A Wearable Life Multimedia Streaming System

Georg J. Schneider
Computer Science, University of Applied Sciences Trier
PO Box 1826, D-54208 Trier, Germany

and

Thorsten Roth
Computer Science, University of Applied Sciences Trier
PO Box 1826, D-54208 Trier, Germany

ABSTRACT

The paper describes the design and the implementation of a system that enables life video streaming in the internet using the Java Media Framework. The sender or presenter walks with a portable computer over a fair and transmits life video and audio data over a WLAN to a server, which is connected to the internet. The users that want to see the video can connect to this server using a web interface. Furthermore a user can interact with the sender either by chatting with him or by requesting the permission to talk. If the permission is granted, the user can communicate with the sender over a voice channel.

Keywords: Multimedia streaming applications and systems, Multimedia streaming in wireless systems

1. INTRODUCTION

Streaming of Multimedia Content trough the internet is a technique, which has become very popular during the last years. The better Network infrastructure allowing high data rates also for home users (e.g. using DSL connections or cable modems) as well as the improvement of the video and audio codecs contribute to that fact. Information like tutorials, interviews or video clips can be viewed on a computer just by clicking on the appropriate link on a web page. More and more radio stations stream their audio program in the internet; sometimes they even transmit live video streams from music concerts. The users have to install special components e.g. Real Player, Windows Media Player or the Java Media Framework (JMF) to render the streams on their computers. These components are often freely available but depending on the system, the support for different operating systems is limited.

These kinds of applications usually don't consider any way of interaction between the sender and receiver. Applications that are targeted to the audio/video (AV) interaction between different parties are called video conferencing systems. These systems provide a voice and video channel between the users (often combined with application sharing components). However they are usually targeted to small groups of people that want to communicate and interact. Furthermore specialised video conferencing software has to be installed on the computers of the participants. Even so hard- or software components to distribute video and audio files for multi party connections, so called MCUs (Multi Point Control Units) must be installed or rented.

For many application areas a combination of the advantages of both systems are of a great benefit. Consider a scenario were one user sends audio and video data from a life event to a large group of users, which could in turn interact with him in real time. E.g. a user visits a fair with a portable transmission unit, which is streamed in the internet and the viewers can tell him to move closer to certain objects or to carry on to another site.

The advantage of such a system is that almost anybody could transmit life video stream to a large audience, like we know it from television today using standard computer equipment with the benefit of providing immediate feedback to the sender.

The requirements on such a system are consequently that it is built on standard components. It must support a wide range of operating systems and the transmission in both directions has to be near real time. Furthermore it must be adaptable in a way that it can be integrated in a specialised user interface for the specific application.

2. RELATED WORK

As already mentioned in the introduction there are different types of systems that target the interaction with one or many parties using video and audio.

Looking at video conferencing software, there are not many freely available implementations that meet the requirements of the targeted system. The so called MBone tools (see [1]) require that the user is connected to a part of the internet that supports multicasting. Especially in the area of E-learning, which has similar requirements than our domain, these tools are frequently used [2].

Commercially available tools like Microsoft Netmeeting [3] can be nicely customized since it provides a COM API yet it is not available for many operating systems.

Different streaming infrastructures can be used for audio/video content over the internet. Companies like Apple [4], Microsoft [5] or Real [6] provide a fully functional infrastructure that is easily to deploy. These infrastructures compress the data very efficiently mostly using proprietary formats. As a consequence the rendering of the streams requires an appropriate viewer/player, usually available only from these companies.

Since only few players are available on all operating systems, only few streaming infrastructures are suitable for a platform independent implementation.
Furthermore the players usually cannot be completely adapted to the look and feel of a custom user interface. Finally there is a delay of about ten seconds on almost all platforms we have tested so far, which cannot be eliminated neither trough configuration nor by guaranteeing higher data rates on the network.
The last point is the biggest drawback if the system shall provide a near real time feedback channel to the point where the event takes place. The system would behave unpredictable from the point of view of the users, if they will refer to an object they just saw on the display but the presenter would be already at the next site and relates the utterance to his current context.
The Java Media Framework on the other hand is a platform independent library for capturing, sending and rendering multimedia content, based on the Java programming language [7]. JMF supports different audio (e.g. MPEG, GSM) and video formats (e.g. MJPEG, H.263) and network protocols, especially RTP (for further details see [8]. Since the Java programming language is (freely) available on most operating systems it is a popular middleware for multimedia systems (e.g. [9]).
The "Wearable Life Multimedia Streaming System" uses JMF as well. The following chapter will describe the design of the system in more detail.

3. THE WEARABLE LIFE MULTIMEDIA STREAMING SYSTEM

In the following the background of the system and the realization will be described.

The state garden show domain

The "Wearable Life Multimedia Streaming System" is targeted in order to enable life video transmission form the state garden show, which takes place between April and October 2004 in the city of Trier, Germany. The goal is to open this event to visitors, who cannot visit the show personally, either because they life far away from Trier or because they are disabled. For this reason, the users must be able to access live video and audio streams very easily. They shall be able to interact with the sender of the video, i.e. the presenter to tell him to move closer to certain objects or to remain longer or shorter on a specific site.
There are two different ways to realize the interaction with the presenter.

- One way is to use a voice channel as in the video conferencing systems.

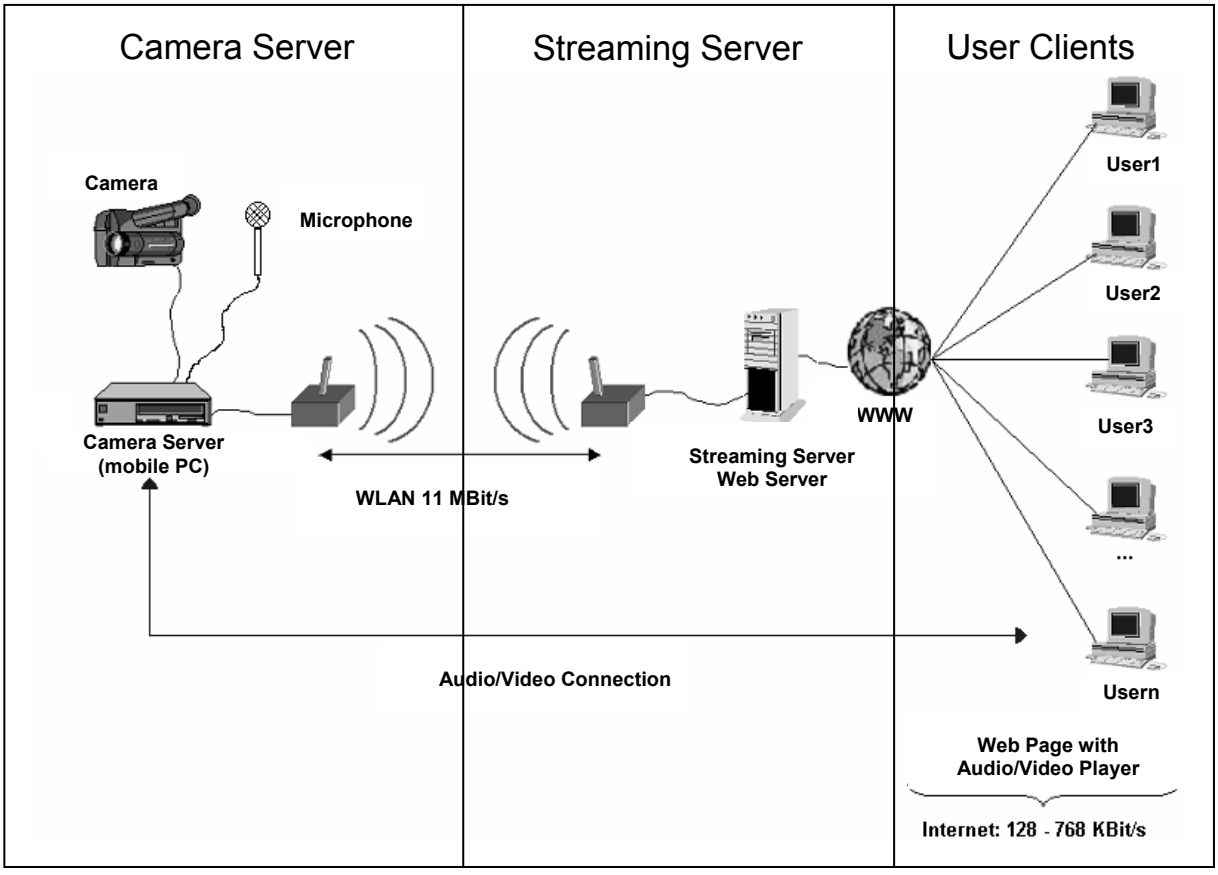

Figure 1: Three tired Architecture of the Wearable Life Multimedia Streaming System

- The other way is to use a chat interface, which is especially suitable for a large audience, where also interaction between the listeners takes place.

Since the state garden show area is relatively big (several miles), a means has to be provided so that the users are able to find out, where the video transmission is currently located in this area. This can be achieved through incorporating a map in the user interface, where the position of the presenter is displayed.

Realization of the Wearable Life Multimedia Streaming System

As mentioned above the JMF serves to provide the core functionality in regard to audio and video coding, decoding and transmission.
The Wearable Life Multimedia Streaming System has a three tired architecture, as illustrated in figure 1.
The *Camera Server* is responsible to capture audio and video streams. Basically it consists of three components:

- A so called *medialocator* points to a
- *capture device* like a microphone or a camera [10].
- Then the data are passed to a *processor,* which converts the input stream to a suitable format for the transmission (e.g. MJPEG for video and MPEG for audio).

The following code snippet shows these steps. First the medialocator and the data source object are created then a processor is created with a suitable codec plugin for the camera. Finally the media tracks can be accessed.

```
public StreamManager(CaptureDeviceInfo
camera)
{
       locator = camera.getLocator();
       ds =
Manager.createDataSource(locator);
…
       processor =
javax.media.Manager.createProcessor(ds);
…
       tracks =
processor.getTrackControls();
}
```

Figure 2: The Web User Interface of the Wearable Life Multimedia Streaming System

Afterwards the data are transmitted to the server over the internet using the RTP protocol.
In order to provide the required feedback functionality the server also has a chat interface for written messages and it has a voice feedback channel for audio transmissions from one of the users.
The middle tier called *Streaming Server* is responsible for the management of the connections between user and presenter. The Streaming server receives the AV stream from the sender. A user who wants to view this stream connects to the server, which in turn generates copies of the AV streams and transmits them to the client. Furthermore it distributes the chat messages from and to the users and the presenter. Finally it manages the audio connection to the presenter. Since only one user can get the permission to talk to the presenter at the same time, it has to additionally provide the appropriate management functionality.
A web based *User Interface* has been realized to provide an easy access to the AV stream. The user interface incorporates the video player as an applet (see figure 2). The page is basically written in HTML together with some javascript functions for handling the interaction with the Streaming Server.
The left part of the web page shows the video from the presenter. The window underneath displays the chat messages. The user can type in own messages in the input line under this window. The help button displays a help page. The user can leave the transmission by pushing the quit button. Pushing the "Get Voice" transmits the request to the presenter, who can grant or reject the permission to talk to the user. The right part of the page shows the area, where the transmission takes place. The presenter can display his current position on this map so that the users have a possibility to better follow the presentation.

4. CONCLUSION AND FUTURE WORK

We have presented the concept and the realization of the Wearable Life Multimedia Streaming System.
The system has been designed for the state garden show and will be evaluated in the real world later this year.
The system uses the java media framework, which has showed to be a robust and easy to use middleware for these kinds of systems. Especially the support of different operating systems and data formats (codecs), the integration of the player as an applet into the web user interface are great advantages of the JMF framework.
However the documentation is inaccurate in some points.
The system is implemented on a wearable computer Xybernaut MATC with Pentium III 400 MHz and a Logitech Quickcam.
In the future we want to integrate a camera with a hardware encoder in order to increase the performance of the system. Furthermore we are about to prototype a DirectX implementation to compare the performance between the two different AV libraries.
Together with the UMTS group of the German Research Center For Artificial Intelligence [11] we are also prototyping a Symbian OS client, which is able to stream Video data to the internet using a commercially available cell phone.
Finally we are looking into exploiting and adapting the system to different domains. As already mentioned earlier the system can be integrated into an E-learning application as well, allowing immediate feedback from the audience.
The source code of the Wearable Life Multimedia Streaming System is available over the internet [12].

5. REFERENCES

[1] Kevin Savetz, Neil Randall, Yves Lepage, MBONE: Multicasting Tomorrow's Internet, John Wiley & Sons Inc, 1996

[2] W. Hürst, G. Maass, R. Müller, T. Ottmann, The `Authoring on the Fly' System for Automatic Presentation Recording, in Proc. of ACM CHI 2001 Conference on Human Factors in Computing Systems, Seattle, WA, USA, March 2001.

[3] Netmeeting Home, Microsoft Corp., http://www.microsoft.com/windows/netmeeting/ Download, 06/25/2004

[4] Darwin Streaming Server, Apple Corp., http://developer.apple.com/darwin/projects/streaming/, Download, 06/25/2004

[5] Microsoft Windows Media Home, Microsoft Corp. http://www.microsoft.com/windows/windowsmedia/default.aspx, Download, 06/25/2004

[6] Helix Community, https://helixcommunity.org/, Download, 06/25/2004

[7] Java Media Framework, http://java.sun.com/products/java-media/jmf/, Download, 06/25/2004

[8] JMF 2.1.1. Supported Formats http://java.sun.com/products/java-media/jmf/2.1.1/formats.html, Download, 06/25/2004

[9] Peter Ziewer, Helmut Seidl, Transparent TeleTeaching, ASCLITE 2002, Unitec, Auckland, New Zealand

[10] Linden DeCarmo, Core Java Media Framework, Prentice Hall, 1999

[11] DFKI GmbH, German Research Center For Artificial Intelligence, UMTS-Doit Project, http://www.umts-doit.de/, Download, 06/25/2004

[12] Thorsten Roth, Mobile Realtime Multimedia Streaming, http://www.ainformatik.fh-trier.de/~schneider/mrms/IndexMRMS.html, 11/30/2003