition, its task based and asynchronous model could be difficult to use, due to its differences with the classical procedural programming. The use of a single thread per agent could be inefficient and a limitation for the programmer. Tryllian provides a large set of configuration options, which could be overwhelming. Finally, it does not offer facilities for synchronous communication or conventional method invocation.

VII. PROPOSED FRAMEWORK

We use the mobile agent approach for designing, implementing and deploying a system for distance evaluation of students. We consider the entire examination process: (i) Paper setting, where the examiners spread over the internet collaborate to produce a question paper, (ii) Examination conduction, where the question papers are distributed and the answer papers are collected and (iii) Answer-paper evaluation, result compilation and publishing.

a) Examination Setting

The examination setting process takes place in a collaborative manner where the examiners sitting at different remote locations prepare their questions. Mobile Agents are then dispatched to these examiners. These MAs fetch the question papers from all of the examiners. The central controlling authority decides on the final question paper based on the inputs from different examiners.

a) Distribution and Testing

Once a question paper is prepared, it is dispatched to the different examination centers with the help of Courier Mobile Agents. Having finished their distribution work, the Courier Agents get either terminated or they return to their place of origin. The distribution servers at these centers have a list of candidates enrolled for that center. The examination paper at each center is cloned to the number of students in each center. The examination papers can time-out themselves after a fixed interval of time. Once a student finishes answering a question or the examination paper times out, the answers are given back to the distribution center, which launches a Answer Mobile Agent for each student answer paper. These Mobile Agents then make their way to the Evaluation Center

b) Evaluation and Result Compilation

Once an Answer Agent reaches the evaluation center, it is supplied with an itinerary of the examiners. The Answer Agents can also move to an Objective Question Evaluator if it possesses answers to multiple-choice questions, to automatically evaluate their answers. The Answer Agents move from one examiner to other, until all of the questions are evaluated. They then move to the Publishing Center where they supply

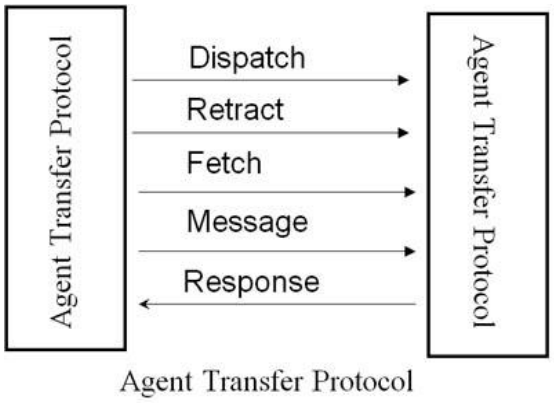
their results and where the final comprehensive results are published.

VIII. MOBILE AGENTS FOR MPLEMENTING THE TECHNIQUE IN DISTANCE EVALUATION

- *Agent Transfer Protocol*

  ATP is a simple application-level protocol designed to transmit an agent in an agent system-independent manner. An ATP request consists of a request line, header fields, and content. The request line specifies the method of the request, while the header fields contain the parameters of the request. ATP defines the following four standard request methods:

- *Dispatch*: The dispatch method requests a destination agent system to reconstruct an agent from the content of a request and to start executing the agent. If the request is successful, the sender must terminate the agent and release any resources consumed by it.



Agent Transfer Protocol

- *Retract*: The retract method requests a destination agent system to send a specified agent back to the sender. The receiver is responsible for reconstructing and resuming the agent. If the agent is successfully transferred, the receiver must terminate the agent and release any resources consumed by it.
- *Fetch:* The fetch method is similar to the GET method in HTTP; it requests a receiver to retrieve and send any identified information (normally class files).
- *Response:* We define response time as the time taken between a student making a request, such as, request for next question or request for next section in the question paper, and getting the appropriate response.

IX. PERFORMANCE EVALUATION

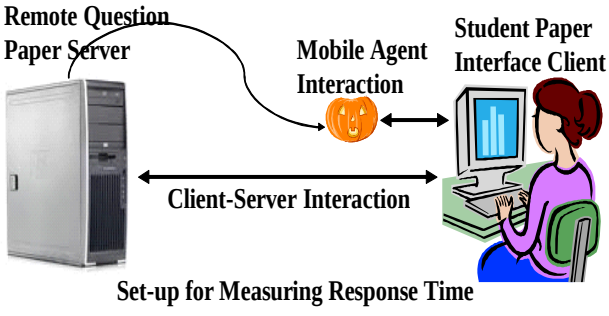
The performance criterion most relevant for our application is the response time for the students. We define response time as the time taken between a student making a request, such as, request for next question or request for next section in the question paper, and getting the appropriate response.

We have performed experiments to make the following two set of measurements:

Response Time for:

Test a - Mobile Agent Interactions

Test b - Client-Server Interactions



Set-up for Measuring Response Time

➢ **Observations**

We see that in case of client-server, the response times will remain more or less constant whereas in the case of MA, the initial response takes much longer while the remaining requests take negligible time as compared to client-server responses. The initial longer response in case of MA is because of the additional time taken for agent creation, dispatch and transfer. Response-time determines the user-experience and hence is critical for our application. In future with the content getting richer (graphics and multimedia support), this difference will become even more pronounced. Traditional client-server distributed programs avoid this problem by techniques like pre-fetching, caching etc. Mobile agents inherently provide these capabilities in our application.

The 'Start', causes a mobile agent to be launched from a remote machine, which brings in the new question paper/ section for the student in the first case. In second case the same first page of question paper/ section is fetched as data from the remote-server. The students, browsing through the given set of questions, generate further queries. In case of MA, these questions would be been pre-fetched by the mobile agent and hence the responses will be local. In the second case every request will cause a remote request to be placed in typical client-server mode.

X. CONCLUSION

We studied the client server technology and Mobile Agent Technology. We also investigate the existing distance evaluation scheme. We survey and analysis the various Mobile Agent paradigm. We check the performance of mobile agent over client server technology and used more powerful mobile agent paradigm for implementing our research project Computer Assisted Testing System in Distance Evaluation i.e. Mobile Agent in Distance Evaluation.

A typical distance evaluation scenario is characterized by large geographical distances, disconnected operations, dynamically interchangeable client-server roles and large number of interacting nodes. By using Mobile Agents as the principal design paradigm in Mobile Agent in Distance Evaluation, we will be overcome many of the limitations of existing systems. Mobile Agent in Distance Evaluation to the full gamut of distance evaluation viz. paper-setting, distribution and testing, and evaluation by means of various types of MAs. We will research how

MAs can be used effectively for structuring such large-scale distributed applications. The gains would be in terms of: scalability, flexibility, dynamic extendibility and independence from the network variations. We also investigate to improving the overall performance of system, reliability of system and System security.

XI. REFERENCE

[1] Mane Dnyaneshwar Rangnath, June, 2010, "Mobile Agent Security – Aglet Platform", Thesis Submitted in partial fulfillment of the requirements for the degree of M.Tech.
[2] Giovanni Caire, 30 June 2009, "Jade Tutorial Jade Programming for Beginners", TILAB, formerly CSELT
[3] "JACK Intelligent Agents" ,Agent Practical 2009
[4] Jingqiao Zhang, September 1, 2009, "JADE: Adaptive Differential Evolution with Optional External Archive"
[5] Mihai Caramihai, and Irina Severin, 2009, "eLearning Tools Evaluation based on Quality Concept Distance Computing. A Case Study", World Academy of Science, Engineering and Technology
[6] A. Kannammal, N.Ch.S.N. Iyengar, December 2007, "A Model for Mobile Agent Security in E-Business Applications", Volume 2, Number 2
[7] Amanpreet Kaur, Sukhdeep Kaur, March 23, 2007, "Role of Mobile Agents in Mobile Computing", Proceedings of National Conference on Challenges & Opportunities in Information Technology (COIT-2007).
[8] Ametller, S. Robles, J. A. Ortega-Ruiz, "Self-Protected Mobile Agents"
[9] Ahlem BenHassinea, Tu Bao Hob, (2007) ,"An agent-based approach to solve dynamic meeting scheduling problems with preferences", Engineering Applications of Artificial Intelligence 20 858, 857–873
[10] Ole Martin Mevassvik, Mr. Karsten Bråthen and Mr. Richard Moe Gustavsen, March 2006, "JADE – An Experiment in Distributed Simulation Based Joint Tactical Training"
[11] Acklin and Interpolis, November 2005, "Software Agents in International Traffic Insurance", A Business Case Study
[12] Dag Johansen, 2004, "Mobile Agents: Right Concept, Wrong Approach", Proceedings of the 2004 IEEE International Conference on Mobile Data Management
[13] Krzysztof Chmiela, Dominik Tomiaka, at all, "Testing the Efficiency of JADE Agent Platform"
[14] KK Wong, CK Heng, PC Leong & MT Yap, 2003, "A Mobile Agent Based Registration System", International Conference WWW/Internet
[15] Kunal Shah, August 2003, ''Performance Analysis Of Mobile Agents In Wireless Internet Applications Using Simulation ''
[16] Lilley, M., & Barker, T. (2003). An evaluation of a computer adaptive test in a UK University Context. In Danson
[17] Burstein, J., Leacock, C., & Swartz, R. (2001). Automated evaluation of essay and short answers.
[18] Chou, Mobile Agent Work across Heterogeneous Systems. Web: http://www.mowahs.com, 2001
[19] Kalle Burbeck, Daniel Garpe, Simin Nadjm-Tehrani, "Scale-up and Performance Studies of Three Agent Platforms"
[20] Linden, W. J. van der, & Glas, C. A. W, 2000, "Computerized Adaptive Testing: Theory and Practice", Norwell, [21] Zoran Putnik, Zoran Budimac, Sept 6-7, 2000, "Mobile Agent - A New and Advanced Concept?" Proceeding of the TARA 2000 Conference Novi Sad, Yugoslavia,
[21] Puliafito, O. Tomarchio, and L. Vita. MAP: Design and Implementation of a Mobile Agents Platform. *Journal of System Architecture*, 46(2):145-162, 2000
[22] Constructing a computer-assisted testing and evaluation system on the World Wide Web- the CATES experience. IEEE Transactions on Education, 3 (43), 266-272. (Ed.), Proceedings of the Seventh Computer Assisted Assessment Conference, Loughborough, UK.
[23] Green, S. et al. Software Agents: A review, *T*
[24] David Kotz and Robert S. Gray. "Mobile Agents and the Future of the Internet", ACM Operating Systems Review
[25] Bradshaw, Jeffrey M. (Editor). Software Agents. Mit Press. ISBN: 0262522349
[26]Agent Systems, Mobile Agents, and Applications by David Kotz

AUTHORS PROFILE

RAJGURU PRAKASH VITHOBA received the M.C.M.from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, M.Phil. Degree in Computer Science from Algappa University Karikudi & B.Ed from SRTMU Nanded in 2005, 2008 & 2011 respectively. From 2005 he is working as Asst. Professor 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 Associate Professor from last 11 year and has 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, and ICT.