

# A Performance Evaluation of a Multi-Agent Mobile Learning System

<http://dx.doi.org/10.3991/ijim.v10i2.4873>

Collins N. Udanor and O.U. Oparaku  
University of Nigeria, Nsukka, Nigeria

**Abstract**—Mobile devices have emerged as our daily companion whose applicability evolves as the day unfolds. One of such applications is in the area of learning, called mobile learning (M-learning). However, as with all new technologies, M-learning is faced with the issues of standard, content packaging, and deployment. And like other distributed applications on ubiquitous networks, M-learning is challenged with performance issues. This work shows the implementation and evaluation of a model for intelligent mobile learning system (IMLS) using a multi-agent system (MAS), such as the Java Agent Development Environment (JADE) framework. Two M-learning applications (agent-based and non-agent) were developed, deployed and tested. The agent-based application is deployed using an HTTP-based Message Transfer Protocol (MTP). The results suggest that agents can increase the performance of an M-learning application up to eight times.

**Index terms**—M-learning, JADE, multi-agents, IMLS, network monitoring, response time

## I. INTRODUCTION

The growing number and sophistication of smart phones and tablet PCs are radically changing the way we access the almost near infinite volume of information on the World Wide Web. Today almost every mobile device has the capability to access the internet, while on the go. Mobile technologies are so pervasive in modern societies that more and more people are using mobile devices for personal, business and public services. These services like M-banking, mobile TV, mobile agricultural information systems, etc, have enhanced lifestyles and improved livelihoods. On the other hand, while universities are still grappling with providing adequate infrastructure to provide quality education to their students, an unprecedented amount of lessons are being accessed on a daily basis from YouTube, and some other online university sites with mobile devices. It is unfortunate that in spite of the myriads of contents uploaded daily to the web, little work has been done in creating structured learning contents for mobile devices. With over six billion mobile subscriptions worldwide, there is great potential in deploying learning contents targeted at mobile devices.

Learning is an active process of building knowledge and skills through practice within a supportive community, which comprises not only a process of continual personal development and enrichment, but also the possibility of rapid and radical conceptual change [1]. There is therefore, a need to re-conceptualize learning in this mobile age in recognition of the essential role that mobility and communication play in the process of learning. Mobile learning (or M-learning) is a type of flexible learning that

makes use of mobile devices to deliver authentic learning environments to students and learners. It is an affordable extension and implementation of e-learning. M-learning is not just about learning with mobile devices, but the mobility of the learner himself. The essential difference between M-learning and other types of learning is that the learner is constantly on the move, therefore promoting the concept of ‘learning anywhere, anytime’. In previous works [2, 3] we proposed and developed a model of an intelligent mobile learning system. This model, unlike Silander’s [4] and Kazi’s [5] models which were non-agent based, took into consideration the effect of response time in delivering contents across wireless and ad-hoc networks by introducing multi-agents. Agents are software components that have been widely used in the fields of artificial intelligence, database management, computer networking, etc. These agents have autonomy, provide an interoperable interface to an arbitrary system and behave like human agents, working for some clients in pursuit of their own agenda [6, 7]. In networks, agents ensure effective bandwidth utilization which in turn optimizes the processing time. Unlike Kinshuk’s Bee-gent framework [8], our system was implemented using the JADE multi-agent framework [7], which is well researched, open source with active user support and is continuously gaining popularity among many researchers in agent technology. JADE is built on Java technologies, which makes it interoperable, platform-independent, robust and suitable to many web-based and mobile applications.

The overall goal of our research was to develop a model of an intelligent mobile learning system that operates well in low-bandwidth networks, such as the wireless and ubiquitous networks, without additional overhead costs or reduction in the quality of service.

In addition to formulating the model, we went further to verify it with the development of an agent-based M-learning software application for teachers and users. This model acts as a template on which teachers can build learning contents to provide authentic learning environments for students to access using their mobile devices. To further strengthen the validity of our work, we also developed a non-agent version of the M-learning software with which to compare the performance of the agent-based one. In this paper we are going to show the results of the comparison of the two applications with regards to their response times and packet handling capabilities.

In the remaining sections, we present a brief review of the design and implementation of the Intelligent M-learning System (IMLS), and the deployment, testing and evaluation of its performance compared to its non-agent based counterpart.



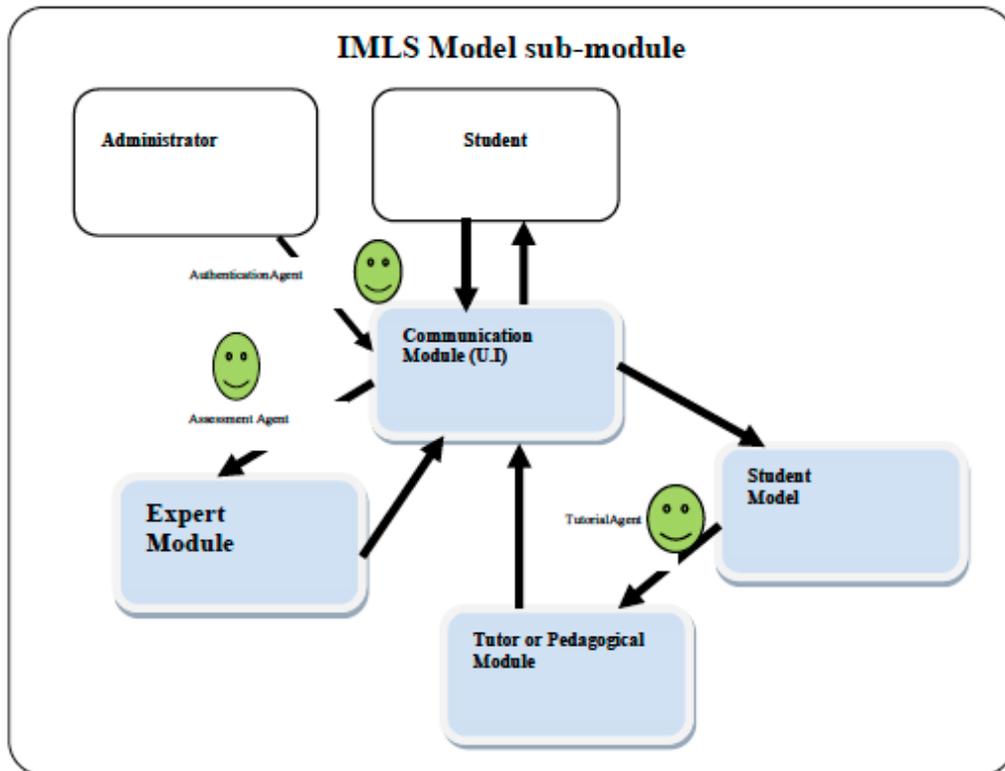


Figure 2. IMLS model sub-module

communicative acts and the content language [29] as defined by FIPA [16]. It is worthy of note, that it is the use of ACL messages that distinguishes agent communication from other distributed systems. ACL is designed specifically to describe and facilitate the communication process between agents. It does not deal with the physical exchange over the network, but with the specification of the content of the exchange [27]. ACL enables agents to cooperate and share contents in agreed format [28]. The FIPA Agent communication act uses a seven layer model from the application layer of the classical OSI reference model. The sub-layers include, in ascending order; Transport, Encoding, Messaging, Ontology, Content Expression, Communication Act and Interaction Protocol (IP) [29]. The layered architecture is geared towards optimizing communication performance between agents in MAS [27]. FIPA defined three Message Transport Protocols (MTPs) as: MTP based on IIOP, MTP based on WAP, and MTP based on HTTP. Our MAS uses the MTP based on HTTP for communication between the agent world and external world. An ACL message is encapsulated in an HTTP POST request. The HTTP request for agent communication consists of the original ACL message, payload, envelope or destination address, HTTP message header and HTTP message body [29]. The payload, according to FIPA may be encoded using any of these three schemes, Bit Efficient ACL, String ACL, or XML encoded ACL [7][27][28][29]. As can be observed from figure 1, there is no direct communication between the agent world and the physical world. User requests are handled by the WSIG servlet via HTTP, while the WSIG agent communicates with the MAS using ACL.

#### B. The IMLS Model sub-module

The model consists of the components that make up the Intelligent Tutoring/Learning System. These include the

student model, expert module, pedagogical module and the communication or user interface module as depicted in figure 2. The model contains the actual course content, the student records, assessment, as well as administrative information. These contents are stored in the MySQL data store using the ORMLite. ORMLite provides a lightweight Object Relational Mapping between Java classes and SQL databases [17]. To interact with these contents, the respective agents are invoked. The components of this model are described briefly below.

**The Student:** A new student registers with the system, choosing the study area of interest. At login the authentication agent verifies the student's id and keeps a log of his activities using session tracking.

**Communications Module:** controls interactions with the learner, including the dialogue and the screen layouts. This module determines how the materials should be presented to the student in the most effective way.

The **Expert Module:** references an expert or domain model containing a description of the knowledge or behaviours that represent expertise in the subject-matter domain the intelligent tutoring system (ITS) is teaching, and provides the basis for interpreting the student's actions.

The **Student Model:** describes the knowledge and behaviour of a sample student of the subject. A mismatch between a student's behavior or knowledge, and the expert's presumed behavior or knowledge, triggers a signal to the final component, the tutor.

The **Tutor or Pedagogical Module:** is responsible for structuring the instructional interventions. It can subsequently act to provide feedback or remedial instruction. To fulfill this function, it needs information about what a human tutor in such situations would do. The pedagogical module may operate at the curriculum level, which se-

quences topics to ensure that an appropriate prerequisite structure is observed and the individualized level, in which the amount of practice at each level ensures that the students master the material.

The IMLS follows a behaviourist pedagogical approach. Within the behaviourist learning paradigm, learning is thought to be best facilitated through the reinforcement of an association between a particular stimulus and a response (drill and feedback) [1]. Mobile devices in particular can enhance the behaviourist learning process. The use of mobile devices to present teaching materials/content specific questions (stimulus), obtain responses from learners (response), and provide appropriate feedback (reinforcement) – (i.e. provide ‘drill and feedback’ activities), fits within the behaviourist learning paradigm. The use of competence tests (both pre- and post) in the IMLS application is an example of the behaviourist approach.

C. Deployed Agents

**Authentication agent:** this agent is deployed as soon as the JADE agent platform is started. It handles all user and administrative authentications and also monitors user activities and sessions. This agent also determines what type of content each category of user has access to, thus ensuring adequate authorization of users. A user gets access to only the study area he/she is registered for. Figure 3 shows the JADE agent platform running on port 1099 and the sniffer agent on the right hand side of the window monitoring the agent activities, while figure 4 shows the authentication agent details in the WSIG Console window.

**Tutorial agent:** this agent searches the pedagogical module stored in the database to retrieve learning contents on behalf of the user and delivers it to his mobile device. Unlike the conventional client-server scenario, in which processing vast volumes of data stored at remote locations in the network can prove inefficient. Mobile agents allow the processing to be done locally, instead of transmitting the data over a network [8]. This is one of the features that make agents effective at bandwidth utilization.

**Assessment agent:** The assessment agent does both pre- and post-tutorial assessments. In the pre-tutorial assessment, the agent tests the knowledge level of the intended learner to find out his level of proficiency in the selected study area. Depending on the result of the assessment, a learner can be placed in the basic, intermediate or advanced class. At the end of the lesson, the agent delivers a post-tutorial assessment that determines the performance of the learner in the course.

D. Implementation of the IMLS Application

In this section we describe the implementation of the server component of the IMLS application. To do a comparison on a level playing ground, a contemporary M-learning system was developed. The major differences between the two applications are that, while the original IMLS1 application is built on the JADE multi-agent framework, IMLS2 is built without any agent technology or framework. Also, IMLS1 is developed using Java Servlet technologies, and IMLS2 is built on PHP scripting language. Both applications have a post tutorial assessment. But, IMLS1 has a pre-tutorial competence assessment that tests the candidate’s level of proficiency in the course field, and ranks the candidate’s competence level as Basic, Intermediate, or Advanced. This determines

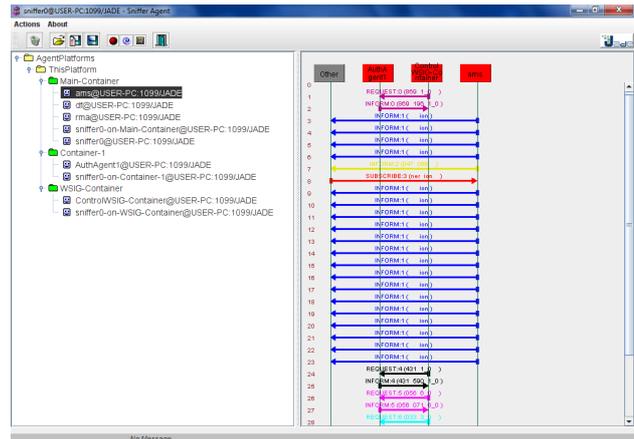


Figure 3. JADE Platform

Figure 4. Authentication agent details

TABLE I. A COMPARISON OF THE TWO IMLS APPLICATIONS

APPLICATION/ PROPERTIES	IMLS 1	IMLS 2
MODEL USED	IMLS MODEL	Not specified
DATABASE USED	MYSQL	MYSQL
AGENTS	JADE multi-agent framework (Authentication agent, Tutorial agent, Assessment agent)	No Agent
DEVELOPMENT ENVIRONMENT	Java (jdk 1.6), Java Servlet Technologies, on Eclipse Java IDE.	PHP scripting language, HTML, Java script.
SITE DEPLOYED/URL	http://196.222.0.3:8080/IMLS_V2/	http://ce1.grid.unn.edu.ng
CONTENT ACCESS	Mobile	Mobile
COURSE ASSESSMENT	Pre and post assessment	Post assessment

what course module the tutorial agent delivers to the candidate at the beginning of his tutorial. Table 1 summarizes the comparison between the two IMLS applications.

The tutorial agent in IMLS1 enables a learner to select a specific *study area*, for example, Computer Science. Then he chooses an *area field* like programming language. The learner then selects from the *field courses* a specific course he wants to learn like Java, PHP, C#, etc. At this point, the assessment agent prompts him to take an optional pre-tutorial assessment, in which he answers ten random questions drawn from the different modules of the *field courses*. A score of 40% or below places the learner in the beginner class, 50-60% places him in the intermediate class, while 70% and above places him in the expert

class. At the end of the pre-assessment, the tutorial agent delivers the *course modules*. From this point the user selects topics from the *course modules* and starts his learning experience. It is important to note that the authentication agent registers a new user and authenticates him before the learning experience begins. This agent also stores the student's registration and session data in the *Student\_Data* table, as well as stores his study records in the *Student\_Study\_Record* table for future retrieval.

The learning experience in IMLS2 is similar to that of IMLS1 described above, only that there are no agents and no pre-tutorial assessment. The inclusion of the pre-tutorial test in IMLS1 is based on adding more intelligence to the model. The intelligence of IMLS2 is done using sessions. Figures 5a-e show screen shots of the IMLS1 as the user interacts with the application.

E. Application Deployment

Both IMLS1 and IMLS2 were deployed on two different HP Proliant branded servers with public IP addresses. Figures 6a and 6b show the deployment diagrams of the IMLS application. A mobile client connects to the applications using the URL of each of the application hosted on the web. The machines on which the applications were deployed have Apache Tomcat web server 7.0 and Mysql database installed. Agent configuration files were created in the Eclipse IDE. These configuration files, like JADE agent platform, Authentication agent, etc are started first. After which the IMLS1 application is started on the Tomcat server, using JSP containers.

IV. APPLICATIONS PERFORMANCES: TESTING AND EVALUATION

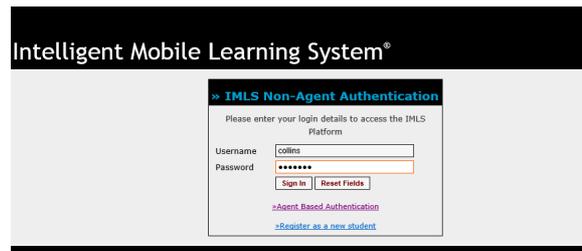
Using network monitoring tools from remote and local locations, the URL of each application is monitored to determine its performance. The response time measured includes both time taken to respond to a user's request and communication across the network. The major performance indicators of interest are the application's response time and packet losses. Response time indicates how well bandwidth is managed and packet loss management on the other hand, indicates the application's throughput. The following section presents the details of the evaluations.

We used network monitoring tools to take various kinds of readings. Two different network monitoring tools were used to monitor the applications. These include Monitor.US ([www.monitor.us](http://www.monitor.us)), a tool that monitors sites using URLs from different locations, such as US, EU, etc. The other is ManageEngine

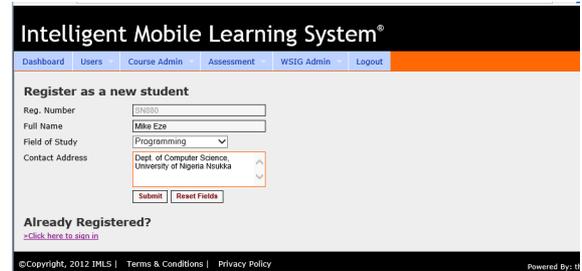
OpManager. This application was downloaded and installed on neutral desktop and laptop PCs. While Monitor.US is used to monitor the IMLS applications with other M-learning applications (like Blackboard, Eacademy's m-learning sites, etc.), in other locations like the US and UK, OpManager was used to monitor only the two IMLS applications. Table 2 shows the average response times of the IMLS applications with other M-learning applications as monitored by Monitor.US. Screen captures of the monitoring are shown in figures 7 and 8. The results are presented in Sections 4.1 and 4.3.

A. Monitoring

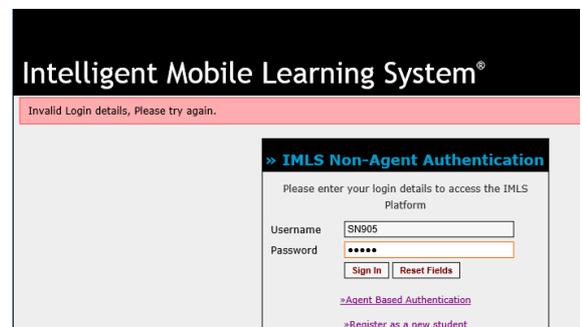
**Using Monitor.US** – This tool monitors URLs, showing the network performance and plots various types of



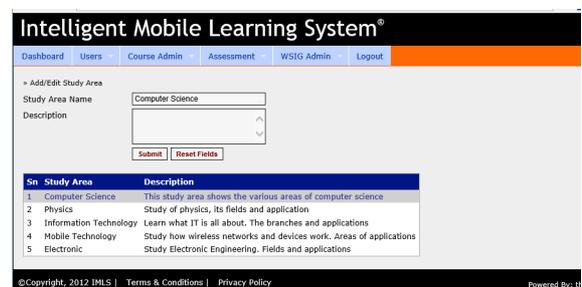
a) User login or registration



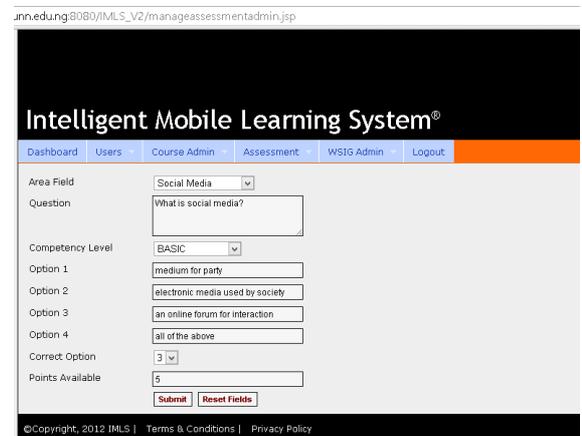
b) user registration



c) new user login



d) study area and field

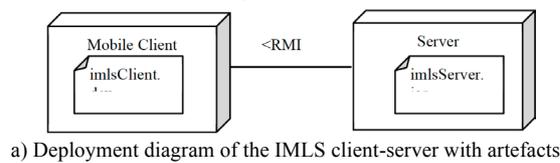


e) assessment test

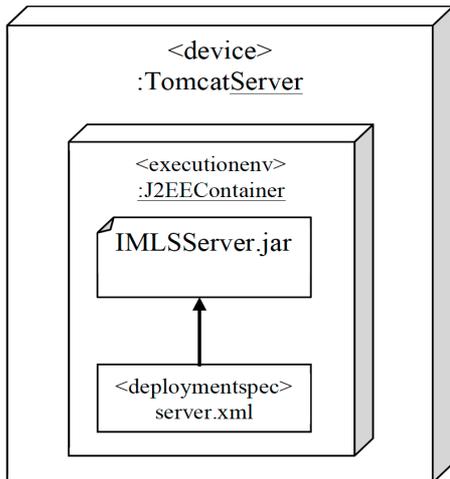
Figure 5.

charts. This tool was used to monitor site availability and response times. Figure 6 shows the live data chart of the IMLS sites with other external M-learning sites being monitored with their respective response times. IMLS sites are shown on the right side of the window, while one of the external sites is shown on the left hand side. The average response time of each site is shown on the top right hand corner of the window for the last 24 hours of monitoring. Table II shows average response time readings on a typical day as captured from Monitor.US.

Figure 8 shows the bar chart plot of the response times for the two IMLS applications being monitored. The IMLS1 is shown on the upper window with the URL, [http://196.222.0.3/localhost:8080/IMLS\\_V2/auth.jsp](http://196.222.0.3/localhost:8080/IMLS_V2/auth.jsp) with an average response time of 539ms for that day. While IMLS2 is shown in the lower window with the URL, <http://ce1.grid.unn.edu.ng> shows an average response time of 634ms for the same day.



a) Deployment diagram of the IMLS client-server with artefacts



b) Device containing an execution environment with an artefact and a deployment specification.

Figure 6.

TABLE II. AVERAGE RESPONSE TIME READINGS ON A TYPICAL DAY AS CAPTURED FROM MONITOR.US.

Sites urls	Site Name	Site Location	Avg. Response time (ms) – All locations
<a href="http://196.222.0.3:8080/IMLS_V2/noagent.jsp">http://196.222.0.3:8080/IMLS_V2/noagent.jsp</a>	IMLS 1-noagent authentication	Nigeria	411
<a href="http://196.222.0.3:8080/IMLS_V2/auth.jsp">http://196.222.0.3:8080/IMLS_V2/auth.jsp</a>	IMLS 1- agent authentication	Nigeria	475
<a href="http://ce1.grid.unn.edu.ng/">http://ce1.grid.unn.edu.ng/</a>	IMLS 2	Nigeria	453
<a href="http://www.gc21-eacademy.org/mobile/">http://www.gc21-eacademy.org/mobile/</a>	E-Academy	US	205
<a href="http://mle.sourceforge.net/mlemoodle/">http://mle.sourceforge.net/mlemoodle/</a>	Moodle M-learning	US	130
<a href="http://www.blackboard.com/">http://www.blackboard.com/</a>	Blackboard	UK	248

B. Determining Response Time

The response time of the system is a measure of Web performance, which depends on the time required to transmit many small files. Many IT managers argue that the impact of latency on network design, not bandwidth, is often the key to network speed, or lack thereof. This is especially true when it comes to the download speeds (or response times) of Web pages and Web-based applications [18]. Alberto Savoia [19] simplified the response time formula, which was proposed by [18] to six key variables that impact page response time and shows how they relate to each other, as shown in equation 1. Alberto describes the six key variables in the formula as follows:

$$R = Page\ size/Min.\ Bandwidth + (Round\ Trip\ Time \times Turns) + Server\ Processing\ Time + Client\ Processing\ Time. \dots\dots\dots (i)$$

In a September 2006 NetForecast report [20], the author gave a newer version of Alberto’s formula as follows:

$$R \approx Payload/Bandwidth + AppTurns(RTT) + Cs + Cc \dots\dots\dots (ii)$$

Where R is the response time in seconds, Cs is the Server processing time (seconds) and Cc is the Client processing time (seconds). The only difference between this formula (ii) and that of Alberto in (i), is the use of the curly equality sign, which means ‘approximately equal to’.

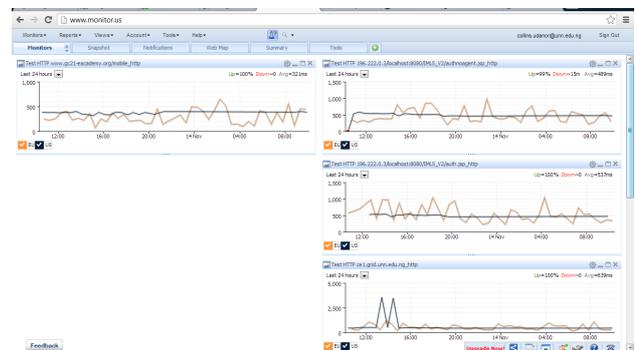


Figure 7. Real time response time data read from sites as monitored from US and EU servers with their average response times.

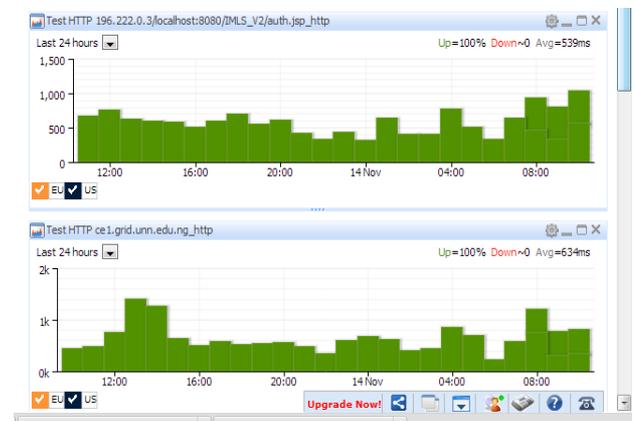


Figure 8. Live chart of the two IMLS applications sites with their average response times.

V. RESULTS AND DISCUSSION

A. Results from Monitor.Us

The results from the readings in Table II above agree with the response time equation 2, which indicates that response time is a function of many factors. From these results, it could be observed that:

1. The farther away the monitor is from the DNS server where the application is hosted, the longer the response time. That explains why the sites from US and EU have shorter response times compared to the sites from Nigeria, as shown in Figures 7 and 8, and Table 2, respectively.
2. The average response time for the two instances of the IMLS1 (in Figure 7) is 443ms as against 453ms of IMLS2. Also, from the bar chart in Figure 8, the minimum response times of IMLS1 is 290ms and that of IMLS2 is 280ms, while the maximum response times are 1,110ms and about 1,400ms, respectively. This sample result indicates that IMLS1, the agent-based application offers a better response time than the non-agent one, IMLS2.

B. Using ManageEngine OpManager Monitor

In other to further verify the results obtained from Monitor.Us, we deployed another monitoring agent, ManageEngine OpManager. As stated earlier, the applications were installed locally on a neutral system. From there we monitored the two IMLS applications. After a week of monitoring, the following results were obtained. Figure 9 shows IMLS1 and IMLS2 being monitored by ManageEngine OpManager.

For want of space we show an abridged response time data as read by the monitoring tool in Table III from the IMLS1 site while figure 10 shows a plot of the IMLS1 data for a week in the month of November, 2012 which shows that the average response time is 31.63 ms.

Table IV shows the response time data for seven days from the Non agent-based IMLS (IMLS2), while figure 11 shows a plot of the IMLS2 data, which shows that the average response time is 253.92 ms.

Figure 12 shows the availability and average packet loss information of the two IMLS sites. IMLS1 is on IP address: 196.222.0.3, while IMLS2 is on: 197.253.4.165. The availability figures are 80.2% and 80.078%, respectively for this screen capture. On the other hand, the average packet losses are 25.0% for IMLS1 and 33.33% for IMLS2, respectively.

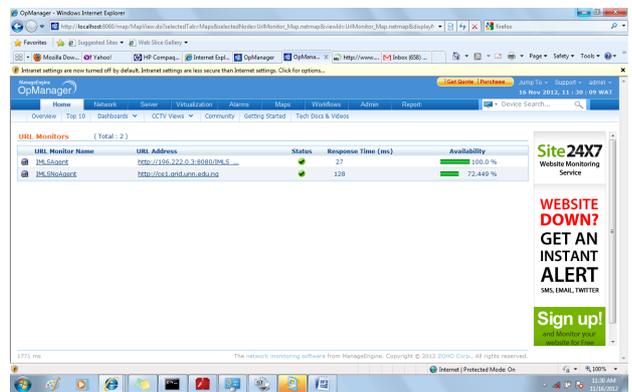


Figure 9. The two IMLS sites being monitored by OpManager Monitor

TABLE III. ABRIDGED RESPONSE TIME DATA FOR SEVEN DAYS FROM IMLS1 DEVICE NAME: IMLSAGENT PERIOD: LAST 7 DAYS

Start Time: 14-Nov-2012 08:55:20 End Time: 21-Nov-2012 08:55:20

Date	Time	Response Time ms
Nov-17-12	2:29	13.0
Nov-17-12	1:29	46.0
Nov-17-12	0:29	15.0
Nov-16-12	22:29	18.0
Nov-16-12	21:29	13.0
Nov-16-12	20:29	17.0
Nov-16-12	19:29	12.0
Nov-16-12	18:29	11.0
Nov-16-12	17:49	16.0
Nov-16-12	13:26	19.0
Nov-16-12	12:29	70.0
Nov-16-12	11:29	24.0
Nov-16-12	10:29	63.0
Nov-16-12	9:29	14.0
Nov-16-12	8:29	14.0
Nov-16-12	7:29	27.0
Nov-16-12	6:29	18.0
Nov-16-12	5:29	15.0
Nov-16-12	4:29	24.0
Nov-16-12	3:29	50.0
Nov-16-12	2:29	12.0
Nov-15-12	1:29	12.0
Nov-15-12	0:29	15.0
Nov-15-12	22:29	13.0
Nov-15-12	21:29	16.0
Nov-15-12	20:29	264.0
Nov-15-12	19:46	23.0



Figure 10. A plot of the IMLS1 response time data for seven days showing an average response time of 31.63 ms.

The scarcity of data for this one week of monitoring indicates that the sites were not 100% available due to power issues as indicated by Figure 12. Also, because the sites were still under test, traffic was low.

Figure 13 shows a new reading from both sites. IMLS1 still maintains the lead.

TABLE IV.  
ABRIDGED RESPONSE TIME DATA FOR SEVEN DAYS FROM IMLS2.

Device Name: IMLSNoAgent    Period: Last 7 days  
Start Time: 14-Nov-2012 08:50:29    End Time: 21-Nov-2012 08:50:29

Generated At: Wed, 21 Nov 2012 08:50:31 WAT

Date	Time	Response Time ms
Nov-19-12	9:48	241.0
Nov-17-12	2:30	302.0
Nov-17-12	1:30	235.0
Nov-17-12	0:30	172.0
Nov-16-12	22:30	240.0
Nov-16-12	21:30	259.0
Nov-16-12	20:30	318.0
Nov-16-12	19:30	229.0
Nov-16-12	18:30	268.0
Nov-16-12	17:50	219.0
Nov-16-12	13:28	258.0
Nov-16-12	12:30	294.0
Nov-16-12	11:35	266.0



Figure 11. A plot of the IMLS2 response time data for seven days showing average response time of 253.92ms.

### C. Results from OpManager Network Monitor

From the above results obtained from the **ManageEngine OpManager Monitor**, we observe as follows:

1. That while IMLS1 had an average response time of 31.63ms or less (figures 10 and 13), IMLS2 had an average response time of 253.92ms or higher (figures 11 and 13). This indicates that IMLS1, the agent-based application is at least eight (8) times faster than IMLS2, the non-agent application.
2. That IMLS1 has a better packet data management compared to IMLS2. Figure 12 shows that while IMLS1 has an average packet loss of 25%, IMLS2 has 33.33%.

## VI. CONCLUSION

From the results presented above, we observe that the agent-based application responds eight times faster than its non-agent counterpart. Multi-agent system such as JADE provides a more efficient management of network facilities. This makes agent-based development platform preferable when building future high performance applications for mobile and ubiquitous learning.

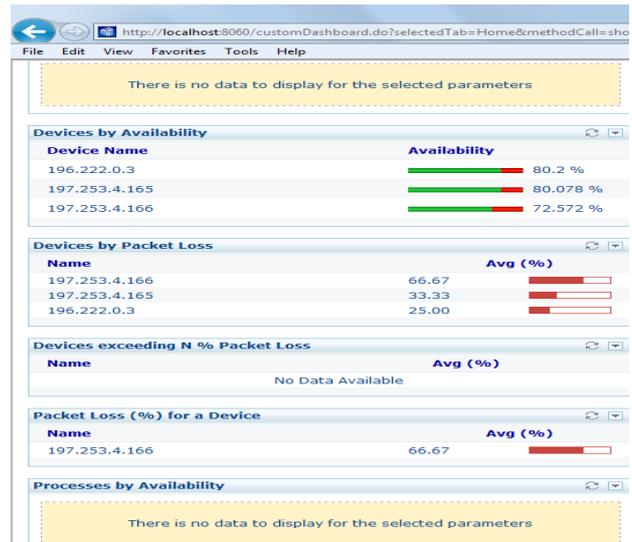


Figure 12. Availability and Packet Loss information of the two sites.

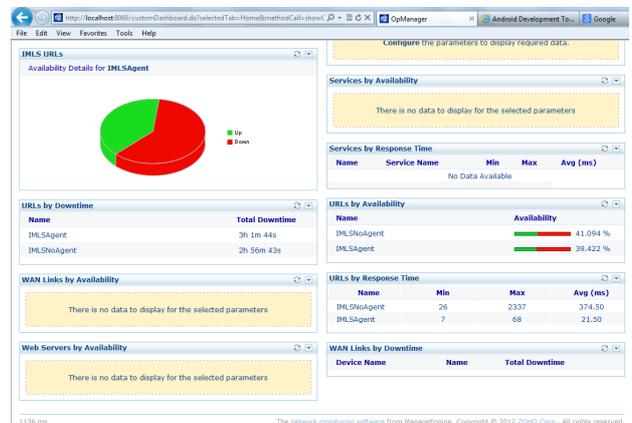


Figure 13. Downtime, Availability, and Response time capture of the two IMLS sites on 22/11/2012.

A number of authors agree that the use of agents have significant improvement on system performance. For example, the use of agents, independently influence the student's motivation and performance in e-learning [22], reorganize resources, thereby, allowing users to discover more and more resources in a shorter amount of time, without increasing the traffic load experienced by distributed hosts [23], show improved communication efficiency [25], achieve better search capability for high-dimensional function optimization [24], and can achieve guaranteed performance consensus [26]. Phuong et al [27] compared the performance of three MTPs with different message encoding schemes and observed that, the HTTP based MTP has the best performance among other protocols for all encoding schemes. This agrees with our finding too, in which the agent-based system (IMLS1) which uses HTTP MTP performed better than the non-agent counterpart.

Agent technology is the next direction for computing, especially the mobile platform that has low resources and runs on wireless ubiquitous networks. Agents have come to stay and to change the client-server approach, as mobile agents allow the processing of information to be done locally [8], instead of transmitting the data over a network thereby reducing the network overhead. This also, explains why the agent-based M-learning application proved to be better in network response time and throughput than

its non-agent version. This is a significant motivation for a user to use an agent-based M-learning platform. It is important to note that the same result can be replicated in other areas of computing besides M-learning when agents are deployed. We advise future developers to begin to look in this direction.

## VII. FUTURE WORK

Our future work will be geared towards developing versions of the agent application that runs across multiple mobile platforms of smart devices. As well as deploying on a reliable host machine that delivers 99.9% uptime to make it available to students and faculties, anywhere, anytime. We shall also conduct user acceptance tests with a view to finding out the users' perception of the application. This test will also help us gain insight into the user's needs, and the possible areas of improvement.

## REFERENCES

- [1] Mike Sharples, Josie Taylor and Giasemi Vavoula (2005). Towards a Theory of Mobile Learning. *International Journal of Continuing Engineering Education and Life Long Learning*, 12(5/6), 504-520. <http://dx.doi.org/10.1504/IJCEELL.2002.002148>
- [2] Collins N. Udanor (2011). An Agent-based Model of Intelligent M-learning System. *International Journal of Science and Advanced Technology*. Volume 1 Number 5. Pages 65-73.
- [3] Collins N. Udanor and T.A. Nwodo (2010). A Review of M-learning Models. *India Journal of Computer Science and Engineering (IJCSE)*. Volume 1 Number 4. Page 426-435.
- [4] Pasi Silander and Anni Rytökönen(2007) An Intelligent Mobile Tutoring Tool Enabling Individualisation of Students' Learning Processes Häme Polytechnic University of Applied Sciences and University of Joensuu, Finland.
- [5] Kassim, A. A., Kazi, S. A. and Ranganath, S (2004). "A Web-based Intelligent Learning Environment for Digital Systems". *International Journal for Engineering Education*, Vol 20, No 1. Pages 13-23, 2004.
- [6] Stuart J. Russell and Peter Norvig (2003). *Artificial Intelligence: A Modern Approach*. Second Edition, Pearson Education, Inc. Upper Saddle River, New Jersey 07458. Bellifemine, F., Poggi, A. and Rimassa, G. (2007). Developing Multi Agent Systems with a FIPA-Compliant Agent Framework. In *Software-Practice and Experience*, John Wiley & Sons, Ltd, Vol. 31, Pages 103-128,
- [7] Kinshuk Taiyu Lin (2004). Improving mobile learning environments by applying mobile agent technology. Massey University, Palmerstone North, New Zealand. Available at: [http://www.col.org/pcf3/papers/pdfs/kinshuk\\_lin\\_2.pdf](http://www.col.org/pcf3/papers/pdfs/kinshuk_lin_2.pdf). viewed 12th March, 2010.
- [8] Effanga, E.N., Asuquo, D. E. and Williams, E. E. (2011). Information Retrieval Using Jade Mobile Agent System. *World Journal of Applied Science and Technology*, Vol.3. No.1. Pages 106-117
- [9] Lavare Henry and Suresh Sankaranarayanan (2010). Intelligent Agent based Mobile Learning System. *International Journal of Computer Information Systems and Industrial Management Applications*. (IJCSIM). <http://www.mirlabs.org/ijcsim>. ISSN: 2150-7988 Vol.2. Pages 306-319
- [10] Floyd Garvey and Suresh Sankaranarayanan (2012). Intelligent Agent based Flight Search and Booking System. (IJARAI) *International Journal of Advanced Research in Artificial Intelligence*, Vol. 1, No. 4. Pages 12-28. <http://dx.doi.org/10.14569/IJARAI.2012.010403>
- [11] R Sasikumar and D Manjula.(2012) : Intrusion Detection System Designed for Wireless using JADE Mobile Agent Framework. *International Journal of Computer Applications*: Vol. 50 NO. 15. Pages 23-27, Published by Foundation of Computer Science, New York, USA.
- [12] Damien Meere, Ivan Ganchev, Máirtín O'Drónma(2011). Mobile Phones as Learning Instruments: mLearning Service Provision within an InfoStation-based Multi-Agent Environment. 2011 International Conference on Telecommunication Technology and Applications Proc. of CSIT Vol.5 (2011) IACSIT Press, Singapore
- [13] Agent-Based Models of Industrial Ecosystems. Rutgers University. October 6, 2003. Available at: <http://policy.rutgers.edu/andrews/projects/abm/abmarticle.htm>.
- [14] Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*. May 14, 2012.
- [15] FIPA. FIPA Abstract Architecture Specification. Foundation for Intelligent Physical Agents. Available at: <http://www.fipa.org/specs/fipa00001/>
- [16] [17] Gray Watson. ORMLite. Available at: <http://ormlite.com/javadoc/ormlite-core/doc-files/ormlite.html>
- [17] Peter Sevcik and John Bartlett: NetForecast Inc. 2001 research report, Understanding Web Performance. Available on: [www.netforecast.com](http://www.netforecast.com), viewed on 5<sup>th</sup> January, 2011.
- [18] Alberto Savoia: Web Page Response Time 101. Published in the July/August 2001 edition of STQE, the software testing and quality engineering magazine. Pages 49-53. Available at: [www.stqemagazine.com](http://www.stqemagazine.com)
- [19] NetForecast report2006, Field Guide to Application Delivery Systems. Available at: <https://www.pdfFiller.com/edit.php?id=4914763>
- [20] Vlado Glavinic, Marko Rosic, Marija Zelic (2007): Agents in m-Learning Systems Based on Intelligent Tutoring. *Universal Access in Human-Computer Interaction. Applications and Services Lecture Notes in Computer Science Volume 4556*, 2007, pp 578-587 [http://dx.doi.org/10.1007/978-3-540-73283-9\\_64](http://dx.doi.org/10.1007/978-3-540-73283-9_64)
- [21] Youssef Shiban, Iris Schelhorn, Verena Jobst, Alexander Hörnlein, Frank Puppe, Paul Pauli, and Andreas Mühlberger: The appearance effect: Influences of virtual agent features on performance and motivation *Computers in Human Behavior*, Volume 49, August 2015, Pages 5-11 <http://dx.doi.org/10.1016/j.chb.2015.01.077>
- [22] Yunsong TAN, Jianjun HAN, and Yuntao WU2 (2010): A Multi-agent Based Efficient Resource Discovery Mechanism for Grid Systems. *Journal of Computational Information Systems* 6:11 (2010) 3623-3631.
- [23] SHI Xuhua and QIAN Feng (2012): A Multi-agent Artificial Immune Network Algorithm for the Tray Efficiency Estimation of Distillation Unit *Chinese Journal of Chemical Engineering*, 20(6) 1148-1153 (2012).
- [24] Mary Berna-Koes, Illah Nourbakhsh, Katia Sycara. *Communication Efficiency in Multi-Agent Systems*. Available at: <http://www.cs.cmu.edu/~illah/PAPERS/usaricra.pdf>
- [25] Zhi-Hong Guan, Bin Hu, Ming Chi, Ding-Xin He, Xin-Ming Cheng (2014): Guaranteed performance consensus in second-order multi-agent systems with hybrid impulsive control. *Automatica* 50(2014)2415-2418 <http://dx.doi.org/10.1016/j.automatica.2014.07.008>
- [26] Phuong T. Nguyen, Volkmar Schau and Wilhelm R. Rossak. Performance Comparison of some Message Transport Protocol Implementations for Agent Communication. Available at: <http://subs.emis.de/LNI/Proceedings/Proceedings186/193.pdf>. Viewed 15th May, 2015.
- [27] Poslad, S. 2007. Specifying protocols for multi-agent system interaction. *ACM Trans. Autonom.*
- [28] *Adapt. Syst.* 2, 4, Article 15 (November 2007), 24 pages. DOI = 10.1145/1293731.1293735
- [29] Collins N. Udanor and O.U. Oparaku, A Model of Intelligent Mobile Learning System using Multi-Agent Systems, ISBN 978-3-659-44209-4, Germany, Lambert Academic Publishing, 2013, pages 93-106.

## AUTHORS

**Collins N. Udanor** is with the Department of Computer Science, University of Nigeria, Nsukka, Nigeria ([collins.udanor@unn.edu.ng](mailto:collins.udanor@unn.edu.ng)).

**O.U. Oparaku** is with the Department of Electronic Engineering, University of Nigeria, Nsukka, Nigeria ([ouoparaku@yahoo.com](mailto:ouoparaku@yahoo.com)).

Submitted 20 July 2015. Published as resubmitted by the authors 20 September 2015.