Towards interactive open source based m-Learning solution: the *m-Chisimba* framework

Received 29 August 2013
Revised 29 August 2013
Accepted 17 September 2013

Nhlanhla Wilton B. Mlitwa

Department of IT, Faculty of Informatics and Design, Cape Peninsula University of Technology (CPUT), Cape Town, South Africa, and

David Wafula Wanyonyi

e-Learning, Support and Innovation Unit (eLSI), Knowledge and Information Management, University of Witwatersrand, Johannesburg, South Africa

Abstract

Purpose – This paper aims to present a research outcome to an investigation towards the innovative use of electronic-learning (e-Learning) software solutions to strengthen the interactive component of mobile learning (m-Learning) in university contexts. This work was motivated by observations of interactive limitations in existing m-Learning solutions in well-intentioned software development and e-Learning efforts. The aim of the project, thus, was to interrogate hindrances to the potency of m-Learning interventions. The idea was to strengthen the interactive component, to improve the compatibility of the desktop e-Learning process – with mobile appliances such as mobile phones and, finally, to explore cost minimizing avenues of downloading learning materials over mobile phones.

Design/methodology/approach – The methodology of the project was twofold: that is, both literature-based and experimental. Literature background offered insight on existing efforts towards interactive m-Learning solutions. Against this background, the project was then designed to explore ways to provide presentations and interactive chats for effective m-Learning over mobile phones. The idea was to reduce the file size of presentations slides for effective display on mobile devices. Finally, the aim was also to minimize airtime costs when transmitting presentation slides and when conducting interactive chats on mobile phones. An experiment research method was, thus, used. In this method, a problem and question were articulated first. Then, a practical technological solution was conceptualized, developed, pre-tested, modified and re-tested until it complied with sought specifications, Finally, the solution was implemented, to confirm its viability and response to a research problem and, ultimately, was reported in the findings.

Findings – Slides were developed with light colour resolutions to open effortlessly on tiny mobile screens. The best quality was between 128 and 201.188 (out of 255) pixels (more visible mobile phone screens), using Open Source Software. Clear slides also had a font greater than size 32. The average data per presentation were 9,872.66 bytes, and, when seven learners were chatting, it was 3,686.63 bytes (0.03 megabytes). For airtime costs, the Vodacom rate of R0.46 per megabyte was used. This translates to 0.8 0.46 R0.0368 cost per slide. The *m-Chisimba* platform therefore, seems to be cost-effective, quality expedient and most appropriate for interactive m-Learning.

Research limitations/implications – Existing m-Learning has been inadequate, failing to align adequately the desktop e-Learning solutions with mobile phone platforms. Desktop presentation slides, for example, have been incompatible with mobile phones' operations and screen sizes. Mostly, content could consume a lot more airtime, owing to large sizes of desktop originated material that was incompatible with mobile platforms. A solution that emerged from original experiments, leading to the enhanced interactive m-Learning solution, the *m-Chisimba* platform, is, therefore, presented in this paper. The thinking is that mobile applications are convenient enablers which extend access to information. Nevertheless, mobile applications (including m-Learning solutions) cannot be seen as panaceas to solve all learning concerns as literacy and affordability remain challenges in many under-developed social environments. As a newly developed innovation, the *m-Chisimba* platform offers an improved m-Learning solution for learning activities over mobile gadgets such as mobile phones.

Practical implications – The research work offers new innovative solutions in a number of practical ways. It improves the interactivity component of the m-Learning process, to empower educators and learners to effectively engage learning over m-Learning devices.

Social implications – The innovation makes it easier to for learning to take place effortlessly, over mobile platforms. Implications are that more people will be able to access learning processes on the move, meaning that obstacles to access learning opportunities are minimised.

Originality/value – Existing m-Learning has been inadequate, failing to align adequately the desktop e-Learning solutions with mobile phone platforms. Desktop presentation slides, for example, have been incompatible with mobile phones' operations and screen sizes. Mostly, content could consume a lot more airtime, owing to large sizes of desktop-originated material that was incompatible with mobile-platforms. A solution that emerged from original experiments, leading to the enhanced interactive m-Learning solution, the *m-Chisimba* platform, is, therefore, presented in this paper. The thinking is that mobile applications are convenient enablers which extend access to information. Nevertheless, mobile applications (including m-Learning solutions) cannot be seen as panaceas to solve all learning concerns as literacy and affordability remain challenges in many under-developed social environments. As a newly developed innovation, the *m-Chisimba* platform offers an improved m-Learning solution for learning activities over mobile gadgets such as mobile phones.

Keywords Design, Implementing, Managing and practicing innovation, Information and communication technologies, Education practices, Knowledge management, Effectiveness, Alternative and new technologies

Paper type Research paper

1. Introduction

From the National Systems of Innovations (NSI) initiatives, to the United Nations Millennium Development Goals (UNMDG), information technology (IT) is recognised as an enabler of efficiencies in almost all aspects of modern life (Mlitwa, 2011). In effect, information and communication technology (ICT) has become a catalyst for economic inclusion, development and social change in the network society (Castells, 2000) of the twenty-first century.

In education, ICT has become a major enabler of teaching, learning and communication efficiencies (Ncubukezi, 2013; Nonyane, 2011). Arguably, networked technology has simplified the development and the exchange of audio, video and textual data in the information age (Swearingen, 2004). With the mobility component, in particular, flexibility to access information and to interact anytime and from anywhere, information-handling processes have become not only more than just efficient but also convenient. The Y2K generation (persons born during the information-age period, from 2000) can interact synchronously for private, societal, educational and business purposes (Kekwaletswe, 2007; O'Neill *et al.*, 2005), regardless of geographical location (Wu *et al.*, 2008) and distance (Mlitwa, 2011). The benefits of mobile technology have since become prominent in mobile learning (m-Learning) within the domain of electronic learning (e-Learning) in university contexts (Wanyonyi, 2008).

Despite numerous advantages of technology in education, e-Learning (including m-Learning) is yet to make a notable impact on learning processes (Bates, 2009). Whilst e-Learning adoptions in academia have grown exponentially, the impact of such attempts on learning outcomes remains invisible if not evasive (Bates, 2009; CCL, 2009; ITC, 2008; WEF, 2008; White, 2008). In fact, Bates (2009) goes as far as to declare a complete failure of electronic technologies to make an impact in universities. Shortsightedness in implementation frameworks is cited as the main problem. For example, whilst cyber attacks are a major concern for systems and applications' developers (Ramin and Levi, 2006) collaborative tools as well as knowledge sharing and transfer capabilities in e-Learning environments have been less than potent (Bates, 2009; Canadian Council on Learning, 2009; World Economic Forum, 2008; Instructional Technology Council, 2008). In the context of non-contact (online and distant) learning, which lacks the face-to-face component, the learning process is further compromised by these weaknesses.

Incoherence between e-Learning (including m-Learning) and pedagogical considerations has been another serious impediment (Mlitwa, 2011). Whilst technology solutions to improve interactive components in fixed and mobile solutions exist, shortsightedness in relevant developments and implementations continue to limit the potency in learning. Unless the status quo is amended, the impact of e-Learning solutions on the learning outcomes may remain sub-marginal.

1.1 Objective

This paper builds on the established thesis that non-contact learning decreases the level of contact (direct/physical interaction) between learners and their teachers (Mlitwa, 2011). It is also understood that the innovative use of different software to advance the interactive component of fixed or m-Learning systems is possible. Thus, this paper presents a framework for innovative use of e-Learning software solutions to strengthen the interactive component of m-Learning in university contexts. The aim is to interrogate hindrances to the potency of m-Learning interventions and to build on an existing Chisimba e-Learning framework as practical way to develop a potent way forward.

In this quest, the idea is to strengthen the mobile component in electronic learning solutions, so as to improve user experiences and learning outcomes (Eom *et al.*, 2006) on m-Learning. The open source-based initiative, the *m-Chisimba* framework, is presented as an interactive m-Learning alternative. The paper opens with a discussion of m-Learning (its description and benefits to learning), as well as a motivation for the study, a research problem, the objective and methodology. The basis of the *m-Chisimba* (Wanyonyi, 2008) framework is then introduced, with resultant interactive m-Learning solutions discussed and demonstrated. Conclusions and recommendations conclude the paper.

2. The m-Learning phenomenon

Whilst e-Learning refers to the use of a combination of electronic tools, systems and procedures to facilitate teaching and learning processes over networked platforms (Pavlović and Seljan, 2008), m-Learning refers to the use of portable and often wireless technology tools and systems in devices such as mobile phones, iPods (Kahiigi *et al.*, 2008), palmtops and laptops or simply, just mobile information appliances (IAs), to facilitate learning (Quinn, 2000) across time and space (Mlitwa, 2011; Wanyonyi, 2008). Whilst electronic communication technology improves efficiencies in education, and simplifies educational (teaching, learning and course-administration) processes, the mobile component adds flexibility, independence and convenience, thereby improving learning experiences for students (Laurillard, 2008).

In fact, m-Learning is an innovative extension of e-Learning with a mobile touch (Mlitwa, 2011). By offering a direct interaction between a mobile device and an e-Learning system, it enables access to e-Learning resources through mobile computing devices, wherever a learner may be located (Wanyonyi, 2008). Access is possible anywhere where a learner can take an "m-Learning-enabled" mobile IA such as a mobile phone, for example, a train, bus, cafeteria or even the toilet, and use it to access information and interact with others anytime, thereby making learning more flexible and convenient (Kekwaletswe, 2007). The phrase "m-Learning enabled" suggests that such mobile appliances need specific configurations, equipping them with enabling features and applications to facilitate learning anywhere. These m-Learning-enabled tools vary from the most basic non-interactive, to the modern and most interactive models. At the very basic level is a text-based model mostly using the Global System for Mobile (GSM) communications networks short message system (SMS) (Wen *et al.*, 2007; Peersman *et al.*, 2000). At the second and slightly advanced level are podcasting- and image-based models, followed by the rich Internet applications model for m-Learning and, finally, the mobile-based learning management systems.

2.1 Basic text-based m-Learning model

The basic reliant text-based model is illustrated in Figure 1.

In a text-based model, Figure 1 illustrates an SMS-based application. In this example, a query is sent to a GSM module, which forwards it to a dialog control module in the appropriate format. The dialog control module then generates a query task for the query-processing module, which searches for information in a knowledge base and

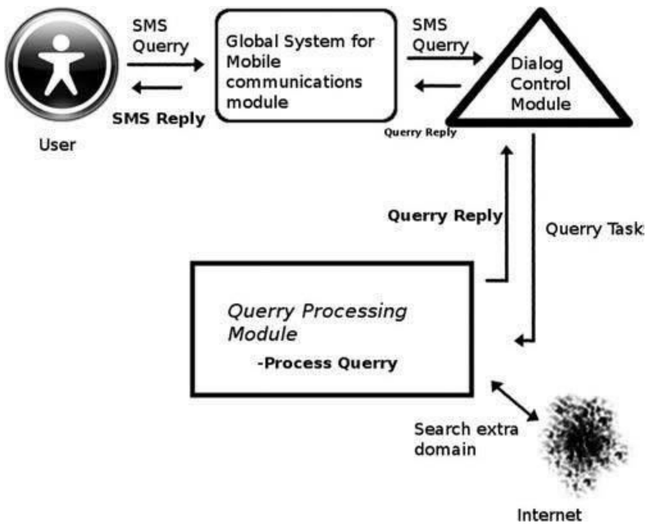

Figure 1.
SMS-based m-Learning application

the Internet if necessary, and gives an appropriate reply to the user. This model of m-Learning system is useful in facilitating timely responses to queries, but falls short on the mix of data it can handle, and also lacks the interactive component. As a text-based format, for example, it can only handle text but not audio, video or picture content. Further, it cannot extend its query function beyond a specific, pre-programmed knowledge base to which it is linked, with no potential for interaction between teaching and learning parties (Wanyonyi, 2008). A slightly different, but also SMS based-model by Ramados and Balasundaram (2007) has similar limitations. Whilst it uses the question-answering model – based on a simple matching algorithm to extract key words form the answer – it can only handle textual (and no audio or video) data and has no capacity to facilitate interactive component. On the other hand, social networks such as Facebook closed groups, MXit (www.mxit.co.za) and Google Talk, among others, also offer text-based solutions with an interactive advantage of chat-room discussions over an information appliance such as a mobile phone. As learning platforms, nevertheless, these facilities are also limited to specific "purpose-based" applications. In the case of MXit, for example, a special "bot" application was built to send a specific set of educational information to anyone who sent a blank message (Tangkuampien, 2009). Whilst this was useful for school pupils who were preparing for exams, the solution was too limited when it came to extending access to a richer and wider variety of learning data and information that has become common to a modern learner (Wanyonyi, 2008). A richer data variety refers to that extend beyond textual, but include picture, video and audio formats of data to enhance student engagement over interactive platforms (Ivala and Gachago, 2012; Mlitwa, 2011). In the context of a knowledge society of the information age (Castells, 2000), a modern learner can be understood in a comparative context between a learner in the twentieth century (non-modern) and a twenty-first-century student who is confronted with, and, from multiple information formats to draw learning experiences. Such learning sources go beyond just paper in library shelves or a teacher in fixed classrooms, to include electronic media such as e-Journals, e-books, learning management systems and interactive online discussion forums among others (Mlitwa, 2011). In this respect, common examples of interactive tools in higher education spaces include social media platforms such as Facebook and blogs (Ivala and Gachaga, 2012).

2.2 Podcasting and image-based m-Learning models

Unlike the text-based models that are only limited to the textual content, a podcast is the most common technology to distribute audio data over Internet platforms (Lightbody *et al.*, 2007). Because a podcast uses XML-based "Really Simple Syndication" (RSS) technology (RSS feed reader) to facilitate the downloading, listening and viewing of XML-formatted video and audio content over mobile devices (Wanyonyi, 2008), podcasting requires XML-compliant mobile devices such as the post-third-generation (3G) appliances (Mlitwa, 2011). In Figure 2, it is illustrated how podcasting technology is used to facilitate m-Learning.

The advantage of podcasts over text-based content models is that the audio handling capability can be exploited to deliver recorded content from lectures, to mobile devices for access by a mobile learner. Interested clients subscribe to a topic to receive messages whenever any content associated with it is published (Wanyonyi, 2008). As shown in Figure 2, the process involves audio-recording of educational content by, and to, one or more recipients. The publish–subscribe scenario is used to facilitate content distribution to subscriber appliances. In this respect, RSS feed software in subscriber appliances facilitates an automatic download of every file that is published – with no queuing to send or receive a message – thereby making it available to subscribers at once (Lightbody *et al.*, 2007).

As an m-Learning platform, therefore, podcasts surpass SMS-based models in a number of ways. A podcast enables live recording of lectures, making it easy for learners to access missed lecturers. The main challenge according to Mlitwa (2011), however, is that educators were displeased with a positive correlation between the use of such technology and the declining attendance to lectures. When used responsibly, however, the model can contribute to learning, in a number of ways. It can be a useful tutorial tool for the visually impaired. m-Learning can also be useful for archiving what would in turn, support life-long learning (Wanyonyi, 2008). As discovered in mobile Learning (m-Learning) experiment at the University of Ulster, however, podcasting fails on the basis of interactivity (Lightbody *et al.*, 2007). For example, it remains an asynchronous tool which neither enables the real-time interaction with instructors and other learning participants nor richer engagement with other forms of learning material outside stored information bases (Mlitwa, 2011). Even promising attempts to improve interactivity of mobile learning platforms involving multimedia applications such as the Mobile Learning Engine, have seemed unsuccessful. In this attempt to embed learning objects with formatted text, images, audio tools, video playback facilities, graphical question management system and, to mark hotspots within a picture (Holzinger *et al.*, 2005), a lack of real-time interactivity remained the limitations (Mlitwa, 2011).

2.3 Rich Internet applications models

Nevertheless, belief in the advantages that m-Learning offer to the quality of learning is paralleled by new efforts to improve the interactive component of m-Learning applications in the period of 2004-2006. For example, prototypes of Rich Internet Applications (RIAs) for mobile learning appliances began emerging in 2005. The Self-Assessment Tool (SATO) as presented by the Medical University of Austria at the Central European Multimedia and Virtual Reality Conference in 2005, is a case in point (Holzinger and Ebner, 2005). Obviously, the advantage of RIA innovations is in their added interactive and responsive components "that support complex graphical user interfaces (GUIs) using advanced interaction patterns" (Wanyonyi, 2008, p. 14). These include social network-compliant multimedia, including those with the drag and drop features. The SATO tool, in particular, was initially designed as a self-directed learning

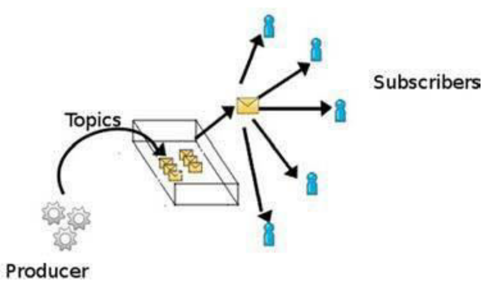

Figure 2.
Podcasting, push-subscribe content distribution model

desktop application that was later enhanced through Flash Lite technology (a mobile version of Adobe's Flash software), to support interactive learning objects in the form of animations in mobile phones (Holzinger and Ebner, 2005). However, there are numerous factors that tend to challenge a productive use of interactive technology solutions in educational settings, and this initiative was also not immune. For example, even if learners do want to use ICT for collaborative purposes, unhelpful attitudes by educators or inadequate infrastructure (Ng'ambi and Rambe, 2008) can be inhibiting. In this instance, a lack of Flash Lite-compliant mobile phones in Austria in 2005, led to the innovation and the prototype being reduced into a mere Personal Digital Application. Although SATO's popularity and impact was not enhanced by this shortfall, SATO offered an insightful basis upon which more interactive m-Learning innovations, the mobile-based learning management systems (m-LMSs), could be built.

2.4 Mobile-based learning management systems

In the context of m-Learning, conventional learning management systems (LMSs) have also been limited on the mobility aspects and on the real-time interactive aspects. Interactive real-time social media features including chat-room applications have since been integrated into SAKAI such as Vula (https://vula.uct.ac.za/portal), Blackboard (www.blackboard.com/) and Moodle (http://moodle.org), among other LMSs (Mlitwa, 2011). Moodle, in particular, is designed with software to facilitate learning activities online and offline, with efforts to extend access to these facilities to mobile users anywhere and at anytime (Wanyonyi, 2008). In this respect, a mini LMS tool, Poodle, was developed to read quizzes from the Moodle server into mobile applications. Through Poodle, students give answers to quizzes and the tool would send them back to the instructor evaluation. Lecturers would also use the tool in classroom to facilitate polling on specific class questions (Houser and Thornton, 2005). Whilst Poodle offered a notable breakthrough towards the extension of LMSs to mobile platforms, interactive features were still too limited to make a meaningful impact to m-Learning. This prompted researchers at the Arabia Open University to develop a Moodle Wireless Application Protocol (WAP)-based m-Learning application, for access in wireless devices (Alsadi and Abushawar, 2009).

As a data-oriented protocol, WAP allows users to securely access data-oriented services via desktop-like applications running on mobile phones, from anywhere where the general packet radio service (GPRS) is available (Wanyonyi, 2008). It operates on the concept of a layered stack, composed of an application layer (i.e. mini browser), session layer, transaction layer, security layer and a transport layer (Erlandson and Ocklind, 1998). Based on a WAP client and a WAP server configuration, a WAP client sends a request that has the server Uniform Resource Locator (URL) using the WAP protocol to a WAP gateway. From the WAP gateway, this requests is forwarded as a conventional HTTP request to the server – which recognizes the URL and responds back via the WAP gateway (Erlandson and Ocklind, 1998). The WAP gateway verifies and encodes replies into appropriate binary formats, and forwards responses to the mobile device using WAP. In the context of m-Learning, therefore, WAP enables access to multiple-choice questions in LMSs through mobile devices. Through a WAP browser, WAP technology can be used to browse the Internet, deliver alerts and reminders and, to enable display tip of the day messages. Possible uses of WAP to support interactive m-Learning are outlined in Figure 3.

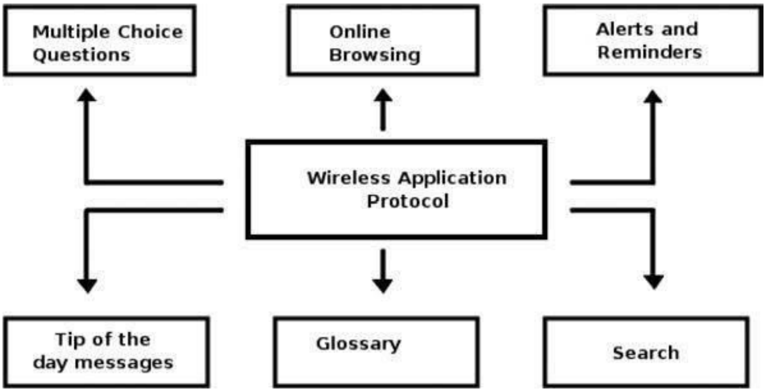

Figure 3.
Various uses of WAP in m-Learning

As shown in Figure 3, the application could be configured to receive a notification from the server (whenever there is a specific activity, be it an email inbox or new learning content in an e-Learning platform), to mobile devices as a short ring tone or vibration. Both the blackboard LMS at the Cape Peninsula University of Technology (CPUT) and the Vula version of the SAKAI project at the University of Cape Town (UCT) are programmed to send automated e-mails to the user whenever there is movement in an LMS (Mlitwa, 2011). An extension of this into mobile phones, by exploiting the proposed *m-Chisimba* innovation, would be a welcome convenience for users in these institutions.

Whilst advancements in interactive technology innovations are commendable, barriers to their effective implementations across all teaching and learning environments remain evident in emerging research. Issues of literacy limitations and attitude by educators (Ng=ambi and Rambe, 2008), as well the association of tensions between lecturers' teaching strategies (pedagogy) and students' conceptions of learning, with the use interactive platforms such as Facebook (Rambe, 2010), are growing in prominence. In fact, even in cases of an ideal technology prototype, the socio-economic issues of poverty, cultural barriers and poor power supply, could still limit access and use of teaching and learning technology in under-developed environments (Rambe and Mawere, 2011). Inability to afford and use networked m-Learning tools and systems, including airtime to connect over a mobile phone, can inhibit access. As Rambe (2012) reports however, although interactive social media such as Facebook enhances "democratic engagement" and "constructivist learning", it is hardly a panacea to all learning concerns. For example, students using Facebook were found to be unable to engage at epistemological and conceptual levels, with claims that the technology fails to ensure deep learning as learners also seemed "unable to develop quality academic discussions" (Rame, 2012). The problem of underdeveloped ICT architecture (Rambe and Mawere, 2011), as well as design and usability limitations in existing innovations (Mlitwa, 2011), further inhibit relevance and usage of interactive technologies. With this critique, it is demonstrated that whilst the interactive component of mobile applications is important for a mobile learner, it is not without limitations and that there are still challenges to its accessibility by the majority in underdeveloped areas.

As shown in Figure 3, however, WAP has increased the potential to link an LMS to mobile devices. In terms of limitations and usability limitations, however, it is by no means an adequate solution to all m-Learning related challenges. For example, a WAP browser is not only useful but also limited in terms of capabilities, especially tiny screen and memory constraints in mobile phones. In the AOU's Poodle project for example, complications with short battery life spans, as well as slow transmissions and failure to send information to some versions of mobile devices – emerged as major shortfalls. The notion of tiny screens in this respect is used to compare the screen of a mobile phone to that of a personal computer (PC). By comparison then, a mobile phone screen is tiny. The challenge is evident when the content transmitted from a PC desktop to an application with such a screen (such as a mobile phone) is not customized to adjust to a tiny mobile phone screen. By slow transmissions, it means that when data are incompatible with a destination device, transmission may fail or it can unfold at a much reduced speed, hence, the notion of slow transmission. That is, it can take longer for the user to download the content from their end. Finally, electronic communication devices have a memory capacity associated with their size and data handling functions. A PC, for example, would have a larger hard drive and a function processor by comparison to that of a mobile device such as a mobile phone. Transmission and access of PC-customized data in a mobile phone, therefore, require considerations in terms of the storage capacity to handle such data in a mobile application. Nevertheless, Poodle developers acknowledge the potential that WAP possesses and are optimistic that further research in mobile-based LMS would eventually yield improved m-learning innovations.

It is on this basis that an innovative integration of mobile-based learning management systems with the new mobile applications for an interactive m-Learning solution, was explored in the project presented in this article.

3. Research methodology

The methodology of the project was twofold. At the first instance, it is literature based, at the second instance, experimental. The methodology was built on the literature background that reveals the maturity of various efforts towards interactive m-Learning solutions. While intrigued by the potential of mobile-based learning management system (m-LMS) efforts, the apparent shortfalls remained a challenge. There were difficulties in transmitting presentations from LMSs to mobile phones, mostly owing to limited memory, content compatibility with small screens of mobile phones. These technical limitations had a negative impact on connection times and related airtime costs, which, in turn, restricted access times to m-Learning content over mobile phones.

The study undertook to explore the ways to provide presentations and interactive chats for effective m-Learning over mobile phones. The objective was to scale down presentations and reduce file size for effective display on mobile devices, obviously, without compromising the quality of slides. Finally, and most significantly, we also wanted to explore measures of minimizing airtime costs when transmitting presentation slides and when conducting interactive chats on mobile phones. An experimental methodology, thus, was more ideal for this type of research. The starting point was to develop software as a base-concept for formatting and transmitting presentations to a mobile phone. Emphasis on the software was to enable instructors to use their desktop to exchange rich educational content in presentations (Microsoft Power Point or Oracle Open Office Impress) with learners over their mobile phones. The software also enables near-real-time instant text-based interaction (chat) between the instructor and learners during learning sessions.

3.1 The experimental process

A laboratory consisting of a SunFire server (www.oracle.com) installed with OpenFire 3.6.4 as an XMPP server, operating on Internet connected Ubuntu 8.10 (www.ubuntu.com) at the University of Witwatersrand in Johannesburg, South Africa, was a physical environment for the experiment. The MySQL 5.1 (www.mysql.com) was also installed into the server to facilitate database functionality. In addition, 21 random presentations that ranged between 17.7 KB and 28.4 MB were downloaded freely from the university library repository (http://presentations.wits.ac.za) and used in the experiment. A J2ME-enabled Nokia N82 mobile phone operating on the Vodacom (www.vodacom.co.za) South Africa network was used in the experiment. To emulate real-life context of a South African instructor and learner, presentations were uploaded into the desktop client of the experimental software. At the initial attempt, the XMPP – which is a protocol that provides real-time messaging – was used to transport slides from a desktop client to a server. During experimentation, however, an appropriate mobile-based XMPP library with a footprint small enough to address size and capacity requirements for a mobile client was unavailable at that time. So, we resorted to J2ME's HTTP libraries to keep our mobile application footprint minimal, thereby evading slowing size effect that had been experienced in previous innovative efforts. As illustrated in Figure 4, communication between the server and the mobile client in this project was HTTP-based.

The core of the experiment was to convert presentation slides uploaded in the instructor desktop by a server, into a series of images that could be easily transported into the mobile client. So, a Java library tool, the JODConverter (http://code.google.com/p/jodconverter/) was used to covert office presentations into HTML. Converted images were then viewed and tested for quality assurance, using a viewing application developed for this purpose, to correct irregularities that might have occurred during the conversion. Image histograms, which described the frequency of the intensity values occurring in an image – were used to effect quality testing of slides. A statistical analysis method using an imaging processing software, the ImageJ 1.4.2 Java tool, was used to

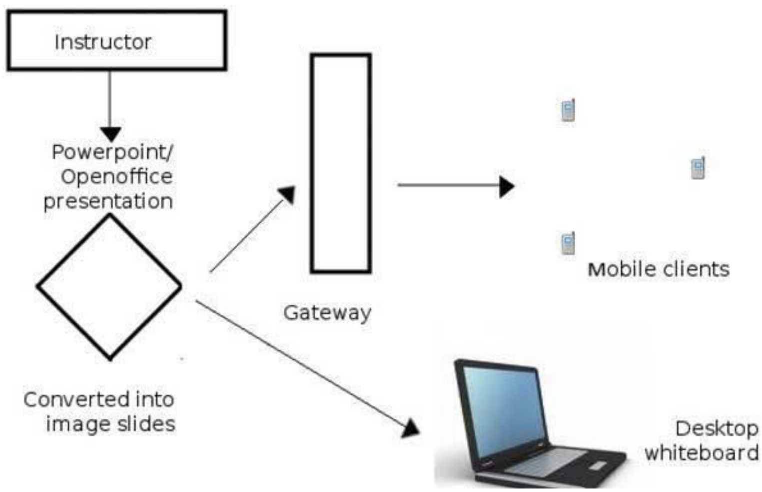

Figure 4.
Conversion of slides from desktop to a mobile client

analyse image histograms. To computer a histogram for an 8-bit image needed a set of 256 counters (as per the 256 colour image base), to quantify the quality of the image by taking a snapshot of a slide. These were used to plot the number of pixels at each point of the slide, to determine the overall intensity in the image. In fact, image histogram mean values ranged between 0 and 255, which was divided into four sections of A, B, C and D, with A being closest to 0 (which is darker and poorer), D being closest to 255 (brighter and more visible in a tiny mobile phone screen). The idea was to quantify the quality of generated images, in terms of brightness and contrast when viewed on mobile phone screens. Then, presentations with high-quality images were transmitted from the desktop into the mobile phone for viewing.

On airtime cost minimization experiments, calculations were based on data rates published by the Vodacom network operator in May 2010. A Micrologtool, which is an open source logging library based on the Log4j API, was used to log measurements, which were then implemented on the server, to enable the recording of all passing data in kilobytes. When data were received in a mobile phone, packet sizes were recorded using the same logging framework. Findings on both experiments are elaborated under the presentation of the project outcome, the *m-Chisimba framework*, in Sections 3.1.1 and 3.1.2.

In summary, this experiment intended to address limitations in the quality of presentation slides when opened on mobile phones. The aim was also to improve the sustainability of a connection to e-Learning platforms and to improve the cost efficiency (in terms of lowering airtime costs) of m-Learning processes over mobile phones.

For the first task, the existing open source software development framework, the Chisimba[1] framework (www.chisimba.com) was used as a basis to transform the existing e-Learning environment into an interactive m-Learning enabling environment. We termed it the *m-Chisimba* framework, naming it after the parent Chisimba framework. Here, we wanted to format presentation slides so that they could display on the mobile phone screen with ease. In the fi phase, we worked on the automatic downscaling of presentations to reduce the fi size, as well as minimizing the time that automatic downscaling and fi transfers took. This also meant the preparation of the new platform, to simplify the transportation of such e-Learning fi into mobile devices. This phase included system confi testing and verifi of the outcomes. The structure a resultant *m-Chisimba* framework is discussed in greater detail in Section 3.1.1.

3.1.1 Structuring the m-Chisimba *framework.* The outcome of the experiment was a successful development of an interactive m-Learning system. As illustrated in Figure 5,

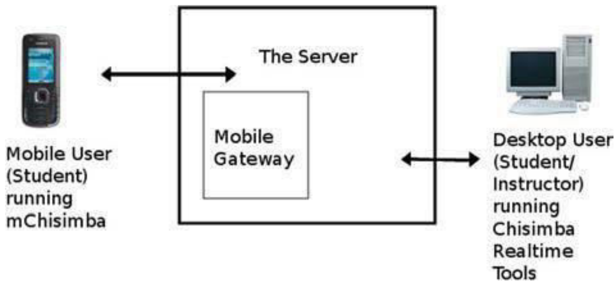

Figure 5. Overview of the *m-Chisimba* framework

the framework consists of three components, the desktop client, the server (which consists of the database) and the mobile client. Within the mobile client is a graphical interface that consists of a presentation viewer, for displaying the slides and the chat-room used for text chatting. The presentations viewer included a set of navigational controls for browsing through slides.

In this framework, the instructor initiatives interaction from the desktop client by preparing a virtual room into which, participants are invited to join via desktop or mobile clients. Because the server has a gateway for transcoding packets between desktop client and a mobile phone, it is responsible for interfacing the desktop and mobile clients, for authenticated and for validating users who attempt to enter the m-Learning network.

With a focus of interactivity and system compatibility with structural limitations of a mobile phone, the mobile client (*m-Chisimba*) was built on the J2ME platform consisting of the viewer and transport layers. Whilst the transport layer facilitated the transportation of packets between the server and a mobile client via the HTTP, the viewer layer interpreted incoming packets and facilitated the display of the appropriate format of output in the mobile client. As outlined in Figure 6, the layer also responded to human input, and supported the generation of appropriate XML packets for the transport layer.

Within the viewer layer in Figure 6, the GUI layer was built using LWUIT, which allowed developers to create a GUI that was almost similar to desktop applications, thereby enhancing compatibility between desktop and mobile applications. In terms of the process, the user interacted with the virtual room consisting of a user list, chat-room and the presentation viewer.

The communication model between the server and the mobile client is discussed in Section 3.1.2.

3.1.2 The m-Chisimba *communication model.* Based on our testing, we found HTTP to be the easiest and the most appropriate connection type for our purpose. The main reasoning was that, unlike push mechanisms that work on inbound connections using socket-based protocols (such as TCP and UDP) that cannot work outside the local area network in data-oriented mobile phone connections, HTTP is supported by cellular networks without need for any special configuration. However, because the HTTP is a client type of network connection, it does not support inbound connections. Thus, HTTP servers do not initiate connection to the clients. Rather, the client must initialize a connection. Therefore, the fixed-interval polling was adapted as a communication model between mobile client and server. That is, we adopted the HTTP-based task timer model so as to be able to synchronize with the server after every n seconds/microseconds. This was achieved by sending HTTP requests to the server at preset intervals. To minimize traffic between the server and the mobile client, the server responded only when the

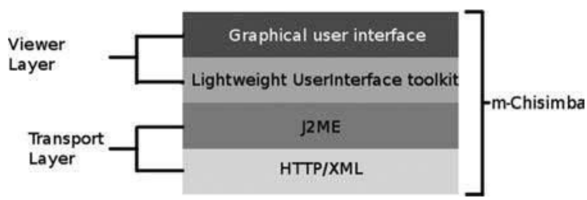

Figure 6.
Structural layout of *m-Chisimba*

instructor's presentation had changed. If there were no changes since the last poll, a client request was simply ignored.

The server is made of two main components implemented as plug-ins inside *Openfire* server. This is a Jabber-compliant, real-time collaboration server, which uses the XMPP (an open Extensible Markup Language) as the communication protocol for near-real-time messaging, presence and request-response services (Figure 7).

The first plug-in, server-plug-in, was used for interpreting client packets, except for HTTP requests from mobile client. The second component, the gateway, transforms packets from server, meant for mobile client, into HTTP packets that were sent to *m-Chisimba* mobile client. It also transformed incoming HTTP packets from the mobile client into valid XMPP packets before being forwarded to the *m-Chisimba* server. As outlined in Figure 8, the server plugin also had a packet processor for handling *Information Query* (IQ) packets from the gateway plugin or from the desktop client. It also had the packet interceptor (Figure 9), which forwarded packets to a database service for querying information from the database, for forwarding packets to the broadcast service and, ultimately, for returning packets along the same route.

Figure 8 shows a typical IQ packet. In effect, the gateway and server plugins communicate by exchanging these packets over XMPP. The desktop client also sends these packets directly to the server plugin. With the exception of the message packets that are handled by the *Openfi* server directly, all other incoming and outgoing packets pass through a packet processor. Packets meant for broadcast to all room participants, when intercepted by the packet processor, are first stored in a database then broadcast to all other participants connected to the session, using the broadcast service. Other packets meant for querying the server for information are forwarded to the database service for information retrieval/storage, then are returned

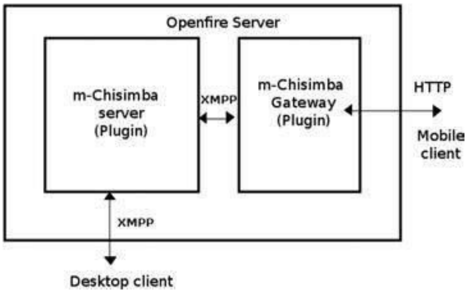

Figure 7.
m-Chisimba server components

```
<iq id="q2s0S-9" type="get">
  <query xmlns="iq:avoir-realtime:realtime">
    <mode>pointer</mode>
    <content>
      <pointer>1</pointer>
      <x>208</x>
      <y>61</y>
    </content>
  </query>
</iq>
```

Figure 8.
An information query (IQ) packet

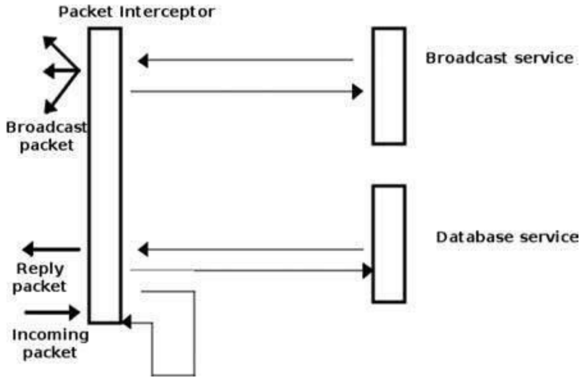

Figure 9.
The server plug-in packet interceptor

via same route or forwarded to the broadcast service. Such packets, for example, are sent when a client queries for available scheduled sessions.

The gateway plugin was compatible with the mobile client and server protocols; hence, it was the "middleman" in the communication framework between the two. A request originating from a mobile client to the gateway typically consisted of an HTTP packet with two major elements: type and content. When the gateway received this packet, it determined the type. Based on the result, the content was extracted from the packet and repacked as a "proper" XMPP IQ packet that was forwarded to the server plug-in for processing. This was accomplished with the Smack API (www.igniterealtime.org). The gateway plugin intercepted packets from the mobile client, transforming them into valid XMPP packets before forwarding them to the server plugin. Reply packets from the server plugins meant for the mobile client were first forwarded to the gateway plug-in, which transforms them into valid HTTP packets before being sent to the mobile client. The gateway plugin was implemented using three core classes, the *m-Chisimba plug-in* (as the main class and interface to the server plug-in), the *m-ChisimbaServlet* (HTTP protocol-based direct interface to mobile clients) and the *RpacketListener* (used by the *m-Chisimba plug-in* to interpret packets from the server plug-in). This is graphically illustrated in Figure 10.

To easily cope with complexities related to varying data formats (and quantities), MySQL was used as the database management system. Database access was implemented using two classes:

(1) the *AvoirRealtimePlug-in* (interface that receives and sends packets to and from other components); and
(2) the *RoomResourceManager* (containing Openfire API methods to access the database directly).

Only the server plugin (using the *Openfire* database access API) had a direct access to MySQL (where all packets are logged and user profiles stored).

In Figure 11, we illustrate how the server plugin component related to the database.

In fact, this process was aimed at more than just downscaling slides, to improve their quality and also their transferability to a mobile device but also to ensure that the format of the interaction process is also cost-effective for the mobile user in terms of airtime used on the mobile device. In the second phase, therefore, airtime costs for transporting

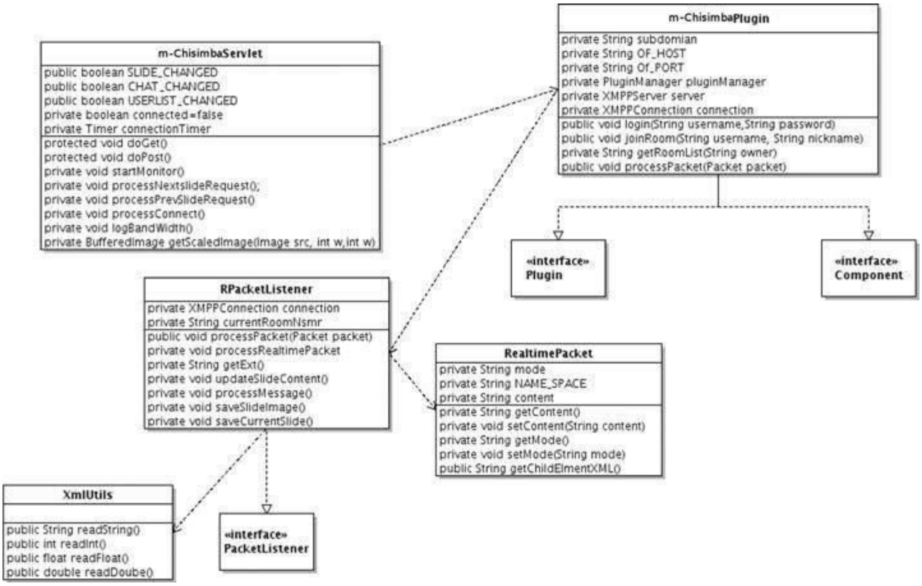

Figure 10.
The gateway class diagram

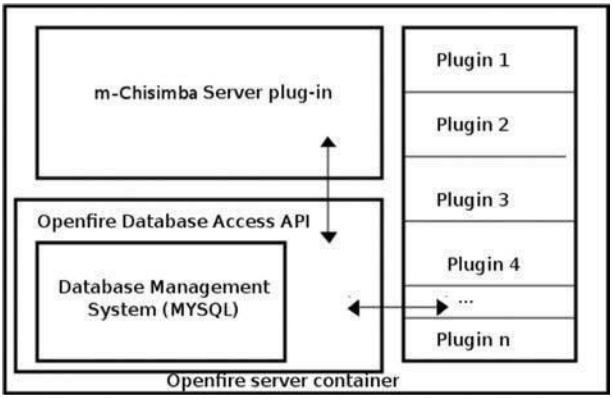

Figure 11.
m-Chisimba server plug-in and the database management system

and downloading of presentations in a mobile phone were measured. In this phase, data based on transportation of slides, polling of users and exchanges of chat message between users were collected and verified.

4. Findings: a resultant *m-Chisimba* framework

4.1 Scaling presentation slides for m-Learning compatibility

The quality of slides was measured in a colour intensity that ranged between 0 and 255. The ideal mean histogram value was at a color intensity of 201.188 (out of 255) that fell under category D (bright and more visible in a tiny mobile phone screen). In the following graphical illustrations, Figure 12 gives a bright desktop-based (control item) slide before conversion into a mobile client-compliant version.

Figure 12.
Base slide from desktop client

Figure 13 shows the automatic distribution process of this slide (as generated in the *ImageJplug-in* embedded in our software) from the desktop to a mobile phone screen, in terms of image quality (by quantity of pixels).

The level of clarity of these presentations was also rated in a score between 1 (low or poor) to 5 (high and brighter). A resultant mobile client image was visibly clear in a mobile phone screen in Figure 14. In a scale between 0 and 255 pixels, the histogram mean value of the conversion process in Figure 14 was 201.188 or category D, which is closer to 255, with clarity value of 4.

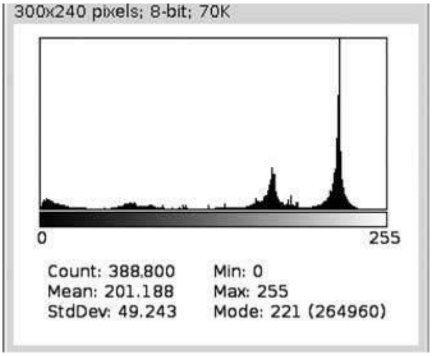

Figure 13.
Image value distribution-bright slide

Figure 14.
Resultant slide on a mobile client

Presentations with a histogram mean value less than 128 pixels were overall, poor as compared to those above this average size. A reverse test of a slide with a mean quantity of 40.907 pixels, for example, resulted into an image of a poor quality in a small mobile phone screen. Obviously, there was more to slide clarity and fluidity than just the pixel size. There were a number of slides with a histogram value of 237.32 pixels (slide number 7 of n 21) and 186.92 pixels (slide number 20 of n 21) turned out to fall into the clarity score of 1 on the mobile client. This was extremely poor, despite the histogram pixel size being higher than the average minimum of 128 pixels. On closer examination, we observed that font size also plays a significant role in the final output on mobile client. Whilst clear slides also had font greater than size 32, for example, poor slides also had a font size smaller than 32.

The time dimension in terms of the speed to slide conversion is a key factor in the m-Learning quality as well. To improve this, we ensured that the scaling process is based on the screen dimension of a mobile phone – supplied in real-time when the mobile phone client connects to the server. This way, content compatibility and ultimately, the scaling time, is significantly improved. The average scale time per presentation is outlined in Figure 15.

As shown in Figure 15, the average time it took to scale a slide image to fit mobile specification was 38.05 milliseconds. This was much lower than 100 milliseconds, and therefore, ideal for real-time content formatting for mobile phone applications, in particular, as it did not seem to have negative implications in terms of mobile phone airtime costs.

4.2 The airtime cost factor for m-Learning

Our analysis of airtime costs is based on the amount, type, quality and speed of data that are transferred from the desktop client to the mobile client.

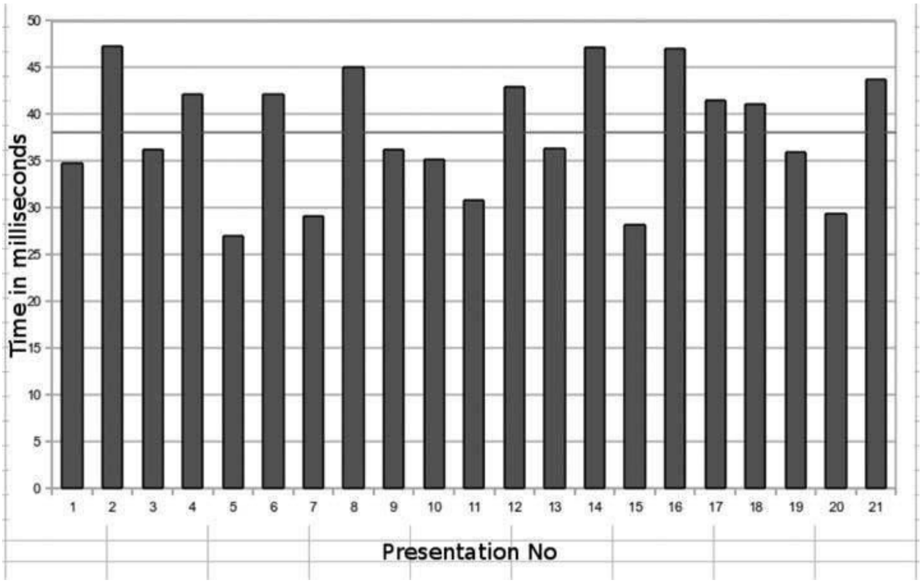

Figure 15. Average scale time per presentation

To obtain the data transferred, we conducted experiments in three phases. Phase one involved recording of data transported when using presentations. To establish whether the polling process introduced any signifi increases on the amount of data transferred, Phase two involved the recording of data transmitted during the polling of users in the virtual room was recorded. Finally, Phase 3 involved a simulation of seven users with randomly generated text, recording the amount of data transferred during the chat session to emulate chatting in a virtual room. Air time cost calculations were based on Vodacom's data rates using Java's specifi tailored logging mechanisms for mobile phones, the Microlog library (http://microlog.microsuite.org/), by embedding it into the class path of the mobile client application. Logs, including packets size, were also implemented on the server, using the Log4J mobile logging framework.

As shown in Figure 16, the average amount of data per presentation is 9,872.66 bytes (or 0.08 megabytes).

Whilst the size of data is fairly small, it is more informative to understand whether the polling of users would make a significant difference in data size.

In Figure 17, the average amount of data that is transported to the mobile phone is as little as 73.9375 bytes. Because this refers only to presentation slides, the size of data in a real-time chat in an *m-Chisimba* environment would add more insight. This is shown in Figure 18.

The average amount of bytes transmitted when seven simulated users are chatting is 3,686.63 bytes (which is 0.03 megabytes). Now the cost of data can be calculated using the measure that a mobile phone operator uses. Vodacom operates what is called "MyMeg 300 in-bundle-data option" which is R0.46 per megabyte. This translates to 0.8 0.46 R0.0368 cost per slide.

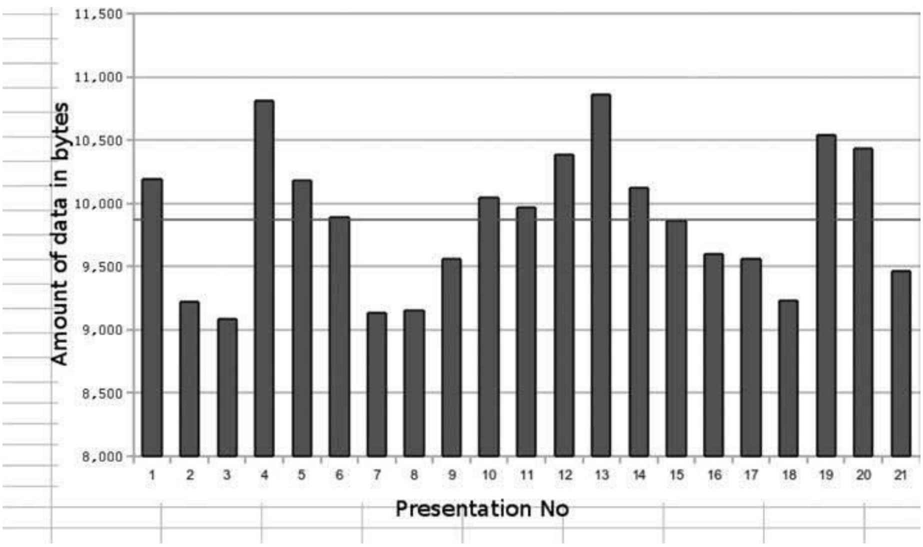

Figure 16.
Average bytes per presentation slide in mobile a client

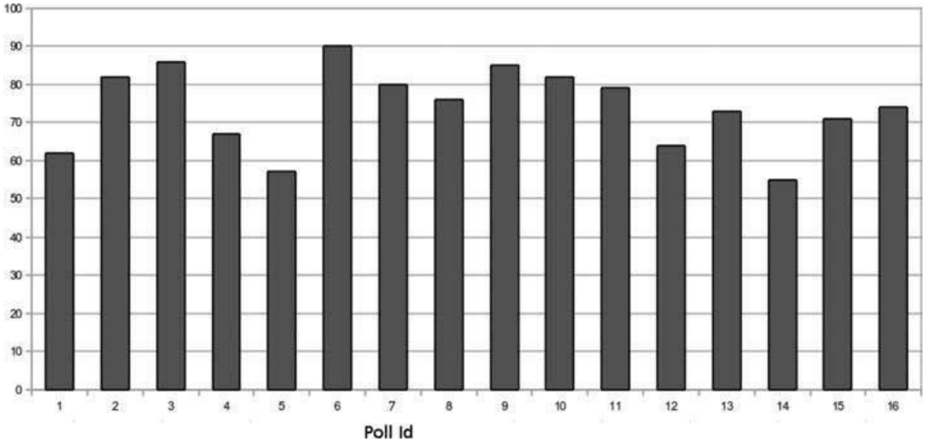

Figure 17. Data amount (bytes) to mobile phone when polling users

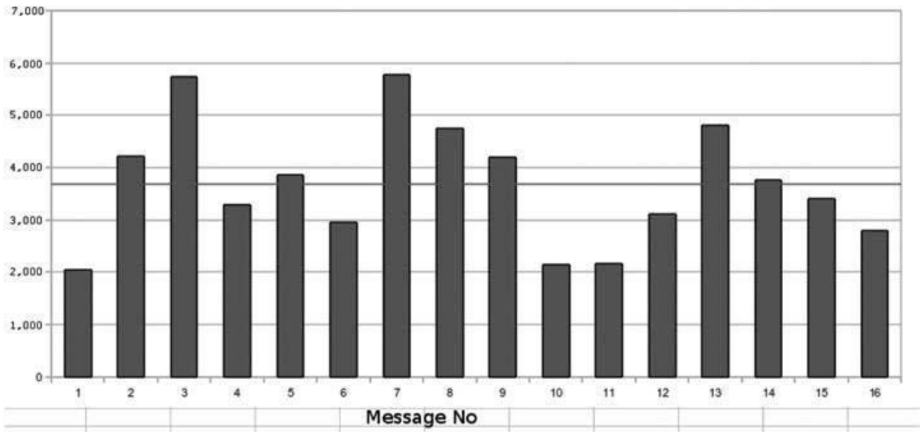

Figure 18. Data size when seven (7) simulated users are chatting

On this basis, we can confidently conclude that the *m-Chisimba* platform is both cost-effective and expedient in terms of quality, making it the most appropriate platform for interactive LMS based m-Learning.

5. Conclusion

In the *m-Chisimba* initiative, we sought to develop an interactive m-Learning platform that would address limitations incompatibility between web-based e-Learning processes and mobile phone platforms. One of the goals was to provide presentations that can be viewed on both desktop and mobile phone platforms – with equal ease. On this point, we wanted to answer the question: "How do we provide presentations and interactive chat to support m-Learning?" This question was addressed by scaling presentation slides down, formatting them according to a mobile phone screen specifi and transporting them to the device for display. This was achieved by converting slides in a presentation into a series of images, which were then scaled down to a format and size appropriate for mobile display.

Then, experiments were carried out with a smart mobile phone running on a 3G cellular network. In addition to automated image analysis techniques to observe and record data, transaction-logging techniques were also used. The XMPP-based instant messaging was also introduced to provide near-real-time text chatting functionality. Further, a proof-of-concept application was also developed to enable the formatting of presentations created on desktop computers so that they can be easily transported to, and, displayed on a mobile phone, by converting the slides in a presentation into a series of images. The next step, then, was to carry out a series of experiments, to verify the adaptability success of e-Learning processes in mobile platforms. Emphasis then shifted to the quality of the slides when transported and opened on mobile phones. After the downscaling and formatting of slides, using image histograms measure the quality, the quality of slides and chat content on mobile phone displays was found to be satisfactory.

One of the main concerns was about airtime cost implications, with extra effort placed on avoiding an m-Learning solution that would escalate airtime costs for the user. On this point, the question was "How do we minimize airtime costs when sending presentations and conducting interactive chat on a mobile phone?"

This question was answered by implementing an algorithm that allows the mobile client to pull the data from the server only when there has been an update from the instructor's screen. The polling process takes place in three separate threads:

(1) the first thread checks the status of slides from the instructor and updates the mobile client accordingly;

(2) the second thread updates the user list; and

(3) the third and most important thread with respect to the above research question was the chat polling.

This thread refreshes the chat transcript as long as there are new messages. However, although a fixed-interval polling model was used, the actual polling process does not introduce any significant airtime cost element.

In view of the experiment success, it can be concluded with utmost confidence that the *m-Chisimba* initiative has drawn lessons from failures in preceding m-Learning projects, successfully offering an operationally sound and cost-effective alternative. Obviously, mobile applications are convenient enablers of extended access to information. Nevertheless, mobile applications (including m-Learning solutions) cannot be seen as panaceas to all learning ills as literacy and affordability remains a challenge in many under-developed social environments.

As an m-Learning innovation, the *m-Chisimba* framework offers stakeholders within the community of practices, a renewed hope towards an interactive and a richer m-Learning experience. On this basis, the rise in m-Learning adoptions across higher education spaces may no longer come as a surprise.

Note

1. As a development framework, Chisimba is drawn from the Chichewa (one of Malawian languages) word that refers to a framework used to build a traditional house (www.chisimba.com).