Enabling Interoperability Between Mobile IM and Different IM Applications Using Jabber

MOHD HILMI HASAN, ZURAIDAH SULAIMAN, NAZLEENI SAMIHA HARON, AZLAN FAZLY MUSTAZA
CIS Department
Universiti Teknologi PETRONAS, 31750 Tronoh, Perak
MALAYSIA

Abstract: - The two most popular text-messaging systems nowadays are Instant Messaging (IM) and Short Messaging System (SMS). IM runs in computers, where SMS runs through mobile phones. Nowadays, there are numerous efforts done to merge the functionalities of IM into mobile phone. This is due to the fact that IM can provide "presence" of users and less costly. In order to provide the best satisfaction to its users, the mobile IM should be able to communicate to multiple IM service providers. In relation to that, this paper presents the research conducted to develop a mobile IM named Message Conveying System (MCS) that allows it to enable communication with other IM service providers. This was made possible through the implementation of Jabber protocol in MCS, which results in its interoperability with other Jabber-based IM service providers. The MCS was tested to be successfully run on the mobile phone and communicate with the other two Jabber-based IM service providers; Psi and Spark.

Key-Words: - Mobile instant messaging, IM, Jabber.

1 Introduction

Instant Messaging (IM) is a popular communication tool used for sending messages via the computer. Apart from its nature for chatting among colleagues, the use of IM has also been extended to some other purposes, for example, as a tool to provide services or helps for users [1]. The use of IM has also grown incredibly, thanks to the increasing number of internet users which surpassed 930 million in 2004 [2]. As opposed to IM, Short Messaging System (SMS) is a text-based tool for sending messages using mobile phones. Currently, there are many efforts done to merge the functionalities of IM into mobile phones. It serves the advantages of using IM services without needed to be attached to the computer, and provide the "presence" of the users. According to [3], IM will have a big impact on wireless services and m-commerce applications.

To date, there are numerous IM service providers available for use, to name a few, AOL and Yahoo. Some of these service providers offer mobile IM service. However, it is not always possible to use IM services between different service providers. This would be diverting from users' expectation that are hoping for interoperability. Interoperability problem must be handled carefully since it is vital for the success of IM [4]. User will only derive value from an IM application that is capable of universal communication, and interoperability would promote the expansion of IM technology into a useful application and communication medium [5]. Therefore lack of interoperability will cause IM to be underutilized. Interoperability has also been identified as one of the challenges that hinder the IM market. It is also one of the pressing issues that currently mobile phone companies need to address since service interoperability between service providers is the key factor for commercial deployment success and revenues [6, 7]. The interoperability issues are related to both desktop and mobile IM. This research was conducted to focus on the issue in the mobile IM.

One way to solve the interoperability issue in mobile IM is by implementing Jabber protocol. This is possible because there are a number of Jabber IM clients available nowadays, to name a few Exodus, Psi and Spark. Jabber is a protocol that consists of a set of streaming XML to enable any two entities on the Internet to exchange messages [8]. Jabber can for example bridge AOL and another IM service provider; hence the user can talk to his friends who are using applications other than AOL [9]. Besides that, Jabber has also being implemented in Telebuddies [10], a gaming system that involves interactive digital television (iDTV) users and the mobile participants. In this project, Jabber protocol enables the data communication between the Telebuddies system and the client mobile devices assisting the chat communication between all heterogeneous participants in the Telebuddies via the ubiquitous XMPP protocol.

Apart from the mobile IM applications, there is another technology known as “Push-to-talk” that is gaining wide acceptance and attempting to do the same; which are discover buddies, and talk to them by merely pushing a button like a walkie-talkie. The basis for “presence” in “Push-to-talk” that is provided by several telecommunication provider is derived from XMPP [11]. It is designed to let the user uses voice communication. In contrast, mobile IM is focusing on providing an alternative for text communication i.e. SMS.

The objective of this research is to enable interoperability of the Mobile IM application with different IM service providers. The IM application developed in this research was named Message Conveying System (MCS). The mobile phone that is installed with MCS is going to be able to communicate not only to another mobile phone that is installed with the MCS, but also with other IM service providers that implement Jabber protocol. This paper presents the implementation of Jabber protocol in the development of MCS, and the testing of its interoperability with different IM service providers.

2 Related Works

Many works have been done related to the mobile IM and IM interoperability. One of the studies showed that mobile IM can improve chat spontaneity and response rate. This was achieved using Affective Gesture approach in its implementation [12]. The study was also done to analyze the effect of “presence” on the message content. The result of this study showed that the “presence” state offered by IM does affect the message content sent by the users [13]. Moreover, mobile IM’s usage has also been extended to be able to interpret sound rather than text. This is especially beneficial for those users who are having spontaneous interactions with their friends [14].

The research was also done to develop mobile IM using Jabber. This application serves as a wireless-based text chatting system, which provides the users with chatting activities using their mobile phones [15]. Moreover, there are also some industry players coming out with the mobile IM clients. One of them is Tipic, which has a product known as TipicME to serve as IM client on mobile phone. TipicME was developed using XMPP/Jabber compatible servers through a wireless proxy [16]. Another project that utilizes Jabber is SPADE (Smart Python multi-Agent Development Environment) [17], a new multiagent-system platform that take advantages of the communication model based on IM protocol. Communications in SPADE that use real-time messaging between applications are handled by the assistance of Jabber protocol. The proliferation of mobile multimedia-enabled devices for real-time communication has also encouraged IM to support multimedia interactions. Hence, Jabber has been given new enhancements in its usage by the development of a new set of multimedia extensions such as in Jingle project [18].

3 Methodology

This paper presents the interoperability of the mobile IM application that has been developed, which is called MCS. The Jabber protocol was used as a protocol in MCS’s implementation, so that it could communicate with other MCS in different mobile phone as well as other IM service providers installed in mobile phone or desktop.

Having MCS implemented with Jabber protocol, it was then installed in a mobile phone for testing. Two IM service providers were used in this research to test on the interoperability. The two IM service providers are Psi and Spark. The Motorola SLVR mobile phone was used in this research to test the MCS’s operation. However, for the purpose of this paper, all the results are shown using Java Wireless Toolkit software [19].

4 Application Development

4.1 Jabber Architecture

Jabber IM uses a standard protocol named Extensible Messaging and Presence Protocol (XMPP) [20]. In Jabber environment, the user addresses are in the form of user@host. Jabber implements a distributed client-server architecture, rather than peer-to-peer that is used by some other IM applications [8]. Any messages that the user sends is sent to the server. This server sends the messages to the other server, which then relays them to the recipient based on the user address as shown in Fig. 1 below.

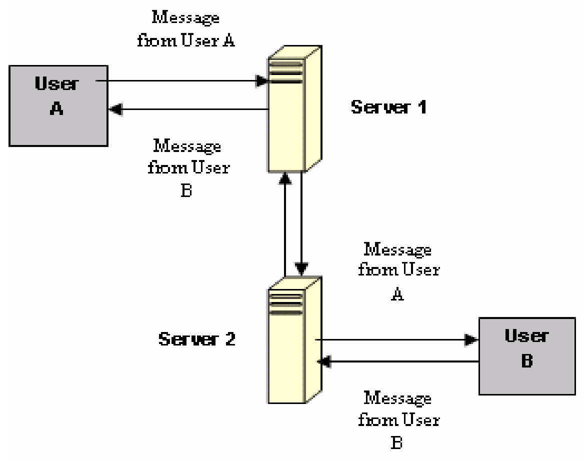


Fig.1. System architecture

The Jabber architecture has been designed so that its client is simple enough, and contrarily, all the important functionalities reside in its server. The server communicates with the clients and other servers by exchanging structured data and instant messages. This is done using streaming XML protocols. It also provides a means for interoperability with other commercial IM systems [8]. This research takes advantage of Jabber protocol in order to enable MCS to communicate with other Jabber-based IM service providers.

4.2 Structure of Jabber XML Protocol Implemented in MCS

Jabber application, which involves client and server, communicates through exchanging XML. Two XML streams are involved in the exchange of data, one stream delivers the data packet from the client to the server, and another delivers the data from the server to the client. In this research, MCS was developed with the code fragment shown in Fig.2 for the process of XML exchange between the Jabber server and the client.

```
SEND: to='aslan.net'
SEND: xmlns='jabber:client'
SEND:
xmlns:stream='http://etherx.jabber.org/
streams'>
RECV: <stream:stream
RECV:
xmlns:stream='http://etherx.jabber.org/
streams'
RECV: id=’38fd8070’
RECV: xmlns='jabber:client'
RECV: from='aslan.net'>
(XML for user session goes here)
SEND: </stream:stream>
RECV: </stream:stream>
```

Fig.2. XML exchange between server and client.

Then, the application needs to handle the sending of messages between two users. This is implemented using `<message />` element. The code fragment in Fig.3 shows the implementation of this element.

```
<message type='chat' from =
'ayeen@aslan.net/JabberMCS'
To='aslan@aslan.net/JabberMCS'>
<body>Hello!</body>
</message>
```

Fig.3. Structure of the message element.

Jabber provides Jabber Session Manager (JSM) to cater messages regardless of the status of the user. The message will be delivered instantly if the user is online or otherwise, it will be stored and delivered when the user comes online.

Another element that is implemented in the development of MCS is `<iq />` element. It manages the conversation between any two users, as well as allowing them to pass XML-formatted queries and respond accordingly. The code fragment in Fig.4 shows the implementation of the `<iq />` element in the application.

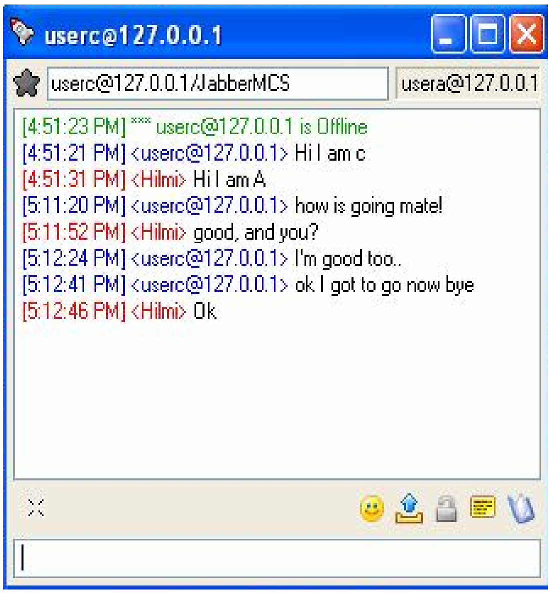
```
<stream:stream id='38fd8070'
from='aslan.net' xml:lang='en'
xmlns='jabber:client'
xmlns:stream='http://etherx.jabber.o
rg/streams'>
<iq to='ayeen@aslan.net/JabberMCS'
id='s3' type='result'>
<query xmlns='jabber:iq:roster'>
<item name='aslan'
subscription='both'
jid='aslan@aslan.net'>
<group>love</group></item>
<item name='mfuzze'
subscription='both'
jid='mfuzze@aslan.net'>
<group>love</group></item></query></
iq>
</stream:stream>
```

Fig.4. Structure of the iq element.

5 Results

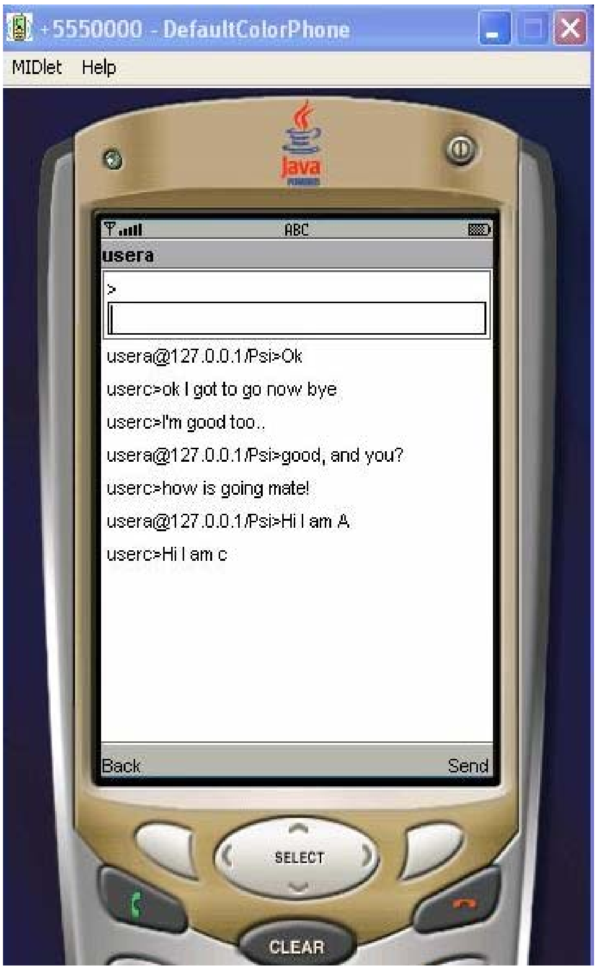
There were two Jabber IM service providers used in this research for the purpose of testing and generating the results. Both applications, Psi and Spark were installed in a desktop PC. The MCS was installed in the Motorola SLVR mobile phone for this testing. However, for the purpose of presenting the snapshots in this paper, the Java Wireless Toolkit Emulator is used for MCS client.

In the testing part, the Psi client was registered with the nickname “usera”, and MCS client as “userc”. Fig.5 and Fig.6 show that the “usera” can interact with “userc”.



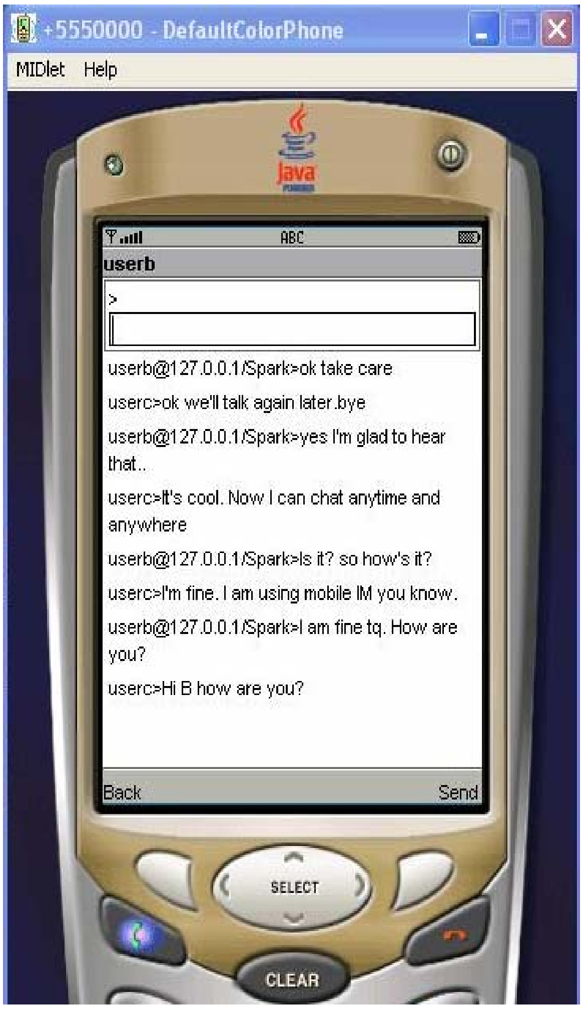
[4:51:23 PM] *** userc@127.0.0.1 is Offline
[4:51:21 PM] <userc@127.0.0.1> Hi I am c
[4:51:31 PM] <Hilmi> Hi I am A
[5:11:20 PM] <userc@127.0.0.1> how is going mate!
[5:11:52 PM] <Hilmi> good, and you?
[5:12:24 PM] <userc@127.0.0.1> I'm good too..
[5:12:41 PM] <userc@127.0.0.1> ok I got to go now bye
[5:12:46 PM] <Hilmi> Ok

Fig.5. Psi client as "usera" interacts with "userc".



>
usera@127.0.0.1/Psi>Ok
userc>ok I got to go now bye
userc>I'm good too..
usera@127.0.0.1/Psi>good, and you?
userc>how is going mate!
usera@127.0.0.1/Psi>Hi I am A
userc>Hi I am c

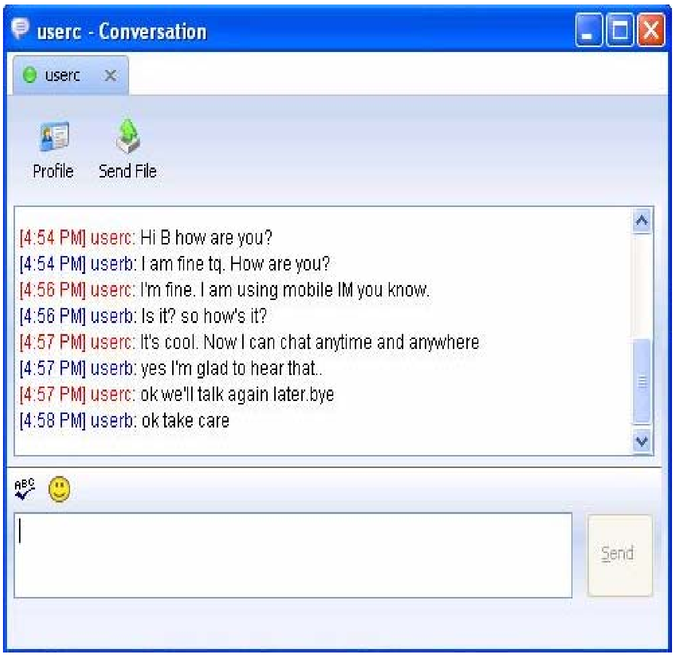
Fig.6. "userc" interacts with "usera".



>
userb@127.0.0.1/Spark>ok take care
userc>ok we'll talk again later.bye
userb@127.0.0.1/Spark>yes I'm glad to hear that..
userc>It's cool. Now I can chat anytime and anywhere
userb@127.0.0.1/Spark>Is it? so how's it?
userc>I'm fine. I am using mobile IM you know.
userb@127.0.0.1/Spark>I am fine tq. How are you?
userc>Hi B how are you?

Fig.7. "userc" interacts with "userb".

Fig.7 shows the same MCS client is now interacting with another client, which is Spark. Spark client was registered as "userb". Fig.8 below shows the Spark client and its communication with "userc".



[4:54 PM] userc: Hi B how are you?
[4:54 PM] userb: I am fine tq. How are you?
[4:56 PM] userc: I'm fine. I am using mobile IM you know.
[4:56 PM] userb: Is it? so how's it?
[4:57 PM] userc: It's cool. Now I can chat anytime and anywhere
[4:57 PM] userb: yes I'm glad to hear that..
[4:57 PM] userc: ok we'll talk again later.bye
[4:58 PM] userb: ok take care

Fig.8. Spark client as “userb” interacts with “userc”.

6 Conclusion

This paper presents a development of MCS using Jabber protocol to enable its interoperability with different IM service providers. Two IM service providers have been chosen in testing the outcome of the research. Those two are Psi and Spark. The objective of the research, which is to enable interoperability of MCS, has been achieved. MCS can also communicate with a few other Jabber IM clients like GoogleTalk, Meebo and Exodus. Based on the outcome of the research, more research work can be done in this area of mobile IM especially in engaging Jabber-based IM application like MCS with third-party IM solutions such as Yahoo and MSN.

7 Acknowledgement

The authors would like to thank Mr. Zulqarnain Abu Bakar for his assistance and contributions in producing this research paper.

References:

[1] Snyder, J., Field, M., We Got the Message: Creating an Instant Messenger Service Point, *SIGUCCS’06*, ACM, 2006.

[2] http://www.etforecasts.com/pr/pr904.htm, 2005.

[3] Debbabi, M., Rahamm, M., The War of Presence and Instant Messaging: Right Protocols and APIs, *IEEE CCNC2004*, Las Vegas, 5-8 January 2004.

[4]Woods, http://www.instantmessagingplanet.com/wireless/article.php/964591, 2002.

[5] McClea, Michael, Yen, David and Huang, Albert H., An analytical study towards the development of a standardized IM application. *International Journal of Computer Standards and Interfaces*, Vol.26 i4. 343-355, 2004.

[6] E. Sylvers, Wireless: Medium and messaging - Instant versus text, http://www.iht.com/articles/2006/07/10/business/wireless11.php, 2006, pp. 1-3.

[7] Neustar, Mobile Instant Messaging : The Next Global Messaging Opportunity, http://whitepaper.samag.com/cmpsamag/ActivityServlet?ksAction=displaySolutionSummaryView&solId=85813, 2007, pp. 1-16.

[8] http://www.jabber.org, 2005.

[9] Searls, D., The Next Bang: The Explosive Combination of Embedded Linux, XML and Instant Messaging, *Linux Journal*, Vol. 2000, 2, 2000.

[10] Luyten, K., Thys, K., Huypens, S., and Coninx K., Telebuddies On The Move: Social Stitching To Enhance The Ubiquitous Networked Gaming Experience, *Netgames'06*, Singapore, ACM, 2006.

[11] XTech, 2005. http://idealliance.org/proceedings /xtech05/papers/04-04-02/.

[12] Adesemowo, A.K., Tucker, W.D., Instant Messaging on Handhelds: An Affective Gesture Approach, *SAICSIT’05*, ACM, 2005.

[13] Ljungstrand, P., Segerstad, Y.H., Awareness of Presence, Instant Messaging and WebWho, *ACM SIGGroup Bulletin*, ACM Press, 2000.

[14] Isaacs, E., Walendowski, A., Ranganathan, D., Hubbub: A Wireless Instant Messenger That Uses Earcons for Awareness and for “Sound Instant Messages”, *CHI’01*, ACM Press, 2001.

[15] Gao, J., Modak, M., Dornadula, S., Shim, S., Mobile Jabber IM: A Wireless-Based Text Chatting System, *CEC 2004*, IEEE, 2004.

[16] http://www.tipic.com/

[17] Gregori, M. E., Cámara, J. P., and Bada, G. A., A Jabber-Based Multi-Agent System Platform, Japan, *AAMAS '06*, ACM, 2006.

[18]Saint-Andre, P., Jingle: Jabber Does Multimedia, *IEEE Multimedia*, Vol.14, Issue 1, Jan.-March 2007.

[19] Java.sun.com, Java Wireless Toolkit

[20]Jabber Software Foundation. Extensible Messaging and Presence Protocol (XMPP): Core Technical Support, http://www.ietf.org/rfc/rfc3920.txt, May 2006.