

XMPP and iDTV or How to Make Television a Social Medium

Kevin Hoekman, Michiel Ide, Tom Deryckere, Luc Martens

Wireless and Cable group, Department of Information Technology, IBBT-Ghent University
Kevin.Hoekman@UGent.be, Michiel.Ide@Intec.UGent.be, Tom.Deryckere@Intec.UGent.be,
Luc.Martens@Intec.UGent.be

Abstract

Instant Messaging (IM) has the potential to become one of the killer applications for interactive Digital Television (iDTV) [12]. However, several factors make it difficult to provide a good implementation of IM services, among which the limited resources of a set-top box and the different user experience compared to computer environments.

This paper proposes the XMPP (Extensible Messaging and Presence Protocol) standard as a solution for implementing IM. When we compare XMPP with other technologies, it reveals itself to be very well adapted to the specific needs of iDTV middleware platforms like the Multimedia Home Platform (MHP) [5].

Moreover, the use of XMPP doesn't limit itself to IM. The flexible architecture of XMPP opens a window of opportunities like the ease of adding new interactive services. To demonstrate the possibilities of XMPP on MHP, an IM client –IM4MHP– is presented in this paper.

1. Introduction

Since several years now, Instant Messaging (IM) has turned out to be an extremely popular application for computers and other devices. In spite of this, a lot of research still has to be done in order to put it into practice on an interactive Digital Television (iDTV) platform [12].

Although most of the concepts and functionality of traditional IM in a computer environment are reusable for iDTV, there are some core differences. First of all there's the difference in audience; iDTV targets a much broader audience who may be unfamiliar with the use of a computer or IM. Secondly the equipment differs; limited graphics display and interaction possibilities (remote control) but also the limited resources (storage, processing power, memory) on a set-top box. A final difference we would like to

mention here is related to the difference in perception. Not only is the viewing distance totally different but also the social context differs. Watching television is most of the time a group experience while IM on a computer is a more individual activity.

An overview of some of the functional requirements is made in the next section where we discuss typical IM functionality for an iDTV environment.

In order to achieve these requirements there are many distinct solutions available. In section three we will show the advantages and disadvantages of the chosen technology and compare it with another well known technology.

The last sections show the adaptation to a Java environment and give a brief look on the prototype we have built on such an environment: the MHP platform [5].

2. Functional Requirements

IM has the potential to become a “killer application” for iDTV but only if we adjust the usage specifically to an iDTV environment. Some requirements however remain the same in every environment, like having the ability to:

- create, destroy and maintain an account,
- adjust your presence information and maintain the list of contacts, let's call them buddies, who may see this information, and
- send real time messages to one or multiple users.

The following innovations make the usage of IM in an iDTV environment really attractive.

- Let the buddies share their current viewing experience.
- Let a buddy know how you feel about the program you are watching.
- Give a buddy a TV program recommendation.
- Expand your social network by joining a chatroom associated with a certain service.

3. Extensible Messaging and Presence Protocol (XMPP)

This section introduces XMPP [8] as a technology and how it is used for IM. As with all technologies some advantages and disadvantages are associated with the usage of XMPP. Once the reader is familiar with XMPP, we will compare it with another major Internet Engineering Task Force (IETF) standard and show why XMPP is considered to be the best solution for iDTV. To end this section other uses of XMPP within an iDTV environment are shown.

3.1. Technology

The XMPP communication model follows a client/server architecture with distributed servers (similar to email systems). Figure 1 illustrates how users are divided in different domains. To send a packet to a particular recipient, the initiator addresses the packet to his recipient and sends it, over a TCP connection, to his own domain server (arrow 1). This server sends the packet to the domain server of the recipient (arrow 2). Finally the recipient's domain server sends the packet to the recipient (arrow 3).

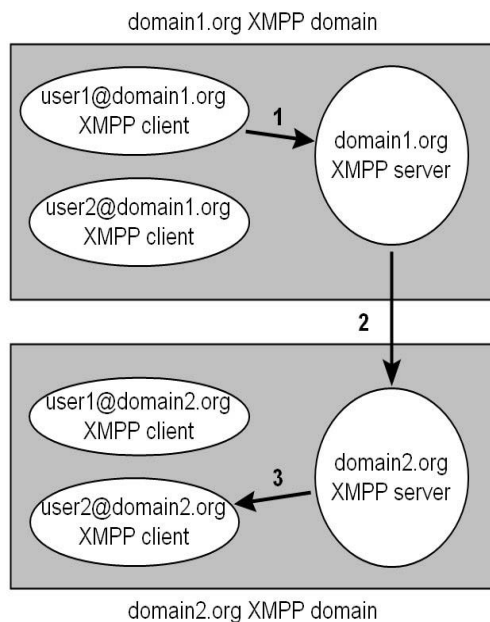


Figure 1 : XMPP architecture

To communicate with each other, the initiator and the recipient set up two XML streams; one in each direction, since XML streams are unidirectional. Once this is done, they can exchange packets through these streams. Each packet is a well-formed XML fragment.

There are only three main XMPP tags (also called stanzas):

- `</message>` : exchanging messages,
- `</presence>` : controlling presence information,
- `</info/query>` : easily extendable request/response mechanism for all the functionality that can not be achieved through the first two stanzas.

Each stanza can have different child tags to specify it further. An example can be a `</status>` tag as a child for the `</presence>` stanza. By using this tag you can tell what you are doing, e.g.

`<status> I'm enjoying my dinner </status>`.

3.2. Advantages

By choosing for a client/server architecture, lightweight protocols and limited communication scenarios (client/server and server/server), one can establish very lightweight clients. The more complex and resource-demanding issues fall under the responsibility of the server (presence and status management, packet routing, user account management, storing user or configuration information...). This characteristic is very interesting for implementation on a device with limited resources as a set-top box.

Other benefits related to the client/server architecture are security and privacy; clients only have direct communication with their server, hiding their location to other users. There are no firewall or NAT issues because every session is initiated by the client. There is also the centralized control over the domain by which it is possible to enforce policies like a spam filter. A last advantage of the client/server architecture we would like to mention is the fact that by using separate domains, XMPP becomes very scalable. This is very important because IM has over 300 million regular users worldwide.

XMPP offers the possibility to interact with any device supporting an XMPP client (PC, mobile phone, PDA...). Through a special server-side translation service, called a gateway, an XMPP client can also interact with other IM services (AIM, ICQ, MSN Messenger, Yahoo! Instant Messenger, ...), as well as with other technologies (SMS, email, ...).

3.3. Disadvantages

The choice of XML makes the bandwidth usage less efficient in comparison to binary formats. With the upcoming broadband connections this is not considered as a main issue. The parsing of the received XML packets on the other hand can be much more critical, as it can be quite resource demanding. Nevertheless, we do not need a full-featured XML parser, the only requirement it must meet, is parsing XMPP packets. Because XMPP uses only a small part of the total functionality of XML, the parsing can be done by a lightweight parser.

The fact that no direct communication is possible between clients can be a serious handicap for some applications, especially those exchanging a lot of data, for example file sharing. In this scenario, all communication goes through the servers, which could be overloaded by the intense traffic. To solve this problem a (non-final at this time) extension is available [3] which allows out-of-band data exchange. Peer to Peer (P2P) systems are also a solution for this problem but for an iDTV environment they are not suited. Advantages of a client/server architecture are not offered by a P2P system but are absolutely necessary for an iDTV environment, as discussed in section 3.3.

Another limitation is the fact that the XMPP core standard does not demand in any way QoS to the clients and servers. This means that there is no guarantee for the time of delivery, the order of delivered packets nor that the packet will arrive once and only once [13].

3.4. XMPP versus SIP/SIMPLE

The most important comparison is the one between XMPP and SIP/SIMPLE (Session Initiation Protocol [10] / SIP for Instant Messaging and Presence Leveraging Extensions [14]), since both are IETF standards who are competing to become the IM standard of the future. The following shows some of the most important drawbacks of the use of SIP/SIMPLE.

The most relevant disadvantage of SIP/SIMPLE is its early standardization phase. This has important consequences for the interoperability. Companies who want to use SIP/SIMPLE fill in the blanks with other (proprietary) technologies and so it happens that different systems are no longer compatible. For example, Microsoft and IBM, two companies who believe in SIP/SIMPLE, have incompatible implementations.

SIMPLE is based on SIP and that brings along some disadvantages as well. The following are some

examples to illustrate this. SIP doesn't restrict the transport protocol; it allows the use of TCP as well as UDP, which can make the delivery unreliable. Most of the media is sent to dynamic ports and addresses which can make it difficult to set up a firewall. Although XMPP and SIP are both text based technologies and therefore bandwidth inefficient, SIP has much more overhead than XMPP which can cause a baleful influence on the scalability.

Until recently there was one good reason to choose for SIP/SIMPLE and that was its support for multimedia. This support is essential if you want to offer a total real-time solution with IM, Voice over IP (VoIP) and videoconferencing (triple play). Since March this year however XMPP developed a new protocol for this purpose, Jingle [15] that defines a framework to create and maintain outbound sessions and will be used in, for example, Google Talk [7].

It will be very interesting to follow the further evolution of both technologies. Will one standard oust his competitor, or will they keep coexisting?

3.5. Other uses of XMPP

XMPP is much more than an IM protocol; basically it's a real time mechanism to exchange structured data between entities. This data is not restricted to IM data and what is more, the entities do not have to be human, but can be any entity connected to the network. This opens lots of opportunities in general but also specific for iDTV.

A first feature that is worth mentioning is the use of chatbots. With chatbots you can offer lots of services to the end users. The usability can be increased by means of a companion guide or basic customer support. Actually it is possible to offer all kinds of information through chatbots like weather reports, lottery numbers... But maybe the most attractive feature (from a commercial point of view) is the possibility to support 'impulsive buying' [11] through chatbots. For example, during a music video a chatbot sends you, through the associated chatroom, a link message. When you choose to follow the link, you end up in the t-commerce shop, where you can easily buy the song or even the music video.

A second application area for XMPP is inter-application communication. In this context XMPP can be considered as a transport protocol for, for example, RPC (Remote Procedure Calls) [2] and SOAP (Simple Object Access Protocol) [6]. For the latter one XMPP offers some benefits in comparison with the most frequently used transport protocol for SOAP, namely HTTP. A major advantage is the possibility to deliver asynchronous messages.

4. Java based XMPP library

In section 5 we will discuss the IM client we have built for the MHP platform. Since MHP is a Java based specification, it would be very convenient to have a Java based XMPP library. It became obvious that no such public library existed for the MHP platform. So there were two options, build one from scratch or adjust an existing one. So let's take a deeper look at the different existing Java libraries (**Table 1**) after which it will become clear that the last option is the way to go.

Table 1: Characteristics of XMPP client libraries

Name	#extensions	#dependencies	doc
Echomine Muse	22	4	++
JabberWookie	0	3	-
Smack	9	1	++
Yaja	0	5	-
Micro-jabber	0	0	-

All of these libraries have a licensing model that allows adjustments. Echomine Muse [4] and Smack [9] turn out to be the most extensive and documented ones. Compared to the other libraries, they both offer a higher level of abstraction, by hiding the lower level details of XMPP. The most important benefit of the Smack library is that it depends on only one library that's not part of MHP, i.e. a lightweight XML parsing library. The latter justifies the choice for the usage of the Smack library in our prototype.

From version 1.1.2 of the MHP middleware stack onwards, MHP will be based on J2ME. For this reason we included Micro-jabber in our comparison since this library is also based on J2ME. This is only done for future purposes since our current test environment is built upon set-top boxes with MHP 1.0.2.

5. IM4MHP, a prototype

The equipment we used to build this prototype was an ADB Q75-DEV box with the Osmosys MHP 1.0.2 middleware stack.

Figure 2 shows a screenshot of the 'IM4MHP' application. It shows the scenario of sending a message (with subject and body) to someone from your contact list.

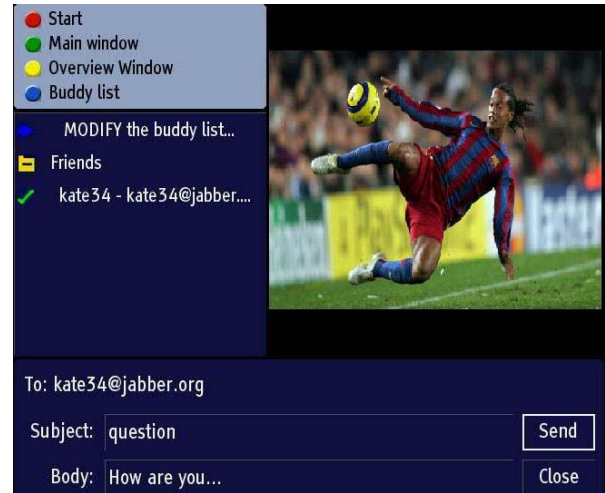


Figure 2: IM4MHP screenshot

As mentioned in section 2 we can divide the functional requirements in global IM requirements and typical iDTV IM requirements. Let's take a look at the global IM requirements IM4MHP fulfills. It is possible to:

- create, maintain and delete an account,
- send and receive real time messages,
- adjust your own presence information,
- view the presence information of your buddies,
- manage the buddy list (add/remove a contact, place your contacts in different groups, ...),
- receive requests to being added on someone's buddy list,
- request the names of the other resources (TV, PDA, ...) and their priorities,
- adjust the priority of the resource you are working on.

Next to these global IM features, IM4MHP also implements some innovative iDTV IM features. You can:

- send TV program recommendations very easily, just select the channel and a program on that channel, add a message like: "very interesting documentary", and send the recommendation to a buddy,
- enable your buddies to see what you are currently watching and look at what they are watching. Of course this is optional: it's very easy to switch it on or off,
- join a chatroom associated with a certain program and expand your social network.

The application provides three window profiles: the maximized, minimized and invisible profile. The maximized window profile takes advantage of the total screen where the minimized profile shows the video in the upper right corner (as shown in Figure 2). The

invisible profile, at last, shows the entire video while IM4MHP still runs in the background. When you push a button you get a reminder that tells you to push the green button to return to the application. With the window profile system it is very easy to create a new profile that fulfills your specific needs, e.g. a simplified IM environment for children.

Each XMPP-client can communicate with IM4MHP regardless of the underlying platform or device.

Two topics were not addressed by IM4MHP. First of all it is not possible to communicate with other IM services because the Smack library does not support gateways. The second unimplemented requirement is a security issue, i.e. the use of encryption. MHP supports Transport Layer Security (TLS) but needs the trusted certificate of the XMPP server to be stored locally. We saw that most of the XMPP servers use certificates from CAcert or StartCom. The presence of their root certificates could be the solution but wasn't implemented at the time of writing.

6. Conclusions & further research

IM can make iDTV a much more social medium. Nevertheless, porting IM from a computer environment to an iDTV environment is much more than just porting the basic IM functionalities. Some very interesting features specific for iDTV can be added, like program recommendations, letting your buddies know what you are watching and taking a look at what they are watching.

After an exhaustive research we came to the conclusion that the best technical way of achieving these purposes right now, is through the XMPP technology. The conclusion of a second part of the research was that the Smack library turned out to be ideal for the usage of XMPP on the MHP platform. In a final stage of the research an extended prototype was built which proved that XMPP fulfilled all the functional requirements.

Further research topics in this area could include:

- Security: the usage of the Belgian electronic identity card for authentication (e.g. secured chatrooms),
- the integration with videoconferencing and VoIP for a complete triple play solution,
- the integration with t-learning and t-health applications,
- making gateways possible with the Smack library in order to achieve interoperability with other IM services.

References

- [1] Abreu, J., Almeida, P. & Branco, V. (2001). *2BeOn – Interactive television supporting interpersonal communication*. Paper presented at the 6th Eurographics Workshop in Multimedia, Manchester, UK
- [2] Adams, D.J. (2002). JEP-0009: Jabber-RPC. *Jabber Enhancement Proposals*, from <http://www.jabber.org/jeps/jep-0009.html>
- [3] Eatmon, R., Miller, M., & Muldowney, T. (2004). *JEP-0096: File Transfer. Jabber Enhancement Proposals*, from <http://www.jabber.org/jeps/jep-0096.html>
- [4] Echomine Muse read on 10/6/'06 <http://open.echomine.org/confluence/display/MUSE/>
- [5] European Telecommunications Standards Institute (ETSI): Digital Video Broadcasting (DVB) (2002). *Multimedia Home Platform (MHP) Specification 1.0.2*. Doc. nr: TS 101812 v1.2.1
- [6] Forno, F., & Saint-Andre, P. (2005). *JEP-0072 SOAP over XMPP. Jabber Enhancement Proposals*, from <http://www.jabber.org/jeps/jep-0072.html>
- [7] Google Talk read on 20/05/'06 <http://www.google.com/talk>
- [8] Internet Engineering Task Force (IETF), Jabber Software Foundation, Saint-Andre, P., Ed. (2004). *Request for Comments: 3920, 3921*.
- [9] Jive Software: SMACK API read on 10/6/'06 <http://www.jivesoftware.org/smack>
- [10] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley & E. Schooler (2002) *SIP: Session Initiation Protocol, RFC 3261*
- [11] M. Chuah (2002) Reality Instant Messenger: The promise of iTV delivered today.
- [12] Quico, C. (2003). Are communication services the killer applications for Interactive TV? *Proceedings of the 1st European Conference on Interactive Television: from Viewers to Actors?* Brighton, UK pp.99-107.
- [13] Shigeoka, I. (2002). *Instant Messaging in Java*. Greenwich, Manning Publications Co.
- [14] S. Leggio (2004) *SIP for Instant Messaging and Presence Leveraging Extensions*
- [15] S. Ludwig, J.Beda, P. Saint-Andre & J. Hildebrand (2006) *Jep-0166: Jingle*



CCNC 2007

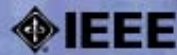
consumer communications
& networking conference

SEAMLESS CONSUMER CONNECTIVITY

HARRAH'S LAS VEGAS, NEVADA

11 - 13 JANUARY 2007

SPONSORED BY:



IEEE COMMUNICATIONS SOCIETY