International Journal of Advanced Computer Science and Information Technology (IJACSIT)
Vol. 3, No. 3, 2014, Page: 263-272, ISSN: 2296-1739

www.elvedit.com

An Agent Driven M-learning Application

Authors

Collins N. Udanor
Faculty of Physical Sciences, Department of Computer Science, University of Nigeria
collins.udanor@unn.edu.ng
Nsukka,, Nigeria

O.U. Oparaku
Faculty of Engineering, Department of Electronic Engineering, University of Nigeria.
ouoparaku@yahoo.com
Nsukka,, Nigeria

Abstract

The future of the web is on mobile devices. Application users have migrated from the desktop to the web. Now the next stage of the Web will be building apps and mobile UIs on top of our collective data. On the part of developers, application development is moving from object-oriented development to agent-oriented programming. This paper presents a fusion of these two trends in computing. The need for ubiquitous access to information and communication, as well as the portability of devices has prompted a lot of research interests in mobile technologies. One of such recent interests is in the field of mobile learning (M-learning), an offshoot of the more established e-learning. This paper presents the development of a multi-agent driven m-learning application using the Java Agent Development Environment (JADE).

Key Words

Agent, JADE, IMLS, M-learning

I. Introduction

The need to re-conceptualize learning in the mobile age has given rise to intensive research work on Mobile learning. In recent times, this need has also propelled researchers to recognize the essential role of mobility and communication in the process of learning. Mobile learning is the delivery of learning contents in mobile devices over wireless infrastructures. M-learning provides learning anywhere, anytime. An agent, on the other hand, is essentially a special software component that has autonomy and provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda [1][12].

This paper demonstrates how a mobile learning application based on the model of Intelligent Mobile Learning System (IMLS) [2] is developed.

II. Review of Related Work

The IMLS model is based on a modified Silander's and Kazi's models [3][11][4], as well as a reference to Kinshuk's Bee-gent framework[5]. Unlike the Bee-gent[6] multi-agent framework used by the author in [5], this work makes use of the JADE multi-agent framework, which is well researched and is gaining popularity among many researchers in agent technology. JADE is developed by Telecom Italia and offered as free software. JADE ensures consistent connection and access to M-learning contents even at low bandwidths by using bit efficient transmission technique.

Berking[7], instead of creating a new Instructional Design (ID) model, presented a framework that can be used to incorporate mobile learning considerations into existing ID models (which optimize them for the paradigm of "anywhere, anytime" mobile learning. Rather than focus on lists of specific design considerations for mobile, they created a framework that provides an organizing principle for these design considerations. Within this framework, they called out the learning theory that underlies the mobile learning strategy as an important determinant of considerations for a new or existing ID model. They stated that the flexible approach proposed by their framework takes both instruction and performance support into consideration for the mobile learning task. Botzer[8], developed a Math for Mobile application that reflected the socio-cultural and situated learning model of mobile learning. They found that the contribution of the mobile environment lies not only in making dynamic mathematical application more available, but also in supporting the execution of mathematical tasks that are closer to the students' experiences and more relevant to them, which has the potential to enhance experiential learning. The authors [9] opined that the quality of the M-learning application represents an important aspect for the education process because it affects the way the information is understood and is learned by users. They presented M-learning application metrics. Among the utilized models of measuring the quality level of M-learning applications were the indicators such as:

Dimension of occupied space, access count of a page or topic, and number of pages read in a working session.

III. JADE Agent Based M-Learning Application Development

In the following sections, we shall show the design and implementation of the agent-based M-learning application developed during our research work.

Figure 1 shows the architecture of the model of Intelligent Mobile Learning System (IMLS). The model is a multi-tier architecture, made up of five sub-units (tiers). These are: the client, the web service, the JADE agents, the IMLS Model and the ORMLite data store.

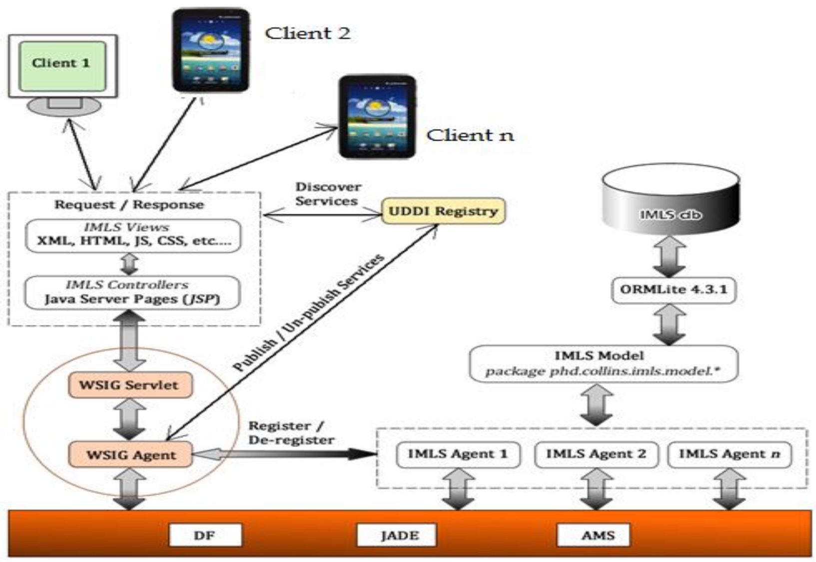

FIGURE 1: MODEL OF INTELLIGENT AGENT MOBILE LEARNING SYSTEM (IMLS)

A. The Client

This can be any mobile device such as smart phones, PDAs, tablet PCs, etc. For the prototype application based on this model, the client is built using the Android smart phone. The mobile device connects to the Internet using a WAP browser on basic HTML, XML, Java scripts and Cascaded Styling Sheet (CSS) for presentation. The Internet provides connection to the server end of the model. XML helps to communicate with the agents in the server to collect data and return the results to the calling session. JavaServer Pages (JSP) technology handles the business logic. JSP is well on its way to becoming the preeminent Java technology for building applications that serve dynamic Web content [10].

It is a server-side scripting language. The biggest advantage of using JSP is that it helps effectively separate presentation from content. Architecturally speaking, JSP can be viewed as a high-level abstraction of servlets that is implemented as an extension of the Servlet 2.1 API.

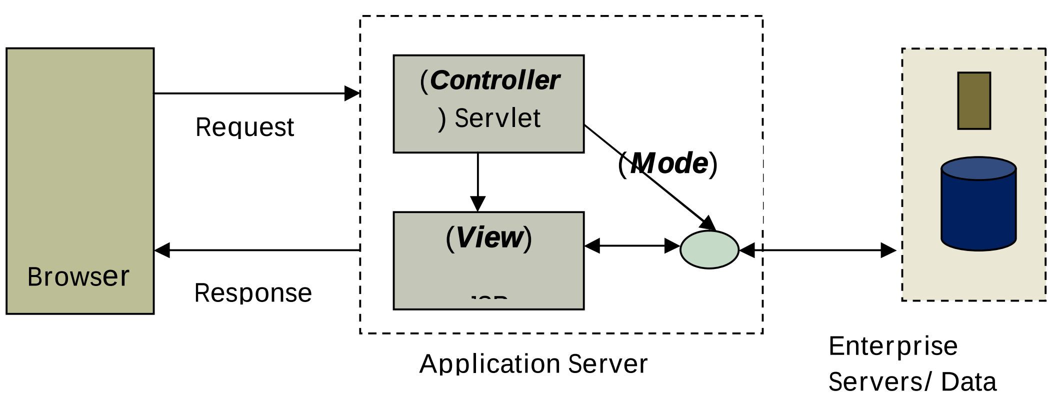

FIGURE 2: JSP MODEL 2 ARCHITECTURE

The early JSP specifications advocated two philosophical approaches for building applications using JSP technology. These approaches, termed the JSP Model 1 and Model 2 architectures, differ essentially in the location at which the bulk of the request processing was performed. In the Model 1 architecture, the JSP page alone is responsible for processing the incoming request and replying back to the client. There is still separation of presentation from content, because all data access is performed using beans.

In the IMLS model, the Model 2 architecture, shown in figure 2, is employed. The model 2 is a hybrid approach for serving dynamic content, since it combines the use of both servlets and JSP. It takes advantage of the predominant strengths of both technologies in using JSP to generate the presentation layer and servlets to perform process-intensive tasks. There is still separation of presentation from content, because all data access is performed using beans. Here, the servlet acts as the controller and is in charge of the request processing and the creation of any beans or objects used by the JSP, as well as deciding, depending on the user's actions, which JSP page to forward the request to. Note particularly that there is no processing logic within the JSP page itself; it is simply responsible for retrieving any objects or beans that may have been previously created by the servlet, and extracting the dynamic content from that servlet for insertion within static templates.

B. The Server, Web Services Producer

This is a Web server, typically behind firewalls and/or proxy gateways. The Web Services Integration Gateway (WSIG), an add-on component to the JADE platform provides an interface between the JSP client and the agent world [12]. The objective of WSIG is to expose services provided by agents and published in the JADE Directory Facilitator (DF) as web services with no or minimal additional effort, though giving developers enough flexibility to meet specific requirements they may have. The process involves the generation of a suitable Web Service Description Language (WSDL) for each service-description registered with the DF and possibly the publication of the exposed services in a UDDI registry. The WSIG add-on supports the standard Web services stack, consisting of WSDL for service descriptions, Simple Object Access Protocol (SOAP) message transport and a UDDI repository for publishing Web services. The WSIG web application consists of two main elements:

- WSIG Servlet
- WSIG Agent

The **WSIG Servlet** is the front-end towards the internet world and is responsible for:

- Serving incoming HTTP/SOAP requests
- Extracting the SOAP message
- Preparing the corresponding agent action and passing it to the WSIG Agent.

- Converting the action result into a SOAP message
- Preparing the HTTP/SOAP response to be sent back to the client

The WSIG Agent is the gateway between the Web and the Agent worlds and is responsible for:

- Forwarding agent actions received from the WSIG Servlet to the agents. It is actually able to serve them and get back responses.
- Subscribing to the JADE DF to receive notifications about agent registrations and de-registrations.
- Creating the WSDL corresponding to each agent service registered with the DF and publishes the service in a UDDI registry if needed.

Two main processes are continuously active in the WSIG web application:

- The process responsible for intercepting DF registrations/deregistrations and converting them into suitable WSDLs. As mentioned, this process is completely carried out by the WSIG Agent.
- The process responsible for serving incoming web service requests and triggering the corresponding agent actions. This process is carried out jointly by the WSIG Servlet (performing the necessary translations) and the WSIG Agent (forwarding requests to agents able to serve them).

JADE agents publish their services in the DF providing a structure called DF-Agent-Description as defined by the FIPA specification (www.fipa.org) [13]. A DF Agent-Description includes one or more Service-Descriptions, each one actually describing a service provided by the registering agent. A Service-Description typically specifies, among others, one or more ontologies that must be known in order to access the published service.

The actions the registering agent is actually able to perform are those defined in the specified ontologies.

C. The IMLS Model sub-module

The model consists of the components that make up the Intelligent Tutoring/Learning Systems. These include the student model, expert module, pedagogical module and the communication or user interface module as depicted in figure 3. The model contains the actual course content, the student records, assessment, as well as administrative information. These contents are stored in the data store using the ORMLite. To interact with these contents, the respective agents are invoked, such as the authentication agent, tutorial agent, assessment and user agents.

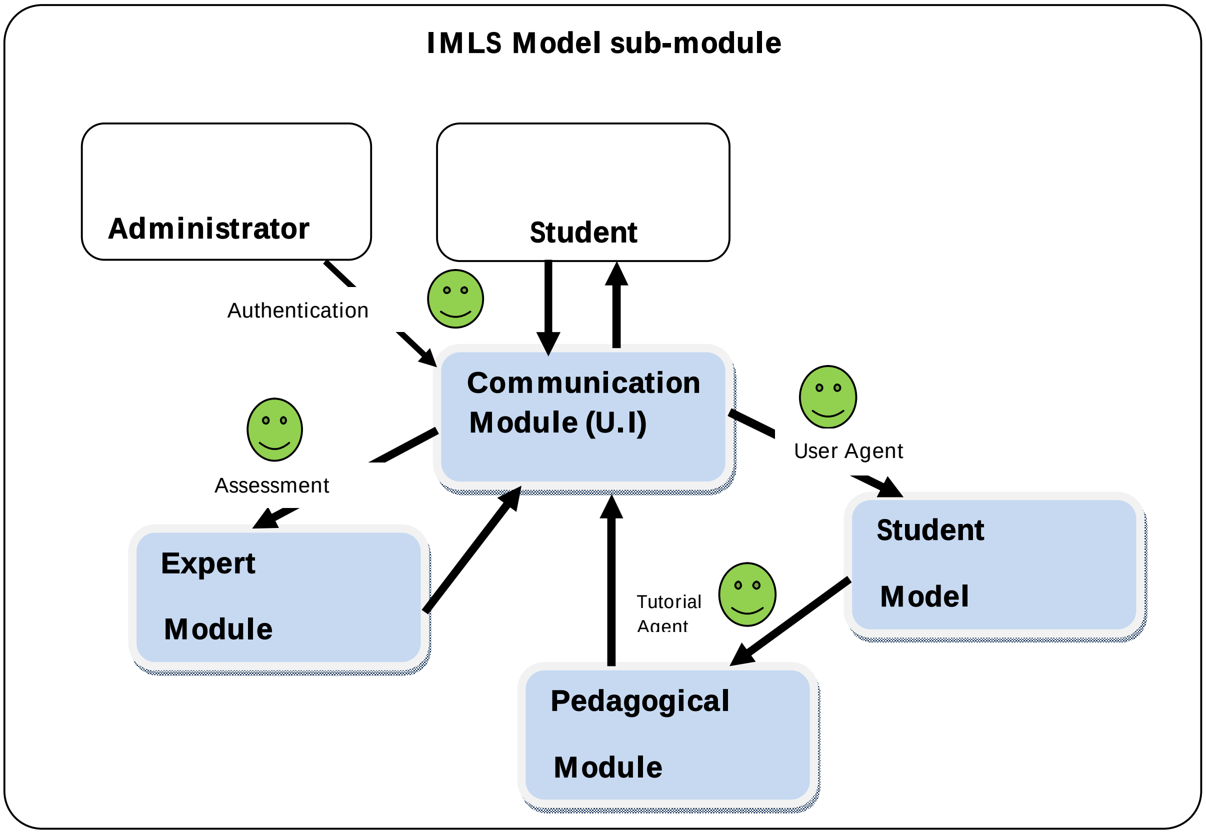

FIGURE 3: IMLS MODEL SUB-MODULE

D. The JADE Agent

The system relies heavily on multi-agents, which ensuring effective bandwidth utilization optimizes the processing time. 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 its own agenda. The IMLS model in figure 1 also contains the JADE framework which consists of the Agent Management System (AMS) and DF, in addition to the IMLS agents created by the programmer. These agents include the Authentication agent, User agent, Tutorial agent and Assessment agent. The Authentication Agent manages user accounts, privileges and sessions. Before the server starts running, the JADE platform is first booted with the AMS and the DF on port 1099 as shown on the left panel in figure 4.

The user agent keeps track of each user's activities after the user has been authenticated. The agent stores session information, tutorial sessions and assessment information. Figure 5 shows a user login and a user session on a web browser. A sniffer agent in the JADE platform keeps track of the interactions between the various agents, as shown on the right panel of figure 4. The tutorial agent delivers the course materials to the user, while the assessment agent delivers test questions to determine the learner's performance. The assessment could be a pre-lesson or post lesson assessment.

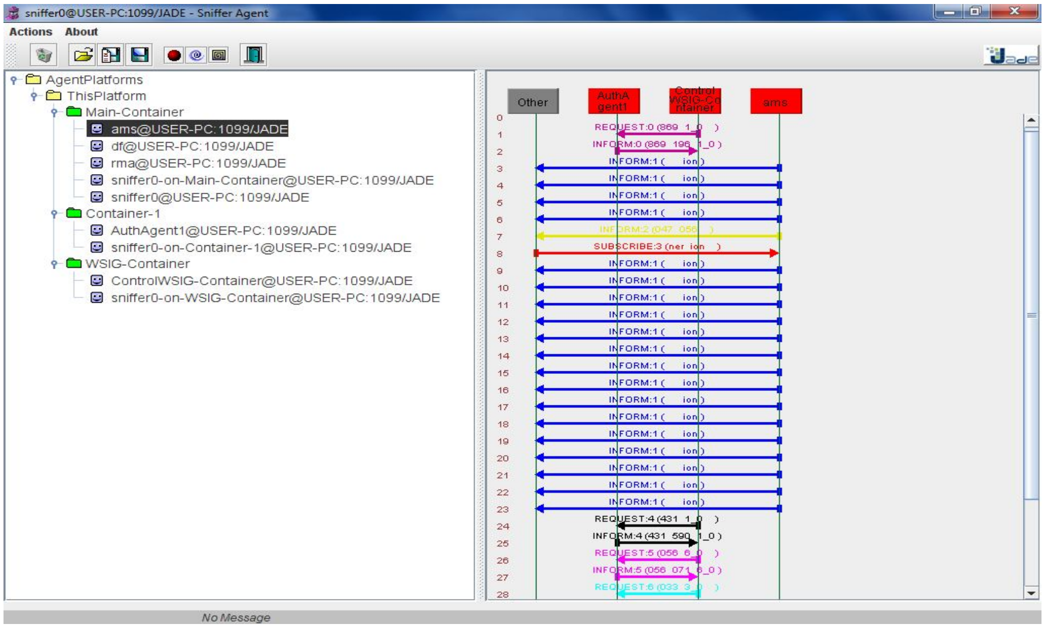

FIGURE 4: JADE PLATFORM RUNNING THE AUTHENTICATION AGENT AND WSIG SERVLET.

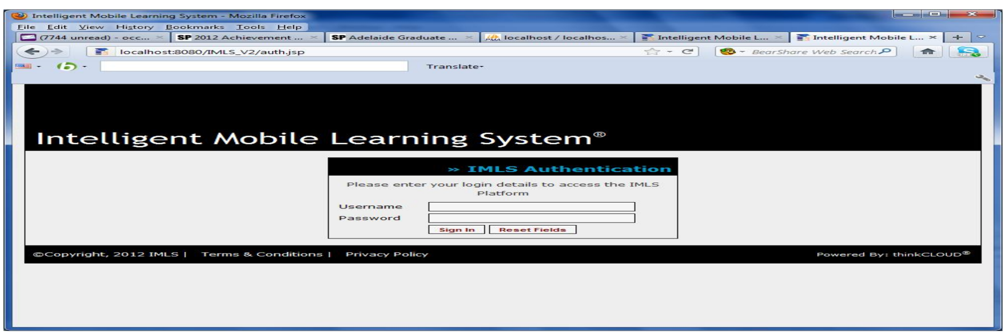

FIGURE 5: A USER SESSION

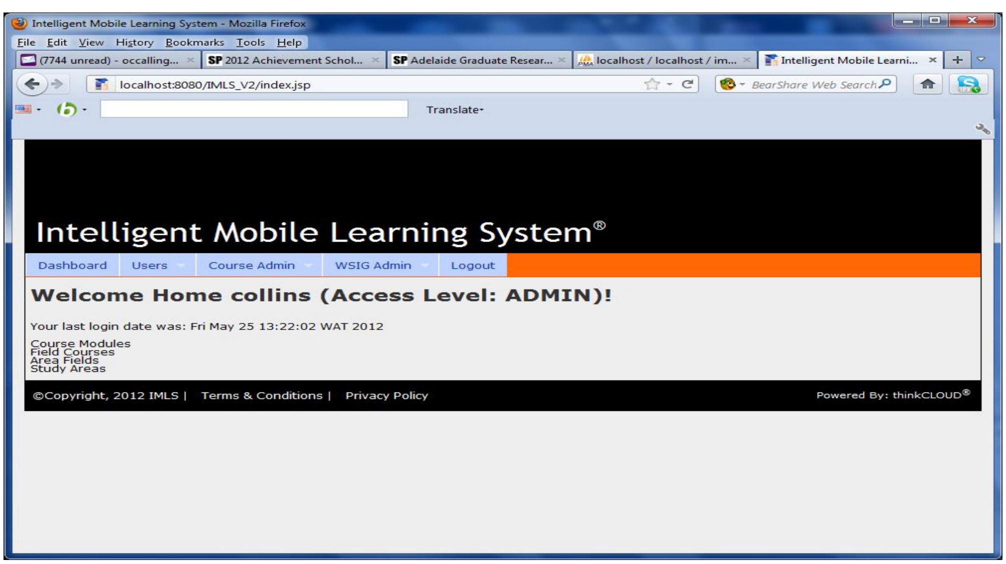

FIGURE 6: ADMINISTRATOR BACKEND

E. The Data Store

This unit is made of the Object Relational Mapping, and a relational data base management system such as MySql, MS-SQL Server, Oracle, etc. This application uses a MySql database.

ORMLite 4.3.1

ORMLite provides a lightweight Object Relational Mapping between Java classes and SQL databases [14]. There are certainly more mature ORMs which provide this functionality including Hibernate and iBatis. ORMLite supports JDBC connections to MySQL, Postgres, H2, SQLite, Derby, HSQLDB, Microsoft SQL Server, and can be extended to additional ones relatively easily. ORMLite also supports native database calls on Android OS. There are also initial implementations for DB2, Oracle, generic ODBC, and Netezza.

F. The Network

This refers to the wireless and wired networks, part of the Internet, and the communication protocols. The network (Internet) connects the mobile device (web service consumer) to the Web services producer via SOAP/XML. It is important to note that JSR 172 [15] does not mandate the use of XML encoding on the device itself, allowing implementations (as long as they are transparent to both consumer and producer) to use more efficient encoding approaches, such as the use of binary protocols between the device and the wireless gateway.

IV. Conclusion

Agent-oriented application development promises the delivery of software models that consist of dynamically interacting rule-based agents. The systems they interact in can create real-world-like agents that are complex, intelligent and purposeful. This system simulates the real world classroom scenario with the tutorial agent acting as a class teacher. Agents have been found to be very useful in complex systems such as data mining, ecological sciences, life and social sciences.

References

[1] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood (2007). *Developing Multi-Agent Systems with JADE*. John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex PO19 8SQ, England.

[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] Pasi Silander , Anni Rytkönen (2007) *An Intelligent Mobile Tutoring Tool Enabling Individualisation of Students' Learning Processes.* Häme Polytechnic University of Applied Sciences and University of Joensuu, Finland (pasi.silander@hamk.fi) Department of Computer Science, University of Helsinki, Finland (anni.rytkonen@cs.helsinki.fi)

[4] Sabbir Ahmed Kazi (2005). *VocaTest: An Intelligent Tutoring System for Vocabulary Learning using the "mLearning" Approach.* Centre for Research in Pedagogy and Practice National Institute of

Education. Available on: http://conference.nie.edu.sg/paper/Converted%20Pdf/ab00283.pdf. Viewed on 20th March, 2010.

[5] Kinshuk, Taiyu Lin (2004). Improving mobile learning environments by applying mobile agent technology. Massey University, Palmerstone North, New Zealand. Available on: http://www.col.org/pcf3/papers/pdfs/kinshuk_lin_2.pdf. viewed 12th March, 2010.

[6] Toshiba (2005). *Multi-Agent Framework for 100% Pure Agent System* .Corporate Research & Development Center TOSHIBA Corporation. Available on: http://www.toshiba.co.jp/rdc/beegent/index.htm. Viewed on 21st April. 2010.

[7] Peter Berking, Jason Haag, Thomas Archibald, Marcus Birtwhistle (2012). *Mobile Learning: Not Just Another Delivery Method*. Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) 2012

[8] Galit Botzer, Michal Yerushalmy2007). *Mobile Application for Mobile Learning.* (IADIS International Conference on Cognition and Exploratory Learning in Digital Age (CELDA 2007)

[9] Catalin Boja, Lorena Batagan (2009). Software Characteristics of M-Learning Applications. *WSEAS Transactions on Computers*, Volume 8, Issue 5, Pages 767-777

[10] Govind Seshadri. Understanding JavaServer Pages Model 2 architecture exploring the MVC design pattern. See:http://www. JavaWorld.com, viewed on 18th June, 2012.

[11] Kassim, A. A., Kazi, S. A. and Ranganath, S (2004)., “A Webbased Intelligent Learning Environment for Digital Systems”. *International Journal for Engineering Education*, Vol 20, No 1, pp 13-23.

[12] Stuart J. Russell and Peter Norvig (2003). *Artificial Intelligence: A Modern Approach*. Second Edition, Pearson Education, Inc. Upper Saddle River, New Jersey 07458.

[10] Bellifemine, F., Poggi, A. and Rimassa, G. Developing Multi Agent Systems with a FIPA-Compliant Agent Framework. In Software–Practice and Experience, John Wiley & Sons, Ltd, vol. 31, pp. 103–128,

[13] [FIPA00001] FIPA Abstract Architecture Specification. Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00001/

[14] Gray Watson. OrmLite -- *Simple Object Relational Mapping (ORM) Java Package* **.** Available at**:** http://256.com/gray/ simple_orm_java.shtml. Viewed on 16th June, 2012.

[15] C. Enrique Ortiz, *Web Services APIs for J2ME, Part 1*: Remote service invocation API. Available on: www.J2MEDeveloper.com. Viewed on 5th January,2011.

AUTHORS' BIOGRAPHY

Dr. Collins Udanor has his B.Eng in Computer Science and Engineering, 1996, an M.Sc in Computer Science (Data Communication option), 2004 and a PhD in Electronic Engineering, University of Nigeria in 2013 (Digital Computer Systems option, with specialization in Multi-agents based Mobile Learning). He has worked over a decade as both a Technical Instructor and as an Academic. His research interests are in the areas of intelligent mobile learning systems, Distributed Computing Infrastructures and High Performance Computing (DCI/HPC). Over the years he has been involved in teaching, research, workshop organization and presentations. He has also been richly involved in software development projects of various capacities and on varying platforms. He has also designed and implemented Distributed Computing Infrastructures (DCIs), notably the LIONGRID-UNN-NG.

Prof. Ogbonna Ukachukwu Oparaku is a 1980 graduate of Electrical/Electronic Engineering from the University of Nigeria. He worked with General Electric Company (Telecomminications Division) Nigeria Limited between 1981 and 1983 as a Special Projects Engineer before joining the Universty of Nigeria, on parallel appointment in the Electronic Engineering Department and the Energy Research Centre. He obtained a PhD in Solid State Electronics from the University of Northumbria at Newcastle – Upon-Tyne, United Kingdom, in 1988. He became a Professor in 2003 and was appointed Director of the National Centre for Energy Research and Development in 2004, a position he occupied till October 2009. He is currently the Head of Department of Electronic Engineering, University of Nigeria, Nsukka.