Instant Music & Messaging

Interconnecting music and messaging

By
Mattias Johansson

Master thesis in applied information technology
Department of Information and Communication Technology
Royal Institute of Technology, Kista, Sweden

March 2009
Abstract

Communication is and has always been important for the human as we are designed by evolution to communicate as a way to survive and reproduce. What many people do not think about is that music and communication are very closely related due to the fact that music is a type of communication. In this thesis we have focused on the field of music and communication to discuss the possibility of combining these to areas to provide better information technology services. More specifically we have focused on discussing the advantages of combining the communication technology of instant messaging with music playback. Our goals are that it will increase the user experience as well as indirectly help the music industry to promote artists and their music as the communicating peers will be able to share information about their music in a more efficient way.
# Table of content

1. Introduction ................................................................................................................. 2  
1.1 Background ............................................................................................................... 2  
1.2 Motivation and purpose........................................................................................... 3  
1.3 Method ..................................................................................................................... 3  

2. Analysis ......................................................................................................................... 4  
2.1 Music ......................................................................................................................... 4  
2.2 Instant Messaging .................................................................................................... 5  
2.3 Instant Music & Messaging....................................................................................... 9  
  2.3.1 Implementation issues ..................................................................................... 12  

3. Proposed application ..................................................................................................... 14  
3.1 Networking .............................................................................................................. 14  
3.2 Internationalization ............................................................................................... 16  
3.3 Customized window graphics ................................................................................. 17  
3.4 Updating .................................................................................................................. 20  
3.5 Music playback ....................................................................................................... 22  
3.6 Instant messaging ................................................................................................. 23  

4. Thesis condense .......................................................................................................... 25  

5. Conclusion and future works ....................................................................................... 27  
5.1 Conclusion ............................................................................................................. 27  
5.2 Future works .......................................................................................................... 28  
  5.2.1 YoumuZic Live Player ..................................................................................... 28  
  5.2.2 Instant Music & Messaging ............................................................................ 30  

6. References ................................................................................................................... 31
List of Abbreviations

C

CORBA. Common Object Request Broker Architecture

I

IDL. Interface Description Language
IETF. Internet Engineering Task Force
IM. Instant Messaging
IMM. Instant Music & Messaging
IRC. Internet Relay Chat

J

JMS. Java Message Service

M

MSNP. Microsoft Notification Protocol

O

OS. Operating System

R

RMI. Remote Method Invocation
RPC. Remote Procedure Call

X

XMPP. Extensible Messaging and Presence Protocol
Acknowledgements

First I would like to thank my supervisor from MMI Sweden, Christian Nicolaisen, for giving me the opportunity to work in this field which inspired me to write this thesis.

I would also like to thank my supervisor from the Royal Institute of Technology, Leif Lindbäck.

I would like to thank my family for the encouragement they have given me throughout the process of my work on this thesis.

Most of all I would like to thank my girlfriend Jiraporn Yindeemoh for supporting me during my work on this thesis, but most of all for pushing me and giving me the encouragement I needed.
1 Introduction

1.1 Background

Communication, the transfer of information such as thoughts or messages by signs, sounds and written text [1], has always been important. The human species is designed by evolution to communicate and thus we use it daily in our work and social life to exchange information and knowledge. The ability to communicate has been important for our ancestors and is also very important for us as a means of survival and reproduction [9]. Communication over large distances was a huge problem before but as communication technology improved the obstacles decreased. Communication today has become tremendously much easier due to the use of tools such as the telegraph, phone and mobile phone. With the introduction of the Internet technology the few remaining obstacles were mitigated. Sending messages across the globe and receiving a reply can be done in a matter of seconds or hundreds of a second. Email is probably the most well known way to communicate over the Internet and while most Internet users own an email address it takes time to have longer conversations. An easier way to communicate and keep a conversation is through instant messaging, a way to keep real-time text conversations over a computer network. Similar to voice conversations over the phone line instant messaging is highly responsive. You can receive a reply instantly, thus the word instant in instant messaging. As it is easier and not even comparable in cost to phone calls instant messaging has become a popular way to communicate around the world.

Music, although not normally consider when speaking of communication, is a way to communicate. Musicians use music as a communication channel where they can share their thoughts and opinions. In combination with the growth of the Internet, musicians have adopted and use it as an important channel to promote and spread their music. Through Internet they can reach listeners all over the world and be heard by people that would otherwise never listen to them. Being able to reach a lot of people and potential fans is important to all artists, especially the less known. Internet brings the listeners to the music, not the music to the listeners as it does through radio and television.

This thesis is part of a project that aims to create a web application where music and communication are the two main focuses. The web application is called Musicbrigade, a
website where artists are provided with tools to promote their music, maintain part of their career and much more. Through the website the artists can spread their music and the listeners can easily find and listen to it. This way the site visitors can discover new music from both established as well as unknown artists. The second part of Musicbrigade is the desktop application which is called YoumuZic Live Player. Through YoumuZic Live Player the users are supposed to be able to access all the music from Musicbrigade through a desktop application similar to Windows Media Player, Winamp or iTunes. Two of the main ideas with the Musicbrigade concept are that the music should be at least as easily available as illegal music and at the same price, i.e. free. In this thesis we focus on the desktop application YoumuZic Live Player.

1.2 Motivation and purpose

Regarding the background we are interested in knowing the potential of applications that combine music playback and instant messaging. We are interested in the benefits that we can get from combining these two functionalities. We will also discuss the future for this kind of applications.

To help us we are going to develop a proposal application that combines these features. This application was assigned to us as a part of the Musicbrigade music community concept and goes by the name YoumuZic Live Player.

1.3 Method

We will develop an application that combines instant messaging with music playback, a so called Instant Music & Messaging (IMM, discussed in chapter 2.3) application. We will also research the topics instant messaging and music playback to find out what the current situation is for applications that combine IM and music playback. We will also try to discern what the future of this type of application is.
2 Analysis

In this section we are going to discuss the topics music and instant messaging as well as introduce and discuss our proposed concept Instant Music and Messaging.

2.1 Music

“Music is everybody’s possession. It’s only publishers who think that people own it.” — John Lennon

Music is considered the highest form of art and culture by many. They imply that music personifies their and other people’s tastes and values [12]. Music is such a multifaceted medium that it contains something for everyone. Music is part of human life and we experience it every day and most of the time we do not even reflect on it. It can be the music played by the bus driver or the music in the stores where we buy our groceries and clothes. We even experience our first encounter with music before we are born. Tia DeNora writes that the first music that reaches the human ear is the heartbeat of the mother in her unborn child’s ears [13].

Nicholas Cook writes that music is music despite the different types of music that exists, that music somehow seems like a natural thing, something that exists apart and is suffused with human values [11]. He defines music as something that is what we make it, and what we make of it, it does not just exist. According to him music is not just something that people listen to, he indicates that music is something that is deeply embedded in human culture and therefore talking about music is generally talking about what music means. When he speaks about music as a part of human culture he means that music is deeply embedded in all cultures around the world, which could be compared to the fact that there are no cultures without a language. Languages and music go hand in hand everywhere.

Music today does not just define who we want to be, it is a significant part of who we are. It helps create our personality and it is part of our personality [11] [12] [13].

To be able to provide the need of music for people it has to be distributed somehow. Music and media distribution can be divided into two categories, streamed and non-streamed
distribution. The former method refers to the delivery method of the media and not to the media itself. The term stream refers to a succession of data elements made available over time. Music and media that you hear on the radio or television network is inherently streamed while in comparison the latter streaming category conforms to media that is distributed in books, on CDs, DVDs etc., i.e. media that is intact.

A mandatory tool to be able to play the distributed media is applications such as media players. They provide functionality to play streamed and non-streamed audio and video, though it depends on the application if both distributions are supported. Most media players provide support for a variation of audio and video formats. Winamp\(^1\), Windows Media Player and VideoLAN Client\(^2\) (VLC) are three popular media players to mention a few.

### 2.2 Instant Messaging

> “Communication is about being effective, not always about being proper.”
> 
> Bo Bennett

Instant Messaging (IM) is the most recent and popular incarnation of near-synchronous communication technology. It is a form of real-time communication based on text between two or more people conveyed over a computer network, such as the Internet. The near-synchronous interaction experience is created when the communication is done over a fast network where transmission times are fractions of a second.

The traditional chat model allows people to gather in a defined shared space, a chat room, where they can interact with each other. IM facilitates this interaction model but conforms to the call model with two participants, i.e. IM provides for two people to meet in a “room” where they can interact with each other. Though the bilateral interaction model is the core model of IM most modern IM systems provide the possibility for conference interactions where additional participants can be invited to an existing chat session.

\(^1\) [http://www.winamp.com](http://www.winamp.com)

\(^2\) [http://www.videolan.org/vlc](http://www.videolan.org/vlc)
IM is a descendent from the antique Unix utilities *talk* and *write*, but first found a wide audience in the mid and late 1990’s with the modern IM applications that started the popular IM culture we have today (ICQ, MSN etc.). The use of IM among teenagers is continuously increasing and indicates that the use of synchronous or near-synchronous messaging and presence awareness has a place in teen life despite competition from other communication media, such as telephones and email [7].

IM is used to accomplish many different tasks, e.g. keeping in touch with near and distant family and friends, coordinating meetings and work assignments in or out of the office. Regardless of the reason for IM Grinter & Palen writes that most of them are about maintaining relationships with someone [7].

Most IM systems provide a “buddy list”, a list of people (friends, family, co-workers etc.) that are available through the IM system. In the buddy list these systems usually provide some sort of presence awareness information. This information spans from a simple status indication (message, icon, etc.) that the users are allowing to change to indicate their presence (e.g. “Away”, “Busy”, “Out to lunch”, “In a meeting” etc.) while other systems provide additional information such as how long a person has been idle and more advanced information. Tang and Begole discuss three research prototypes that are potential solutions for better presence awareness [5]. One of these prototypes, *Rhythm Awareness*, analyze awareness information collected over time to calculate and predict a user’s awareness pattern to provide information about when a user is more or less likely to be available. *Lilsys*, another prototype relies on external sources of information to predict the user availability. As an example, in the article they mention that when some office workers are preoccupied they tend to close the door to their rooms, which would indicate unavailability. Lilsys is in this case connected to a sensor that can sense when the door is closed and indicate to the IM application that the user is most likely occupied and will not return any messages. Though this is a bold experimental approach to improve today’s “lacking” presence awareness it will most likely not be a success. Why would anyone want to buy additional equipment just to provide a more precise awareness profile in IM systems? If the equipment is cheap a small market might exist for it. A more promising solution lies in the line of Rhythm Awareness, providing additional information to the awareness service from a user’s historical pattern of presence.
Several articles mention IM as a common communication medium used not only for private purposes, but also as used as an important tool to be able to work more efficiently in the workplace [4] [5]. The awareness features of modern IM are used to get a notion of the presence of a friend or co-worker, to see if it is an appropriate time to start a conversation or if the initiator should wait. IM is also used as the first step in the process of initiating a communication channel as it is less intrusive then other communication media such as telephone or face to face conversations. Nardi et al [4] explains that during an IM observation project the participants would often begin an IM interaction and eventually switch to another communication medium such as telephone, face to face or even email. They call this phenomenon *media switching*.

It is not only IT-concentrated businesses from big companies such as IBM and Accenture Ltd. to small local companies that use IM to communicate. The US military use it [14]. According to the article the US Navy has been using IM as a communication tool since the terrorist attacks on the World Trade Center on September 11th 2001. The navy’s top admirals are using IM to send messages to their staff and each other. Even several hundreds of the Navy’s ships are connected to an IM system which allows them to communicate with each other. The article also mentions the use of IM at racetracks so that the pit crew members can communicate with each other.

IM allows the recipients more control over the responding then telephone and face to face conversations due to the ease of monitoring, delayed responding and possibility of denial of presence. This allows the interaction to be equally negotiated between initiator and recipient instead of being solely on the convenience of the initiator. This is one of the reasons why IM is used to negotiate the availability for a phone or face to face conversation — media switching.

Compared to other types of communication IM allows for an easy track of conversational history which provides an easy way to trace back conversations. This is a major advantage compared to phone and video conversations that does not supply this functionality at the same level of ease. Most IM applications enables chat conversation logging by default which makes it easy for the users who would like to trace back an old conversation.
An important and interesting perspective of IM is that it is often used in multitasking sessions and that it is easy to integrate with other activities. Users tend to multitask while keeping IM interactions.

As IM is less interruptive than other communication mediums a person can conduct an IM interaction at the same time as talking to someone on the phone or having a face to face conversation. If you have ever tried to talk to someone on the phone while keeping a face to face conversation you know how hard that is. IM is often monitored at the same time as other conversations take place. The easy monitoring activity allows people to prioritize incoming message and decide if they are important or can be left for later. This type of monitoring is not as easy with other types of communication [4].

With regard to the non-interruptive nature of IM it is ideal for quick question and answer conversations. This type of conversation is frequently used in the workplace which in combination with a general idea of co-workers availability and participation in social conversations support the work conduct and reinforces the ties between people [7].

An interesting aspect of IM is that it is not only used for short conversations but also for longer interactions. These interactions can take place over a few hours or days due to the freedom the recipient have of choosing when to respond. While short interactions ideally consist of ask-response contexts these long intermittent interactions contain several topics which come from the fact that long conversations are similar to face to face and phone interaction where topics tend to change rather often. These intermittent interactions are also a result of the availability of keeping a conversation open (in an IM window) and just keep it there until the user judge it a fit time to respond. The participants are able to do other things in between the interactions. One of the participants of Nardi et al’s observation project [4] commented on this and contrasting it to a normal phone conversation by stating that in phone conversations the time is usually limited but in IM conversations you can send a message whenever you find the time for it or when anything comes up, you are not restricted by time.

One of the big advantages IM have over emailing is the rapid exchange of information. IM provides a tremendously more rapid information exchange but without the overhead of face to face conversations [4]. Nardi et al found that some of their research project participants commented that IM allowed them to carry out efficient exchanges of information because
they did not have to follow certain formalities of address associated with phone or email conversations. The name of the initiator indicates to the recipient who sent the message etc. They also observed the participants use of IM and describes that it was mainly used for short questions and clarifications, coordination and scheduling, arranging improvised meetings and keeping in touch with family and friends. That is one of the flexibilities of IM, the ability to use the technology to concurrently have several different types of information exchanges open. Another powerful feature of IM is its expressiveness. It allows for communication about work, social bantering, jokes and intimate communication with family and friends without having to change “workspace”. Nardi et al also writes that IM is an interesting technology and that thus its light weight it succeeds in providing enough contexts to make a variation of exchanges living, pleasant as well as capable of conveying humor and emotional nuances.

They also mention an interesting and promising area of usage for IM as negotiator for availability. By integrating IM in telephones and using it for availability negotiation the high rate of failed phone call initiations can be mitigated.

Large numbers of IM applications exists with varying level of popularity. Three of the more prominent applications are Windows Live Messenger (also known as just MSN), ICQ and Yahoo! Messenger. As an example MSN had more then 294 million active users worldwide as of November 2007 [3] and ICQ has 180 million as of March 2009 [17].

### 2.3 Instant Music & Messaging

"Music is the social act of communication among people, a gesture of friendship, the strongest there is."  

*Malcolm Arnold*

As we have already discussed, music is a big part of human culture. This combined the fact that music is a type of communication indicates that music is part of every person’s life in varying extent. The large quantities of music genres most likely contain music that is appealing to everyone, which lead to the conclusion that most likely everyone listen to music on some occasions. With this in consideration why not include music, both by itself and as a communication catalyst, in the creation of more efficient information technology solutions.
In our world where electronic communication is a commonality we would like to propose a merge between the electronic communication medium of instant messaging with the ever popular music media. With this merge we want to focus on the possibility to make each medium increase the final combined experience. Instant messaging helps people discuss and promote music and music as a part of every person’s life creates new subjects to discuss as well as makes people find similar interests in other people, even though they have nothing else in common.

![Figure 1 – Illustration of the Music and Instant Messaging contribution model](image)

Laughey D. [12] writes that intensive music media users tend to listen to recorded music in favor of music on radio and television. This indicates that these music listeners are more inclined to listen to the music of their own choosing instead of having to endure though less favored music that they did not choose by themselves. He also writes that some people consume a lot of music while surfing the Internet which would endorse the need for an increase of music oriented communication services.

Most people relate or personalize themselves with some kind of music, whether it is classical, house or heavy metal, there is always something for every taste, which means that there is always something to talk about. Due to the huge spectrum of musical compositions worldwide it is impossible for any one person to keep track of all the music that is produced. To be able to select the parts of it that is most appeal is very harder since there is always something new that someone else knows about. If we had a medium that allowed us to more easily promote this knowledge, news about new and interesting music would travel even faster. It would also create new bonds between people because of the interest in music.

We call our solution Instant Music and Messaging (IMM). IMM applications are most likely considered Reality Instant Messaging applications. Reality IM is defined as instant messaging that automatically incorporates relevant events into the IM interaction based on a current live
stream (i.e. a sports game or a live concert) rather then requiring the participants to introduce the topic and explain all the details. As IMM could combine IM with live music streaming (live concerts, live radio, recorded music and much more), the Reality IM provides a matching concept. As described in [6], awareness is a very important part of IM and provides the online presence in Reality IM with activity and interest.

Schiano et al writes that teenagers use IM and listen to music simultaneously which indicates that IMM might have a good impact on the teenage population if the IMM systems are appealing and can compete, or integrate, with existing solutions [8].

Grinter and Palen [7] mentions an application called Aimster that combined IM and music sharing. Apparently Aimster eventually changed name to Madster due to complaints from AOL, owner of the AIM service, and seems to have been brought out of service as of June 2008. In the article Grinter and Palen also mentions that people who share music online also tend to like to talk about music with potential recipients which would indicate that music and communication are matching technologies.

When we get interested in a news story we tend to exchange the information. Since music is one of those personal things that give birth to deep emotions discovery of music triggers the feeling of wanting to share the personal feeling of the discovery with others. The same feeling as with interesting news stories, but deeper. This is due to that people develop strong attachments to songs according to Hu [10]. In combination with the expressional powers of IM this would further improve the IM platform by providing something that many people share, the anxiety to share with others the feelings for and experiences from music.

Many people use Internet as a gateway to access music, weather it is to find new music, review music that is on the “to buy” list or download music.

Plans have been made to move music and messaging closer to each other. According to Hu [10] Yahoo had plans on letting people share and interact with other peoples digital playlists through their IM services back in 2004. Microsoft also had plans in integrating their MSN Music store into MSN Messenger. Executives from Microsoft said that they hoped MSN users would be able to listen to music at the same time as their friends and buy the tracks [15]. As far as we know the music integration in Windows Live Messenger is only that the users can
select to display which music they are currently playing in Windows Media Player to their friends.

### 2.3.1 Implementation issues

#### 2.3.1.1 Established IM and media applications

Users of music and instant messaging already use their selected applications for music playback and communication, i.e. media players and instant messaging applications. How do we make these users start using new IMM applications? The first obvious proposal would be to provide a new product that provides more functionality and that is very appealing to the users. Another proposal is to provide the same functionality but in a nicer package. But is this really a good way or are there any better solutions? Another approach would be to wrap the new functionality around existing solutions, i.e. include existing communication systems such as MSN or Yahoo! into an IMM solution. This of course means that the existing services need to support the integration into other systems. An example is services such as Trillian³, Miranda⁴ and Pidgin⁵ which provide multi-IM solutions where several instant messaging systems are combined in one application.

#### 2.3.1.2 Seamless cross-system communication

Even though these products provide a very attractive alternative to the original products most IM systems do not provide seamless cross-system communication. As an example, MSN Messenger does not communicate with ICQ and neither MSN nor ICQ communicate with AIM. Some improvements have been made as MSN and Yahoo! Messenger users can communicate with each other, but this is due to a co-operation between Microsoft and Yahoo!. We need the different communication systems to allow for a seamless cross-system communication to provide a solution that does not force a bigger system migration for this to work, though that is probably the only solution. There are a few systems that provide this

---

³ [http://www.trillian.cc](http://www.trillian.cc)
⁴ [http://www.miranda-im.org](http://www.miranda-im.org)
⁵ [http://www.pidgin.im](http://www.pidgin.im)
traversing behavior (Jabber/XMPP\textsuperscript{6} and Google Talk\textsuperscript{7}, which is based on XMPP) but it is not as widely spread as we would want.
3 Proposed application

In this section we will describe our proposed application and explain how we have solved some of the issues we encountered and implemented some of our solutions. We will discuss the design we have made and the choices we made prior to and during the development phase.

Our proposed application is developed as part of the music services at Musicbrigade and is called YoumuZic Live Player (YMLP). The YMLP application is tightly coupled with the Musicbrigade website, they share database. Any account created and used from the website can be accessed and used from the YMLP application and vice versa. Since the same database is used on both platforms the information from the website is also accessible through the application. One of the advantages of this is that users that log in with the application show up as logged in on the website. This also allows the users to directly access their accounts on the website without having to log in again, i.e. the users can navigate directly from the application to specific locations on the website through links in the application.

The player’s main functionality is to play the media from the website, but it also contains instant messaging features to allow the users to chat with their friends on Musicbrigade/YMLP. This functionality is only available to users that use the application, though all friends will show up in the friends list of the chat module. As our application is in an initial (prototype) state the functionality has not yet been tested in a real situation with a realistic amount of concurrent users and will therefore contain several flaws. We will take care of them at a later stage in the development phase, during alpha and beta testing.

3.1 Networking

The first step in our development process was to decide how the application should communicate with our application server, the database and the website. As this is an important part of our application design we knew we would need a solid and reliable solution. Our first step was to find different solutions that provide easy, productive and reliable development, and good communication performance. Among the solutions we found we estimated sockets, RMI and CORBA to be the most suitable. Other solutions are connections
over HTTP (e.g. Web Services) and Java Message Service (JMS), but our opinion was that they were not suitable for our application and design.

We knew beforehand that RMI and CORBA are similar technologies that utilize remote method calls. CORBA is the language independent version and RMI is the Java specific implementation. Due to that RMI is a pure java implementation we decided in favor of it rather then having to learn a new language, or more specifically a specification language, IDL, a language-neutral language to describe software interfaces. It is used to specify the interface in CORBA. We were also concerned what communication overhead in CORBA.

Now that CORBA was excluded from our list we still needed to decide weather to use sockets or RMI. Since sockets are the most basic communication channel and developers are able to specify exactly what is send, this is very good from a performance perspective because we could specify exactly what we wanted to send. We realized that if we were to use sockets it would require a lot of time and effort just to manage the sending and receiving of data. If we were to use sockets we would also have to deal with packing and unpacking the data that we send back and forth, maybe even create own protocol, but a protocol would take a long time to develop which would lead to shorter time to develop the actual application. With a bigger team we might have been able to successfully execute this but not with our current team. A custom protocol might of course be an interesting approach in the future. We will discuss this further in chapter 5.2. Our decision fell on RMI.

The choice of RMI partly originated due to the convenience it provides by providing methods calls from client to server and server to client as if they were conducted locally. Another
important reason is that we also found that RMI does not lack the drawbacks compared to socket connections as was previously present. RMI can even provide better performance than sockets in some situations [16].

### 3.2 Internationalization

To be able to provide the application to a wider audience we want to be able to provide the application with support for multiple languages. We also want the users to be able to change the language while the application is still running. The users should not be forced to restart the application just because they want to switch language. Unfortunately Java does not have support for dynamic languages in the standard library. More specifically, AWT and Swing, the packages used to create user interfaces does not support dynamic language changes. In later versions of Java support for this would be a great contribution. We solved this in our application by creating our own library with components that support changing of languages dynamically.

![UML diagram of the language manager library](https://via.placeholder.com/150)

**Figure 3 - UML diagram of the language manager library**

The main class of our library is the `LanguageManager` which notifies all components (LButton, LLabel, LMenu etc.) when the language changes. The language manager object
contains references to each component through the interface that manages language change events, LanguageChangeListener. When the language changes the manager notifies all components through the languageChanged method. When languageChanged is called in each component the component will retrieve the new value from the LanguageManager and update the component with it.

3.3 Customized window graphics

In many modern and popular applications such as Winamp (Figure 4) and MSN (Figure 5) the GUI separates them from other applications because they are providing customized interfaces that distinguish them. Most applications use the same visual styles as the operating system (OS) (e.g. the Windows Calculator in Figure 6). As our application is modern and the aims are to make it popular it was an obvious choice that it would use our own customized interface. We want our application to give the users an impression of modernity as the developers behind MSN and Winamp have been able to accomplish. Younger people are more likely to be drawn to a new application if the GUI is appealing to them. After we had made this decision we had to investigate further how to realize this. The only limitation we could see at that time was that we had to stay within the limits of the Java programming language. As Java is an OS independent language it does not support all native features that other languages such as C and C++ do. After doing some research and prototyping we discovered that Java does not support the change of the windows frame and title bar. The reason for this is that it is handled by the OS and not by the programming language, the language just use the components available from the OS. Now that we knew we could not replace the default frame and title bar we needed to investigate further how we would continue. As part of our research we discovered that IBM had been working with a solution for this around 2004 [2], and found
their solution to be satisfactory. In the article they discuss the architecture they call *IFrame* which allows custom colors, borders, components etc. to be used to decorate Java application windows. We started out by analyzing how they had solved this issue and found that their solution was actually not as satisfactory as we had expected due to that it was not as responsive to window events as we expected. We already knew it would be hard if not impossible to achieve the same responsiveness as native window, but we wanted the responsiveness to be satisfactory enough not to cause irritation from the users.

![Figure 5 - GUI of Windows Live Messenger](https://via.placeholder.com/150)

The first thing we needed to do in order to create our desired customized windows was to remove the window frame and title bar, which as we explained could not be replaced in Java. By calling the `setUndecorated(true)` method in the `Dialog` and `Frame` objects we can remove both the frame and the title bar. When the frame and title bars are gone we need to replace them with our customized components. To replace the window frame we decided to create a frame object that contains a title bar and a content pane object. The title bar contains the same attributes as a normal title bar and the content pane is surrounded by a border. The content of the content pane object is the content of the window. The top structure of our window framework is illustrated in Figure 7.
To be able to give our window component the same functionality as a standard window component we needed to add behaviors for moving and resizing the window. We knew that if we would omit this behavior, even the slightest, it would be catastrophic as the users would not be able to do the basic things they can do with other windows. To transfer this behavior to our component we were forced to utilize both mouse and window listeners to handle different events and act accordingly.

Even though our solution is working according to our expectations we know that replacing the native window handling with custom window handling it is not as efficient. We also know that it might not be able to achieve equivalent performance for various reasons, e.g. that the code to handle these events is not optimized and contains flaws.
3.4 Updating

As with most applications, our application will eventually also need to be updated. As we considered this a vital part of our application design we needed to find a good solution on how to make updating a non-intrusive necessity. We want the updating process to have as little impact on the usage of the application as possible, i.e. we don’t want to users to be forced to download and install updates. Our intentions are that the updating should be done automatically without any interaction from the users. This way we can make sure that everyone that uses our application has the latest version without them ever having to make an active choice. We quickly discovered that Java does not have any updating functionality. Due to this we needed to find a third party updating library or create our own. When we searched for existing solutions we could only find a meager collection of updating libraries and none of them appeared to fit our projects goals. We judged our time was not enough to delve deeply into these solutions to find out if they supported exactly what we were looking for. This forced us to create our own library.
One of the solutions we found was similar to what we were looking for and we decided that our solution could make benefit being built similarly as this. The library we found is called JUpdater\(^8\) and it is based on a Java-to-PHP client-server architecture. We decided our solution would benefit from the same architecture due to the fact that it uses XML-RPC for communication, a Remote Procedure Call (RPC) protocol using XML over HTTP. This way we could continue to keep our client pure Java and simply write the server application in PHP. The advantage of having the server in PHP is that it can be moved to any other web server supporting PHP without having to consider which environment it runs on. The only requirement would be that it needs to be published on a web server. Another advantage is that we can change the language on the server side without forcing a change on the client, i.e. we could decide to rewrite the server in Java or ASP.NET without having to change anything on the client.

![Figure 9 - Workflow structure of the updating functionality](http://www.lokorin.org/jupdater)

The workflow of our updating library is relatively simple (illustrated in Figure 9). When the application starts the client makes a XML-RPC call to the server sending the name of the application as well as the version number as parameters. The application name is registered in the server and identifies our application. The server compares the received version number for

\(^8\) [http://www.lokorin.org/jupdater](http://www.lokorin.org/jupdater)
the specified application against the latest version available in the server to check if a newer version of the application is available. If a newer version is available the download path to that version’s installation file is returned, otherwise the server will return false to indicate that no newer version is available. If the server returns the download path to the client the installation file is immediately downloaded to the client and automatically started.

We designed our updating solution to make it is easy to add additional applications. The only identifier what indicates which application we are interested in is the application name.

As the latest installation file has to be downloaded every time a new version is available this will quickly waste more bandwidth then is actually necessary for the update due to that during an application update only contains parts of it is affected by the updates. How to decrease the bandwidth usage, download and update time is discussed in the chapter 5.2.

3.5 Music playback

As the concept of Musicbrigade is to provide artists with tools to help them promote their careers and distribute their music we needed to provide YMLP with the functionality to play that music.

The first decision we needed to make was in which format to distribute the music. The first format that was written on the list was MP3, but due to the license of the audio format anyone who distributes an application with a MP3 encoder must pay a license fee. The license fee applies to all non-free applications and has to be paid for each copy of the application that is distributed. This is very unfortunate for us and forced us to consider supporting another audio format to avoid paying this license fee. MP3 needed to be substituted with a free option and we found Ogg Vorbis which is a free and open source audio format without any licenses attached to it and a collection of free libraries allowed us to continue working with it.

We also needed to add support for music video playback in our application. The web solution used Flash and Flash Video (FLV). This led to some problems for us because we could not find any stable support for Flash and Flash Video in Java and realized we needed to make an adjustment here as well. As Ogg also contains a video format, Ogg Theora, we decided this would be a suitable replacement as Theora is also an open file format that is free to distribute.
Our searching for a Theora decoder did not result in much. We found one solution but the problem was once again the license. The application was licensed under the GNU General Public License (GPL) which states that any project that uses a GPL licensed library or code need to conform to that license as well and it was not acceptable for us. Apart from this solution we couldn’t find anything else. The only solution we could think of in this situation was to create our own Theora decoder for Java based on the Theora format specification. But writing our own decoder would take more time then we had at our disposal and we had to take the decision not to include video playback in YMLP for the moment and put it on the future to-do list.

3.6 Instant messaging

An important part and one of the main features of the YoumuZic Live Player is the messaging possibilities. Communication is an important part of the Musicbrigade community and it was decided to be included in YMLP as well. It was early on in the project decided that the chat feature should be simple in the beginning yet contain enough features to allow the users to communicate effortlessly with each other. A decision was made that the window dedicated to the chat functionality would contain all the chat conversations instead of opening a new window for each conversation, much like IRC. The main reason is that we don’t want to populate the user’s screen with too many windows if he or she is keeping several simultaneous chat conversations. Another reason is that the user’s list of friends is available on the window’s left hand side at all times, regardless of whom the user is chatting with. A simple click on the friends name will open a chat session with that person.

As our application is in an early stage we have not had time to implement any advanced features common in instant messaging applications. We have only implemented a small amount of basic features. We plan on increasing the functionality as the user base is growing and the demands from the users start coming in. Most users will probably miss some of the basic features that are available in other messaging application and we are planning on implementing them later on.

The reason we haven’t implemented more features is that the focus of our application has been on making it possible to access the music from the website, which is and will always be
the main priority of YMLP. We will discuss more about the future of the messaging feature in our application in chapter 5.2.
4 Thesis condense

The human being was created and has since then constantly been adjusted to be able to communicate as an important ingredient in our fight for survival and mating. We are apt to adjust to changes and one example is that large distances has always made it difficult to communicate but with the invention of the telegraph, telephone and eventually the Internet this problem has been mitigated to only be the slightest problematic. Video conversation, as an improvement of the previously mentioned technologies, is used to mitigate it even further due to that users are able to communicate body language and expressions as well as text, speech and sounds.

Music is a way to communicate thoughts, feelings and much more to other people and many people consider it as the highest form of art. Music is embedded in human culture all around the world which makes it part of everyone’s lives. Music is implemented in so many things around us that we don’t even notice it. Music is played in the TV during commercials, TV-shows and movies. It is played in the store where we buy our clothes and groceries as well as many more places.

As both communication and music are big parts of our every day life we have propose a merge between these two mediums to create music oriented instant messaging applications. The reason for us to choose to integrate music with instant messaging is because instant messaging is an immensely popular way of communicating these days, and the popularity is continuously increasing. In our proposal we see music as a catalyst and common denominator for initiating instant messaging interaction and instant messaging as an easy and very important way to discuss and promote music. We describe our solution as a Reality Instant Messaging application. The concept of Reality IM is defined as instant messaging that is infused with subjects from a live media stream. The live streams can for example be a live concert, a radio program or a live sports event. This way the participants have something in common to talk about and it can also help people to initiate contact with previously unknown people that share the same interest, music.

By proposing this combined solution our assumptions are that it can help promote music and friendship. We expect them to help each other in some aspects, i.e. instant messaging will
help promote music which will increase the amount of people who communicate though instant messaging, which leads to more talk about music etc.

We found that our proposed solution must overcome some obstacles to compete with the already existing instant messaging and music applications. When discussing these issues we mainly focus on the area of IM because it would be easier in our point of view to integrate music playback with an IM application then the other way around. The demands on a new application are very high due to that it has to compete with applications that have millions of active users which are used to the high quality features available to them and that they might not be so keen to start using an additional application or completely switch. The best solution would be to integrate our concept with the existing applications but there are some major issues involved. The first problem is that each of these services does not support cross-system communication which would prevent people on different systems to communicate with each other. A user on MSN can not talk to a user on ICQ for example. There are improvements in this area but it is still unsatisfactory. Users on Windows’ Live Messenger and Yahoo! Messenger can communicate with each other as a result of that Microsoft and Yahoo! are cooperating. We found a solution that aims to mitigate the problems of closed proprietary instant messaging services. The technology is called Jabber and is based on the XMPP, which is developed by Jabber and is part of IETF. It aims to provide a standardized way of IM communication that allows different services, based on Jabber, to communicate over system borders as long as both systems are using the same protocol.
5 Conclusion and future works

5.1 Conclusion

First of all we see an increasing trend in implementing instant messaging into new and existing applications to increase the usability and efficiency. As an example we saw that large corporations such as IBM and Accenture as well as the US Navy use IM to allow for the staff to communicate more efficiently. In the case of the US Navy it allows their staff to communicate with each other as well as the possibility for several of their ships to communicate with each other. This would indicate that the usage of instant messaging is continuously increasing. IM is implemented in many different ways to create more efficient communication solutions.

With regards to the previous intentions from Microsoft and Yahoo to implement music features into their messaging services and the increasing use of instant messaging services we find that IMM applications is part of the future. Music is and will always be a subject of discussion and we see that instant messaging is one of the most important communication channels for it. Music is one of those things that people tend to want to talk about, whether it is when they find something interesting and want to share the discovery or they just want to discuss music in general.

It is impossible to keep track of all the music that is produced around the world, not even within one genre, and instant messaging is an important channel to spread the information. Instant messaging helps people spread the word to people who would otherwise never have heard of it. We see this as one of the most important reason why music oriented and IMM concentrated communication applications is the future.

To produce and maintain successful IMM services it is recommended for the product developers to consider and address the issues of implementation an IMM service that we discussed. By this we do not imply that all of these issues need to be solved, rather that the developers are aware of them and make up a plan for how to address them. With so many different products already available on the market it is hard to reach a big audience unless you have a protruding or very inviting product. That is the main issue the producers need to
consider. A large user base is required to be able to create and maintain a profitable product, thus these issues need to be addressed.

We believe that the number of music oriented instant messaging applications is going to continuously increase in the future. We also believe the majority of this concept’s audience to be teenagers and young adults.

5.2 Future works

Due to the time limitations of our work we were not able to implement all the functionality that we would have liked to. There is also functionality that we regard as possible implementation in the future. We will discuss the future works for our proposed application as well as the IMM concept.

5.2.1 YoumuZic Live Player

- Communication protocol

A way to increase the communication performance might be to provide a customized communication protocol for the YoumuZic Live Player much as the MSNP protocol use in Windows Live Messenger.

- Music playback

In the future it might be convenient to enable the possibility to view the music videos in full screen mode. This will require not only the development of a full screen feature but also most likely high quality videos to be streamed from the streaming servers.

It might also be of interest to add the possibility to play the music that the users have on the local machines. This would probably be considered a great addition to the application and make it an even more central part of the user’s daily computer usage. Why use more then one application when you can play everything from one?
• **Instant messaging**

An important future improvement for the YoumuZic Live player is to increase the instant messaging functionality. The aim would be to implement the most popular features from the most popular instant messaging applications as well as implement some Musicbrigade/YoumuZic Live Player specific features that would increase the instant messaging experience.

An improvement of the instant messaging part of the application might be to implement an interface that allows for the users using the application to chat with the users on web site. Not only send sporadic messages that show up in the desktop application but provide a full featured chat feature that allows for the users on the web to fully communicate with the users in the YoumuZic Live Player.

An interesting development is to improve the instant messaging feature and provide the possibility to talk to more than one friend at the same time. Not like the conference feature in Windows Live Messenger, but more like Internet Relay Chat (IRC). This might open up the possibility for chat sessions with artists or even press conference.

• **Updating**

A great improvement would be to provide a much more sophisticated updating functionality that only updates the files that has been added, updated or removed since the version the user is using. The new implementation would support downloading of separate files which would decrease bandwidth usage and lower the update time as well as the cost of server bandwidth for the provider.

• **Graphical User Interface**

Another improvement is to enhance the structure and performance of the applications user interface and make it more responsive and interactive with the users. Adding drag-and-drop features in the player to improve the interactivity in the playlists, to rearrange the order of the music, as well as update profile images etc.
A big improvement to the GUI would be to implement the possibility for transparency. Though this is not 100% supported in the Java environment it would provide the possibility to create extremely customized windows.

- **Operating system independence**

To increase the user base it is important to expand the amount of platforms that the service is offered on to include other platforms then the Windows platform which is the only currently supported platform. We advocate for an expanded platform base to include at least the Mac and Linux platforms.

### 5.2.2 Instant Music & Messaging

- **Sharing music**

An interesting improvement to the IMM application is to develop functionality that allows users to share music with other people they are talking to in instant messaging conversations. By this we do not imply that the users should be able to send the music to their friend but “play” the music to them. This way a user can let a friend listen to a song that he or she is talking about and propose him or her to it listen to without having to send the song or let your friend find it somewhere else.

One way would be to send a link to the song to another user or send a proposal that would allow another use to add the song directly to his or her playlist.

- **Playlists sharing**

A nice implementation to IMM applications would be a way to allow the users the share their playlists with each other as well as display which music they are playing at the moment. The feature to display the currently playing track already exists in different software applications and helps promote which kind of music a user is listening to.
6 References


