# An Evaluation of a Mobile Phone Platform as a Convergent Technology for Text Based Communication

Curtis Sahd, Hannah Thinyane and Mamello Thinyane
Department of Computer Science, Rhodes University, P. O. Box 94, Grahamstown
Tel: +27 46 603 8640, Fax: +27 46 636 1915
Email: curtissahd@gmail.com; h.thinyane@ru.ac.za; m.thinyane@ru.ac.za

**Abstract— This paper investigates inexpensive means of communication through the use of instant messaging, text messaging over Hyper Text Transfer Protocol, and mobile email. It describes ThEm, a suite of applications consisting of an instant messenger, text messenger, and email application. All applications have been designed in such a way so as to make best use of the intermittent data connections on cell phone networks in South Africa. Since the convenience of mobility is fast becoming a reality, it is important to pay considerable attention to the design and ease of use of software designed for such devices [1]. This paper first presents ThEm, and then describes an evaluation undertaken to compare one particular component of ThEm against MXit, a leading off the shelf instant messaging client. This comparison is performed along the following three facets: core functionality timing, user interface evaluation and timing using the keystroke-level model for advanced mobile phone interaction, and user testing and evaluation.**

**Index Terms— mobile communication, telephone sets, text communication, user interfaces.**

## I. INTRODUCTION

WHEN cell phones initially came on to the market, speech quality was a major deciding factor for consumers when choosing a cell phone. Since then there have been rapid advancements in cell phone technology, the differences in speech quality on various handsets is now less noticeable [1]. As well as this growth in usability of cell phones, the services offered on mobile phones have also shifted to include data services [2]. Many cell phones also offer instant messaging services, and in countries like Japan, China and India, text messages can be sent via e-mail [3]. These convergent technologies blend multiple streams of information into a single presentation on a single device, and are central to the future growth of the information technology (IT) industry worldwide [4]. To date, convergent technologies have generated an increasing demand by consumers for the broadband spectrum and applications for its use [4]. According to a survey conducted by J.D Power and Associates the average cellular customer uses their cell phone for 6 hours every month. Customers receive an average of 7 text messages per month and 60 percent of cellular calls are made outdoors, of which 62 percent of these outdoor calls take place in cars [5]. With the aforementioned points, it can clearly be seen that cell phones are being used as more than mere devices for placing calls. People spend a lot of time in vehicles and this makes the mobility of the cell phone and hence the vast array of communication tools that come with the cell phone vital for a fast paced lifestyle. Naturally, people want to minimize expenditure, and this is where text messaging, instant messaging and e-mail truly show their benefits.

This paper describes an investigation into the use of a mobile phone platform as a convergent technology for text based communication. It presents the design of a prototype that was developed and user tested to determine the viability of the platform as compared to already existing separate text-based communication applications.

## II. DESIGN

This section describes the design of ThEm. It first introduces the system as a whole, and then describes the sub-components of the systems: JabberEm, SendEm, and MailEm. ThEm was implemented as a proof of concept to show that inexpensive and equivalent communication can take place on mobile devices. To investigate mobile devices as a platform for convergent technologies, it was decided to implement an instant messaging application, text messaging application and e-mail application. Even though the functionality of applications designed for mobile devices is somewhat limited when compared to that of the personal computer, the basic tools which are necessary for various implementations are provided in J2ME.

ThEm consists of a number of applications, namely: JabberEm, SendEm and MailEm. Each of these applications was designed as a proof of concept of mobile communication for mobile devices. Instant messaging, text messaging, and e-mail are fast becoming part of everyday communication. JabberEm is an implementation of the Jabber protocol on cell phones and enables users to communicate with Google Talk contacts, MXit contacts, and 2Go contacts. JabberEm is able to send and receive messages between any client that makes use of the Jabber protocol, otherwise referred to as XMPP 1.0 Standard [6]. SendEm is a proof of concept application which enables users to send text messages by making use of the free text messaging service provided by the Vodacom4me website [7]. Users are limited to twenty text messages per day. MailEm is an e-mail client implementation which allows users to send and receive e-mails using their Google email address. MailEm too is a proof of concept application implementation, and thus HTML and attachment support for e-mail has not been included

JabberEm, SendEm and MailEm all make use of one of J2ME's record stores for storage of sensitive information [8]. Record Stores allow information such as usernames and passwords and various other non-persistent information to be stored persistently and securely. Record stores reduce the need for the repetitive entry of information in J2ME applications and also allow for consistent management of application settings and defaults. The implementation of all the applications developed in this project was done in J2ME, with the help of various toolkits. Due to the mobile device design considerations described by Zheng and Ni [9], B. Far *et al.* [10] and Vogiazou [11], all applications were developed with maximum efficiency, compact size, and a simple user interface which caters for the regular and advanced users needs. The user interface was constructed using the standard user interface components of J2ME.

Many applications make use of user interface toolkits which improve the look and feel of the mobile application, but at the same time, the efficiency of the application is diminished considerably [12]. For this reason it was decided to use the standard user interface components of J2ME in such a way so as to provide maximum aesthetic appeal to users without compromising the efficiency of the application. Since ThEm makes use of the native user interface components in J2ME, the aesthetic quality of the applications is diminished. There is however a J2ME component which forms part of the Screen class, namely the List, which allows for the appending of icons and the general improvement of an otherwise dull user interface. Because of the vast amount of varying cell phone models and a slight variation in the interpretation of J2ME on each model, the user interface sometimes differs considerably from the intended look and feel. These differences extend as far as the positioning of cards and decks (otherwise known as menus) in reverse order in some implementations [13].

J2ME was primarily selected as the application platform due to the simplistic publication and installation of JAR files in the J2ME environment. J2ME has integrated security mechanisms which protect the integrity of the data contained in Java Mobile applications [14]. One of these security measures implemented in J2ME is application level security, where access to libraries and resources is only granted where the user has explicitly authorized such actions [15]. Since all the applications make use of network connectivity, the need for threads is imperative to support user navigation at the same time as core functionality such as data exchange over a network.

All of the applications provide error prevention, thus minimizing application exceptions. Error prevention is implemented by ensuring that users have entered expected data types and that required fields have been filled in. ThEm has been designed for MIDP 2.0 and CLDC 1.1 compliant devices. All applications were tested on Nokia devices, ranging from lower end Nokia handsets to Symbian Series 40 and Symbian Series 60 devices. Emulators for Motorola, Sony Ericsson, Nokia, and Samsung devices were used, but the majority of the testing was performed on the physical Nokia Series 40 and Series 60 platforms.

In total, ThEm takes up 253 KB of memory. Clearly this applications can be installed on cell phones with the lowest processing and memory capabilities. One of the major benefits of messaging, both instant messaging and text messaging, is that messages are able to be sent asynchronously. Asynchronous messaging is non-blocking, meaning that users can send messages and continue to do other operations without having to wait for the sending of the messages. Synchronous messaging blocks the process until the sending and receiving of messages is complete. This means that users cannot continue working while messages are being sent and received. It is usually said that asynchronous messaging is superior to synchronous messaging, but in certain circumstances this is not the case, as will be explained next. For seamless user interaction with JabberEm it is necessary for an implementation of asynchronous messaging, since the user often wants to send a message to one contact and then continue chatting to another contact while the message is in transit to the first contact. This however differs somewhat in the implementation of SendEm, where the number of messages sent simultaneously as well as sequentially is governed by the Vodacom4me server. SendEm is thus more of a synchronous messaging implementation, apart from the fact that certain buttons and menus can be accessed [16].

## III. Evaluation

As ThEm is designed with a consistent interface across each of its three components, this paper presents an evaluation of only one of the sub-components. This section presents an evaluation of JabberEm, undertaken across four experiments that can be categorized as testing: core functionality timing, user interface evaluation and timing, and user testing and evaluation. Core functionality timing is tested across experiments 1 to 3. Experiment 1 deals with timing between chat clients and Google Talk across peak times. Experiment 2 performs a similar comparison but during off-peak hours. Experiment 3 measures the timing between JabberEm clients and timing between MXit clients during peak hours. User interface evaluation and timing is investigated in Experiment 4. In order to effectively evaluate the user interfaces of JabberEm and MXit, Experiment 4 was conducted in order to evaluate the user interfaces using the Keystroke-Level Model for Advanced Mobile Phone Interaction (KLM-AMPI). The third category of evaluation is user evaluation which takes into account user satisfaction while using JabberEm and MXit. A combination of these three categories will allow for an overall and consistent comparison between the two applications, and will allow us to draw rough conclusions so as to determine the application with the more efficient user interface and core functionality performance.

### *A. Design of Experiments*

In order to compare the efficiency of sending and receiving of messages, it is necessary to compile timing information taken during peak and off-peak hours so as to get an accurate representation of the results. The most important timing data that should be noted is that which is taken when exchanging messages between two JabberEm clients and two MXit clients. In the previous section it was mentioned that JabberEm is a proof of concept implementation which supports sending and receiving of messages to and from multiple Jabber server implementations. Since MXit also allows messages to be exchanged between multiple Jabber servers, this enables us to evaluate the time taken when exchanging messages between JabberEm and Google Talk, as well as between MXit and Google Talk.

In order to evaluate the efficiency of the user interfaces of JabberEm and MXit, we need a way of qualitatively and quantitatively representing the results. The way in which this is done is by making use of Keystroke-Level Model for Advanced Mobile Phone Interaction, which effectively calculates the time taken to complete common user operations, such as key presses and finger movements. The addition of these times gives a quantitative indication of the efficiency of the user interface. Without user evaluation and approval, the time to exchange messages and the efficiency of the user interface are rendered useless. User evaluation will consist of categories such as aesthetic appeal, ease of use, and functionality of the application.

### B. *Methodology*

The first experiment collects and analyses the times of messages sent and received between JabberEm and a Google Talk client, and MXit and a Google Talk client. Experiment one consists of a sample period of five messages sent from a Google Talk client to both JabberEm and MXit and vice versa. By combining these results a total of ten messages for JabberEm and MXit are analysed, which enables relatively consistent conclusions to be drawn regarding which client possesses faster communication with a Google Talk client (with repetition used to cater for network and server latencies). The sending and receiving of messages in Experiments 1 and 2 was performed between a Nokia N95 8GB and Google Talk running on Windows XP.

Experiments 2 and 3, however, provide more conclusive results in determining the faster client with regards to message exchange. Experiment 2 monitors and analyses the time it takes for messages to be delivered between two JabberEm clients and the time taken for message delivery between two MXit clients during off-peak hours. Experiment 3 is similar to Experiment 2, only that it is performed during peak hours. Since most people make use of MXit during the early evening, it was decided that timings of message exchange should be conducted both during peak and off-peak hours. Peak sampling was conducted between 7:15 pm and 8:45 pm, and off-peak sampling was conducted between 10:00 pm and 11:30 pm. Across experiments 2 and 3, the Nokia N95 8GB and the Sony Ericsson W850i. A sample period of 20 messages was used across Experiments 2 and 3. Again, times are averaged across the repeated messages to cater for lag due to network and server latencies.

Experiment 4 makes use of the Keystroke-Level Model for Advanced Mobile Phone Interaction, and evaluates the user interfaces of both JabberEm and MXit when performing certain key functions. All timing was performed using a stopwatch in Ubuntu Linux. The stopwatch began timing when the send button in JabberEm and MXit was pressed. When the messages arrived at the Google Talk and/or the other clients, the stopwatch was stopped and the time recorded. The margin of error for this experiment was assumed as MP from the KLM model (M – mental preparation, and P – pressing the button. Using the standard measures from KLM this is 1.2 + 0.1 = 1.3 seconds). The following experiments were conducted in order to obtain results for user evaluation of the applications. In order to get an accurate set of results for learnability, efficiency and memorability it was decided to make use of timing methods. Since learnability is defined as the ease of which applications can be used for the first time, a sample of six users were asked to perform each of the following core functions in JabberEm: Send message (The message “hi” was sent to a contact); Add contact (A Google Talk contact “test” was added); Remove contact (Contact at the top of contact list was removed); Show offline contacts; Change contact nickname (Contact at top of contact list changed to “hi”); Change current user nickname (Currently logged in user nickname changed to “hi”); and Message Vibrate.

| |
|---|
| 1. Please rate the ease of use of use of the application (Easy – Difficult) |
| 2. What do you think about the menu structures in the application, are they clearly labelled? (Clear–Unclear) |
| 3. Is this application aesthetically appealing (Attractive - Unattractive) |
| 4. What is the BEST feature of the application (Text) |
| 5. What is the WORST feature of the application (Text) |
| 6. What new content of features would you like to see in the application? (Text) |
| 7. Can you recover from mistakes easily? (Easy - Difficult) |
| 8. Your overall reaction to the application (Satisfied - Unsatisfied) |
| 9. Do you feel lost when using the application? (Yes / No) |
| 10. Is the application easy to navigate? (Yes / No) |
| 11. When you press a button in the application, do you expect it to lead to the correct answer? (Yes / No) |
| 12. Are there any comments on the application? (Text) |

**Table 1: User Evaluation Questions**

The times taken to perform each function for the first time were recorded, and then averaged for both MXit and JabberEm in order to determine the learnability of the applications. Since the timing between the learnability and efficiency test results is irrelevant, the efficiency test was performed immediately after the learnability test. Efficiency can be defined as the ease at which a user can perform certain functions once the design has been mastered. The way in which the efficiency of both applications was quantified was by asking each of the 6 test subjects to cycle through the above core functions twice over with a 2 minute break in-between each cycle. The times for each round were added and then averaged which then gave an indication of the efficiency of both applications. In order to determine the memorability of the applications, the 6 test subjects were asked to perform each of the core functions exactly 2 days after the tests for learnability and efficiency were conducted. All usability tests were conducted on a Nokia N95 8GB.

Questions 1 to 12 below, derived from a paper written by Yeng [16] were used in combination with the results from the above experiments to determine user satisfaction with the applications. The questions below cover some of the important areas of user evaluation, such as: investigating ease of use; aesthetic appeal; error recovery; and efficiency and learnability. The questions in Table 1 were subsequently used in the user evaluation after the timings for the core functions were performed.

### *C. Results and Analysis*

This section presents the results of each of the four experiments mentioned above, and compares JabberEm and MXit in terms of user interface efficiency and core functionality performance.

#### *1) Experiment 1: Peak Chat Clients to Google Talk*

The time taken for JabberEm and MXit clients to receive messages from Google Talk is presented in Table 2 below:

| | JabberEm | MXit |
|---|---|---|
| Msg1 | 3.03 | 3.62 |
| Msg2 | 3.36 | 2.85 |
| Msg3 | 2.41 | 3.16 |
| Msg4 | 4.64 | 3.86 |
| Msg5 | 3.13 | 5.17 |
| Total Time | 16.56 | 18.66 |

**Table 2: Time to receive message from Google Talk (seconds)**

The time taken for JabberEm and MXit clients to send messages to a Google Talk client is presented in Table 3 below:

| | JabberEm | MXit |
|---|---|---|
| Msg1 | 1.53 | 1.83 |
| Msg2 | 3.14 | 2.06 |
| Msg3 | 1.57 | 1.86 |
| Msg4 | 1.57 | 1.67 |
| Msg5 | 1.46 | 1.75 |
| Total Time | 9.26 | 9.17 |

**Table 3: Time to send message from Google Talk (seconds)**

It can clearly be seen from Tables 2 and 3 that the time taken to receive a message in JabberEm from a Google Talk client is 2.1 seconds faster than receiving the same message using MXit. MXit is however 0.09 seconds faster than JabberEm when sending a message to a Google Talk client. Due to the small number of data points, these timings can not be proved to be statistically significant. Also by taking into consideration the margin of error described in the previous section, the results do not show a statistically significant difference.

#### *2) Experiment 2:Off-Peak Chat Clients to Google Talk*

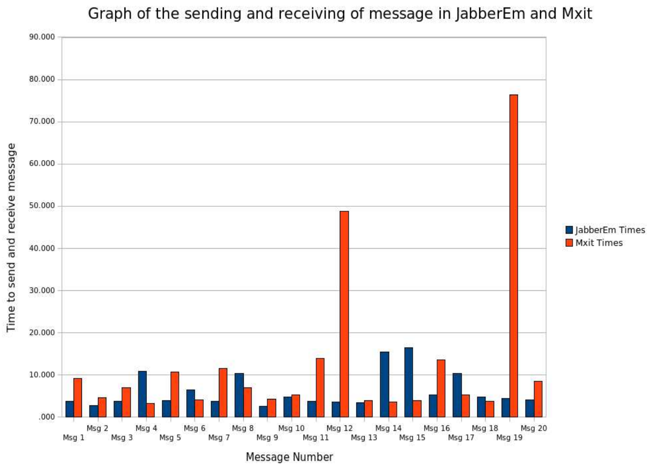


**Figure 1:Off-peak hours message transfer times**

A sample period of twenty messages was taken to determine the timing of the sending and receiving of messages between clients. All timing was done between 10:00 pm and 11:50 pm. Over a sample period of twenty messages, the sending and receiving of messages between JabberEm clients was 8.754 seconds faster than the sending and receiving of messages between MXit clients. From Figure 1 it can be seen that the sending and receiving of messages in JabberEm follows a more gradual increase and decrease in time than that of MXit. A possible reason for this is that the MXit's Jabber servers aren't equipped to handle volume as efficiently as those of Google.

#### *3) Experiment 3: Peak JabberEm to MXit*

The following timings were done between 7:15 pm and 8:35 pm, which is when most users are logged onto MXit during the week.

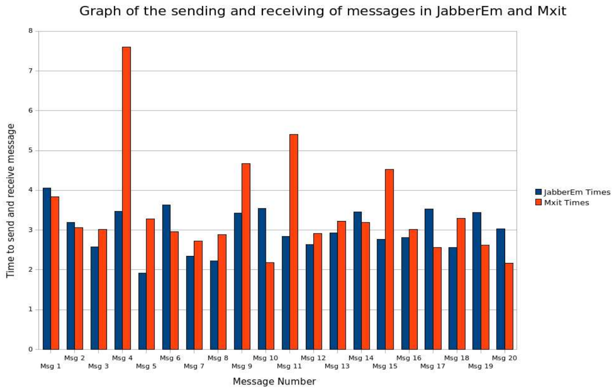


**Figure 2: Peak hours message transfer times**

During off-peak hours, the amount of time required to send twenty messages on MXit equated to 69.171 seconds. During peak hours the time required to send twenty messages equated to 248.770 seconds which is 359.64% slower than sending the messages during off-peak hours. During off-peak hours the time needed to send twenty messages on JabberEm equated to 60.417 seconds. The time required during peak hours equated to 125.155 seconds, which is 207.15% slower than sending the messages during off-peak hours. During peak hours MXit is 123.615 seconds slower than JabberEm at sending twenty messages. Clearly during peak and off-peak hours JabberEm is able to deliver messages more quickly, which in turn results in less user frustration and confusion.

#### *4) Experiment 4: KLM evaluation of user interface*

Modelling user tasks and processes has increasingly become an important factor in user interface design. The GOMS (Goals, Operators, Methods, Selection rules) model allows for the modelling of user behaviour during given tasks and also analyses use complexity for interactive systems [18]. The KLM is a customized instance of GOMS and allows tasks to be described using operators that model component tasks such as key presses, mouse movements, decision timing, and system response times.

Since the KLM is aimed at modelling user behaviour on desktop computers, it cannot be directly applied the modelling of user behaviour on mobile devices. A variation of the KLM tailored for mobile devices has been produced by Holleis, Otto, Hußmann and Schmidt [19], KLM for Advanced Mobile Phone Interaction (KLM-AMPI) which includes slight modifications of the original KLM. KLM-

AMPI introduces new operators such as macro attention shifts, micro attention shifts, and finger movements, to allow a more accurate model to be created of mobile-based interaction. In order to evaluate the efficiency of the JabberEm and MXit's user interfaces, it was decided to use KLM-AMPI to determine the time required to perform the following core components of an instant messaging client: send message; add contact; remove contact; show offline contacts; change contact nickname; change current user nickname; and message vibrate.

The results of the evaluation are based on the assumption that the user has the cell phone in hand, and that predictive text is activated. Since contacts may be listed in different positions on the contact list, the number of key presses required to select a contact was considered to be one. As this is constant across both interfaces, it does not affect the outcome of this analysis. All evaluation was performed on a Nokia N95 8GB handset.

It should be noted that the time required to send the message was not taken into account. All test results are independent of the time needed for data transmission. This is purely an indication of the time required to navigate and execute various functions embedded in the user interfaces of JabberEm and MXit.

| | *JabberEm* | *MXit* |
|---|---|---|
| *Send message* | 5.66 | 5.88 |
| *Add MXit contact* | 6.43 | 9.56 |
| *Add Google Talk contact* | 14.26 | 28.15 |
| *Remove contact* | 2.04 | 4.09 |
| *Show offline contacts* | 2.08 | 4.77 |
| *Change contact nickname* | 3.31 | 6.46 |
| *Change current user nickname* | 3.82 | 9.03 |
| *Message Vibrate* | 1.52 | 4.16 |
| ***Total*** | 39.12 | 72.10 |

**Table 4: Predicted user interface navigation time using KLM-AMPI (seconds)**

From Table 4 it can be deduced that JabberEm has a considerably more efficient user interface than MXit. This is can be explained by considering that JabberEm only has one deck, namely the "Settings" deck, whereas MXit has multiple decks as well as sub-decks. Since users only ever need to navigate one deck in JabberEm in order to carry out all the functions in Table 4, this results in a more efficient user interface than MXit where users often need to navigate multiple decks and sub-decks.

*5) User evaluation of applications*

As described in the methodology section of this paper, timing methods were used to quantify learnability, efficiency and memorability of the different interfaces. Results from these tests have been graphed below and are provided in Figures 3 and 4. From Figure 3 it can be seen that JabberEm has a much lower learnability time than MXit, meaning that JabberEm is easier to use for the first time when compared to MXit. Figure 3 was obtained by taking the total time for each user to perform each of the seven core functions in both JabberEm and MXit. JabberEm had a total time of 376.33 seconds and MXit had a total time of 602.79 seconds.

To calculate the efficiency of the applications, the users were asked to cycle through each of the core functions twice. The average of the two core function cycle times for each user was then obtained and summed. JabberEm obtained a total time of 268.13 seconds and MXit obtained a time of 337.7 seconds. JabberEm proves to be a lot faster at establishing proficiency than MXit. Figure 4 shows that JabberEm has a much better learnability than MXit, and since it is initially easier to use than MXit, it is logical to expect that users would be able to establish proficiency more easily in JabberEm than MXit.

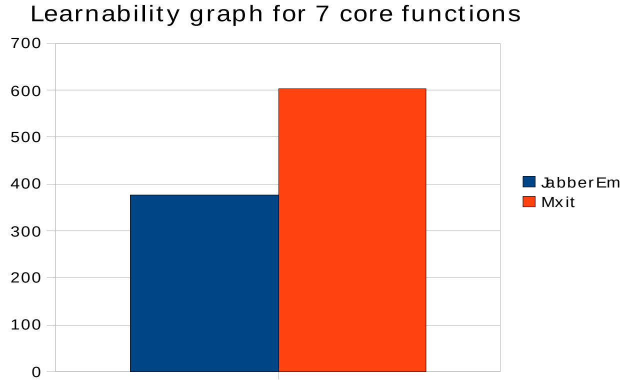


**Figure 3: Total time taken to perform core functions**

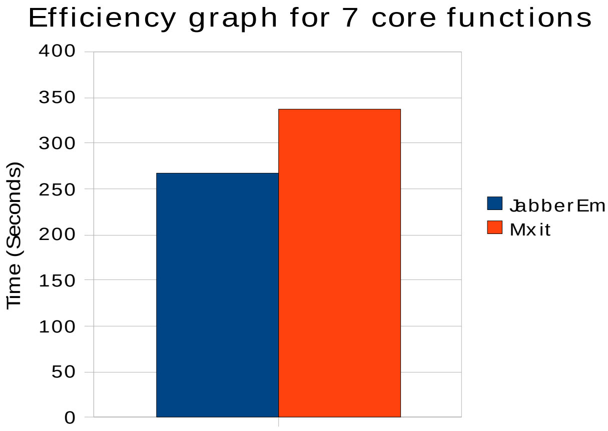


**Figure 4: Total time to perform core functions twice over**

The six users that performed the user evaluation are represented by three males and three females. User 5 (Male) had used MXit extensively before the user evaluation for this thesis. User 1 (female) also used MXit extensively before the user evaluation. Even with two out of the six users being regular users of MXit, their timings from JabberEm still proved to be better than those from MXit.

To prove the accuracy of the Keystroke-Level Model for Advanced Mobile Phone Interaction (KLMAMPI), the times taken for the six users to send a message in JabberEm was totalled an averaged. The average time of the users, excluding the transmission time of the packet over the network equals 5.5609 seconds which is remarkably close to the calculated KLMAMPI value of 5.66 seconds.

JabberEm has an efficiency average of 44.68 seconds and a memorability average of 51.08 seconds. MXit has an efficiency average of 56.28 seconds and a memorability average of 73.63 seconds. In order to calculate the memorability, the times obtained in the memorability tests of both applications cannot simply be compared, but the difference in time between efficiency results and the results obtained two days after this (i.e. the memorability results) are needed. The difference between the average efficiency time and the average memorability time of JabberEm is 6.4

seconds, and the difference between the two times for MXit is 17.35 seconds. Clearly JabberEm has a much more memorable interface than MXit.

JabberEm thus has faster learnability, efficiency and memorability times, and it can be concluded, with a high degree of certainty that JabberEm is a much more usable and efficient application than MXit. JabberEm not only has a more efficient user interface than MXit, but it is also outshines MXit in message delivery time.

## IV. Conclusion

Without formal and user evaluation of applications developers have little idea of the quality of the developed applications. Formal evaluation makes use of quantitative methods and scientifically proven principles in order generate a set of results which can be used to analyse the efficiency of certain aspects of an application. Since these applications are of no use without user interaction, it is very important to take into account user opinions and evaluations.

This paper has introduced ThEm, a suite of applications designed as a convergent technology for text based communication on a mobile phone. As each of the components of ThEm has a consistent theme, this paper has presented a detailed evaluation of a particular component, JabberEm as a comparison to a leading off the shelp alternative MXit. This evaluation consisted of three main components. First the core timings of sending and receiving messages between desktop and mobile clients was presented. Second, a predictive evaluation of the user interface is undertaken and compared to MXit using the Keystroke-Level Model for Advanced Mobile Phone Interaction. And finally, a user evaluation is presented to quantify the efficiency and learnability of the interface. Across all experiments, JabberEm has received comparable if not better results than MXit.

## V. Acknowledgements

This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom, Comverse, Stortech, Tellabs, Amatole Telecom Services, Mars Technologies, Bright Ideas 39, and THRIP.

## References

[1] Preece, J. Rogers, Y. and Sharp, H. (2002) "Interaction Design, beyond human-computer interaction", Wiley

[2] Bedlow, R. (2008) "Blurring the Lines", Mobile World Congress Daily.

[3] Answers.com. (2008 ) "Cellular Telephone". Available At:http://www.answers.com/topic/mobile-phone?cat=technology. [Accessed 08-08-2008]

[4] Jussawalla, M. (1999) "The impact of ICT convergence on development in the Asian region" in Telecommunication Policy. Published by Elsevier Science. Vol 24. p. 217-234

[5] CNN, (2003) "Cell Phone quality study ranks providers"http://money.cnn.com/2003/07/31/technology/cellular_survey/index.htm?cnn=yes [Accessed 08-05-2009]

[6] IETF (2004) "Extensible Messaging and Presence Protocol (XMPP)" Standard. Internet Engineering Task Force. [Accessed 08-05-2009] Accessed at: http://www.ietf.org/html.charters/OLD/xmpp-charter.html

[7] Vodacom4me (2009) "On the Web: Vodacom4me". Vodacom, Available at: http://www.vodacom.co.za [Accessed 08-05-2009]

[8] Knudsen, J. (2007) "Kick Butt with MIDP and MSA: Create Great Mobile Applications". Prentice Hall PTR, Upper Saddle River, NJ, USA.

[9] Zheng, P. and Ni, L. M. "Smart Phone & Next Generation Mobile Computing". Morgan Kaufmann Publishers, San Francisco, 2006. 440 - 441, 237 – 239.

[10] B'Far, R., Richards, R. and Ditlinger, S. (2008) "Designing Effective User Interfaces for Wireless Devices". By quantifying quality assurance and use case analysis, you can create an efficient wireless user interface for transmitting content via XML. Available At: http://archive.devx.com/wireless/articles/rb0501/rb0501-1.asp. [Accessed 05-05-2008]

[11] Vogiazou, Y. (2002). "Wireless Presence and Instant Messaging". JISC Technology & Standards Watch. TSW 02-07. JISC: Bristol. Available at: http://www.jisc.ac.uk/index.cfm?name=techwatch_report_0207. [Accessed 17-04-2008]

[12] Kenteris, M., Economou, D. (2006) "A Novel Method for the Development of Personalised Mobile Tourist Applications" 5th IASTED International Conference on Communication Systems and Networks (CSN'2006), Plama de Mallorca, Spain, 28-30 August 2006.

[13] Hays, G. "Successful User Interfaces for Mobile Devices". Available At: http://www.developer.com/ws/article.php/10927_1143541. [Accessed 14-04-2008]

[14] Sun (2002) "Java Card Technology Overview", Sun Developer Network (SDN). Available At:

http://java.sun.com/javacard [Accessed 05-09-2008]

[15] Sun (2008) "Java ME at a Glance", Sun Developer Network (SDN). Available At: http://java.sun.com/javame [Accessed 16-08-2008]

[16] Mallick, M. (2003) "Mobile and Wireless Design Essentials". John Wiley & Sons. p117-118.

[17] Yeng, J. (2005) "Usability Assessment of Academic Digital Libraries: Effectiveness, Efficiency, Satisfaction, and Learnability ", Libri, vol. 55, pp. 96–121.

[18] Card, S. K., Moran, T. P., Newell, A. (1980), The Keystroke-Level Model for User Performance Time with Interactive Systems. Comm. ACM 23, 7. 396-410.

[19] Holleis, P., Otto, F., Hußmann, H. and Schmidt, A. (2007) "Keystroke-Level Model for Advanced Mobile Phone Interaction". SIGCHI Conference on Human Factors in Computing Systems 2007 (CHI '07), San Jose, CA, USA, April/May

**Curtis Sahd** is currently undertaking his Masters at Rhodes University investigating Overlaid Bluetooth piconets for the construction of MANETS and PANS.