

# Digital Jewellery – The technical part

Nico Fremann

Institute for Geoinformatics, Münster  
n\_frem01@uni-muenster.de

Dennis Wilmsmann

Institute for Geoinformatics, Münster  
dennis.wilmsmann@uni-muenster.de

## ABSTRACT

In this paper we describe the technical part of the Digital Jewellery project at the Institute for Geoinformatics Münster, Germany. This project takes place during summer term 2007. The two main parts are a video player containing a bluetooth scanner running on our iDisplays and several digital jewellery created by Jayne Wallace and Patrick Olivier at the Culture Lab Newcastle of the University of Newcastle. This text focuses on the software part, not the digital jewellery part.

## KEYWORDS

public displays, java, bluetooth, scanner, videoplayer, digital, jewellery, media, video

## INTRODUCTION

During the last years, large displays have been getting cheaper and cheaper. So more and more large displays are placed into the environment. The Digital Jewellery project tries to establish a more personal connection between people and these displays. Normally, only impersonal informations are shown on displays, like news, timetables or commercials. But these displays could also be used for showing some more of „you“. Our idea is to create some digital jewellery what will be recognized by displays and show some personal information about you. In case you think, personal information are only your name, your address, your birthday, your mobile phone number, etc., you're wrong! How about showing some holiday photos or movie clips? Or movies from your work or hobbies? Or a movie clip you've just taken on your way to work? Maybe it would be cool to show to your friends or colleagues or just any person walking by the displays with you. We try to find out how you and the environment will react to this.

## TEST SETUP

### Technical requirements

Currently six iDisplays are running 24h at the Institute for Geoinformatic. They are placed at both entrances of the institute and evenly spreaded at the office corridors, so every IfGI member walks by at least one iDisplay on his way to his office or lecture room. During a normal day, everybody will walk by more than one display. This is

necessary because our iDisplay are a major part of the test setup. The digital jewellery is created individually for every person based upon the video clips they recorded. The jewellery are invented by Jayne Wallace and Patrick Olivier at the Culture Lab Newcastle.

A bluetooth scanner software has been developed during the project "Interaction with public displays" at the IfGI in summer term 2006. The code will be re-used and adjusted to the needs of the Digital Jewellery project. Additionally a video player software will be developed using Java and coupled with the bluetooth scanner.

The software will run on the iDisplays and scans enduring its environment for the digital jewellery. If any digital jewellery is found, a video clip will be loaded and shown on the iDisplays.

The software will run 24h a day during the evaluation period.

### General conditions

The evaluation of the whole project will take place during august and September 2007. Because of the focus of this paper on the technical part of the project, the result of the evaluation will be available in another paper. The four selected participants are a professor, an institute employee, a PhD student and a Graduate student, so there is no problem because of the evaluation period isn't located directly in the summer term.

## TEST PROCEDURE

### Technical part

During the last projects using bluetooth in any way, the open source java bluetooth implementation BlueCove has proved its vantages. The major disadvantage of BlueCove, the missing Linux implementation and the only usage of the Microsoft bluetooth stack, can be left out of focus because all of the iDisplay systems are running Windows XP SP2. The used video clips from the participants are located on (different) webserver so that the video player is able to load them on demand and video clips can be exchanged without the need of changing anything on the iDisplays directly. Any configuration needed for the project and all of the produced information for the

evaluation is stored in a central database.

### **Bluetooth scanner**

The iDisplays need to know if any of the participants wearing the digital jewellery are in range. Face recognition is no alternative because of privacy aspects and the problem, that participants should be recognized “walking by” without the need for standing directly in front of a camera. A bluetooth scanner has been developed during the “Location Based Services” seminar by the project “Interaction with public displays”. The scanner source was already used by different projects and has proved its vantages, so it will be re-used also for this project here.

The bluetooth scanner works in a passive way without the need of any software saved or running on the client bluetooth devices. There is just no alternative to this because the digital jewellery only contains a bluetooth device without an operating system around it like a mobile phone has.

A continuous scan must be guaranteed. Our bluetooth scanner looks for the digital jewellery available in range and commits this information to the video player part of the software. A major problem is the definition of “range” in this context. Bluetooth is working invisible, so not only bluetooth devices in front of the iDisplays running the bluetooth scanner are recognized, also devices located in offices or even on another floor are recognized. Also lots of devices and mainly humans are disrupting the range. For example, a single person wearing the digital jewellery walking down the corridor is recognized by the scanner within a range of 10 or more meters. If a group of humans is walking through the corridor, the person wearing the digital jewellery is first recognized within a range of 2 or 3 meters. These problems are well known and a solution hasn't been found yet.

The duration of a bluetooth scan is about 11 seconds. After these 11 seconds, all of the found bluetooth devices are presented by the scanner, not during the scan. This is a problem of the Windows bluetooth implementation, as far as we know, so this has to be taken as it is. During other projects using this bluetooth scanner, it has been discovered, that it's dependent on the bluetooth device, when this device is found by the scanner. Some devices have to be “visible” during the whole scan duration, others just have to be “visible” for some seconds or just a piece of a second. So some devices are recognized better than others. All of the digital jewellery pieces contain the same bluetooth technology, so it can be adopted that they will be act in a same way with the scanner producing alike results.

After a bluetooth scan has finished, the gathered information is committed to the video player and the next scan is started. If the video player decides to show a video clip, the bluetooth scan will be paused. The reason for that

and more information can be found in the next part of this paper.

### **Video player**

The video player is the second major part of the software. In general, Java and multimedia are two things that don't consort very much. A proof for that is the Java Media Framework (JMF). It's latest achievement is the mp3 support. And this achievement has been introduced in November 2004! So there is a problem to deal with because the JMF doesn't know anything about modern video codecs like MPEG-4, DivX or XviD. There are several further (open source) project based upon the old JMF but nearly all of them are in a early beta or almost alpha stadium and not adapted for this project.

One project however, looked very interesting because it has its beginnings in 2004 and is based upon FFMpeg, a wide spread open source codec library with support of nearly all common used video and audio codecs. The name of this project is FOBS. This project is still active and the latest version of the software has been released in January 2007. Another great advantage is that FOBS places itself between the old JMF and FFMpeg. So JMF can be used for developing and all encoding is done by FFMpeg. No installation of FFMpeg on the system is required, which reduces the administration complexity on the iDisplay systems.

Our idea is, that the video player will hide itself in the background if its unused. It will get in front of all applications if a video clip is shown. Information about that is gathered by the bluetooth scanner; see section “Bluetooth Scanner” for more information about that. The video clip will be shown in a full screen mode without any borders around it. So the full attention of people walking by or standing in front of the iDisplays could be guaranteed. Video is the only possibility to get the attention because the PC hardware of the iDisplays is normally not directly placed in range of the display and an audio connection between the PC hardware and the iDisplay isn't available everywhere.

### **Database**

A PostgreSQL database is used to store information about the participants, the participants video clips and log entries for evaluation purposes. The following data is saved in three different tables.

1. Participants: Name, eMail address, bluetooth address of the the weared digital jewellery
2. Participants video clips: Number, URL, owner, length
3. Logs: Number, participant, video, time stamp, iDisplay name

The fields of tables one and two are clearly. If a video clip is shown, the owner of this video clip, the time stamp and the name of the iDisplay, where the video clip has just been shown, are saved to the log table. This information is required for the evaluation of the project.

### Digital Jewellery

The digital jewellery will contain a small bluetooth device. What exactly isn't know yet because the devices are still under development. The requirements are that the device will operate continuously, without any external input and will be "visible". In this combination, "visible" means that it will be visible to the bluetooth scanner, like any bluetooth mouse or headset. Higher bluetooth devices can be set to an invisible mode that will hide them from the bluetooth scanner.

### IMPLEMENTATION

#### Bluetooth Scanner

The bluetooth scanner mainly is one single thread which starts several threads for the scan events. During the bluetooth scan, the main thread is sleeping and starts a new scan just after the old one has finished. After, not during the scan, a list of found bluetooth devices is generated. Not only devices belonging to the digital jewellery are found, also mobile phones or laptops with bluetooth are in the list. A lookup for all digital jewellery devices is done by the database and a list of these devices is generated. The list of all found devices is compared with the list of all participants. Only matching devices are taken and committed to the video player.

#### Video Player

The first goal while developing the video player has been the ability to play video clips independent of any used compression codes, e.g. MPEG-4, DivX, XviD. This problem has been solved completely by using FOBS. No configuration has to be done except adding the jmf.jar (from the Java Media Framework) and the fobs4jmf.jar (from the FOBS project) to the build path of the project. Additionally a system library file fobs4jmf.dll has to be placed either into a Windows system directory or just into the directory the project will finally run from. The second possibility has been chosen because so all needed files are stored in a single directory and not widespread over the whole system. To tell the JMF to use FOBS is also very easy; it's done by the jmf.properties file. It's also just placed in the main directory of the project. That's the whole configuration! No external software has to be installed nor any configuration files has to be edited. This is a very great advantage because under this terms and conditions, the whole software will straight run "out of the box" on the iDisplay systems.

The second goal has been the "full screen" ability of the video player. This has been exposed to be a greater problem than adopted. To get rid of any borders or menu bars of the mainFrame, two commands are used: `setUndecorated(true)` and `setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE)`. Now the problem was that a video can be played but it wasn't visible. Only after a manual resize of the window, the video became visible. The solution is a JInternalFrame (Figure 1). Adding the video player to the JInternalFrame and adding this to the mainFrame solves the problem. So a new JInternalFrame is created every time a video is shown and destroyed after the video has finished. Preventing a JInternalFrame from showing any borders or menu bars, the commands `((javax.swing.plaf.basic.BasicInternalFrameUI) getUI()).setNorthPane(null)` and `setBorder(null)` are used.

Goal number three has been a local cache of all available video clips from the project. A video clip must be shown without any delay if a participant has been found by the bluetooth scanner. So it has been necessary to get a local copy of each video clip from the different webservers the participants are hosting their videos on. To clearly identify a video clip, a "primary key" has to be chosen. This is done by the PostgreSQL database using a serial field. Everytime a new video clip is added, a definite number is assigned by the database. This number is used in combination with the bluetooth address of the participants digital jewellery to store a video clip locally. For example, the bluetooth address of a sample participants digital jewellery is 0017e44df731 and a sample video clip of the participant is `http://ifgi.uni-muenster.de/~sampleparticipant/videoclip.avi` with number 13 assigned by the database. Based upon this information, the video clip is stored locally as `13-0017e44df731.avi`. On start of the program, a database lookup for all video clips is done, compared with already locally saved (cached) video clips and missing clips are downloaded and saved. This will prevent the player from loading the video clip directly from a webserver when this video clip has to be shown. This solution reduces the delay between the time when a digital jewellery device is found and the start of a video clip from more than 5 seconds under 1 second; the video is shown nearly instantly.

The video player gets a list of found participants from the bluetooth scanner. A participant is randomly chosen from this list. Based upon this information, a database lookup is done for all available video clips of this participant. In the same way, a video clip is chosen randomly. If available in the cache, the video clip is shown nearly instantly. Because the cache of video clips is generated at the start of the program, it could be possible, that a video clip has been added to the database during runtime of the program. If a

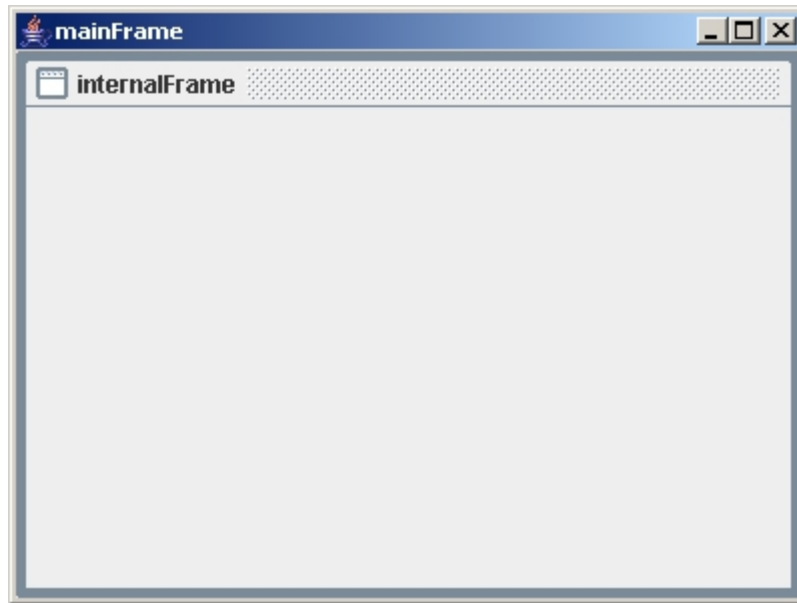
video clip isn't found in the cache, is loaded from the webservice and added to the cache. So the next time the video is chosen, it's available in the local cache without the need of downloading it again. So there is no need to restart the program on the iDisplay system if new video clips are added. This also reduces the administration complexity heavily.

Information about time and location of the video play-back

are stored in the database for evaluation and maintenance.

The main implementation of the video player has been taken from several documentations available at the JMF homepage by the Sun Developer Network, so no more details are mentioned here.

The program sequence is shown by figure 2 on the last page of this paper.



**Figure 1:** Using a JInternalFrame inside a JFrame

## CONCLUSION

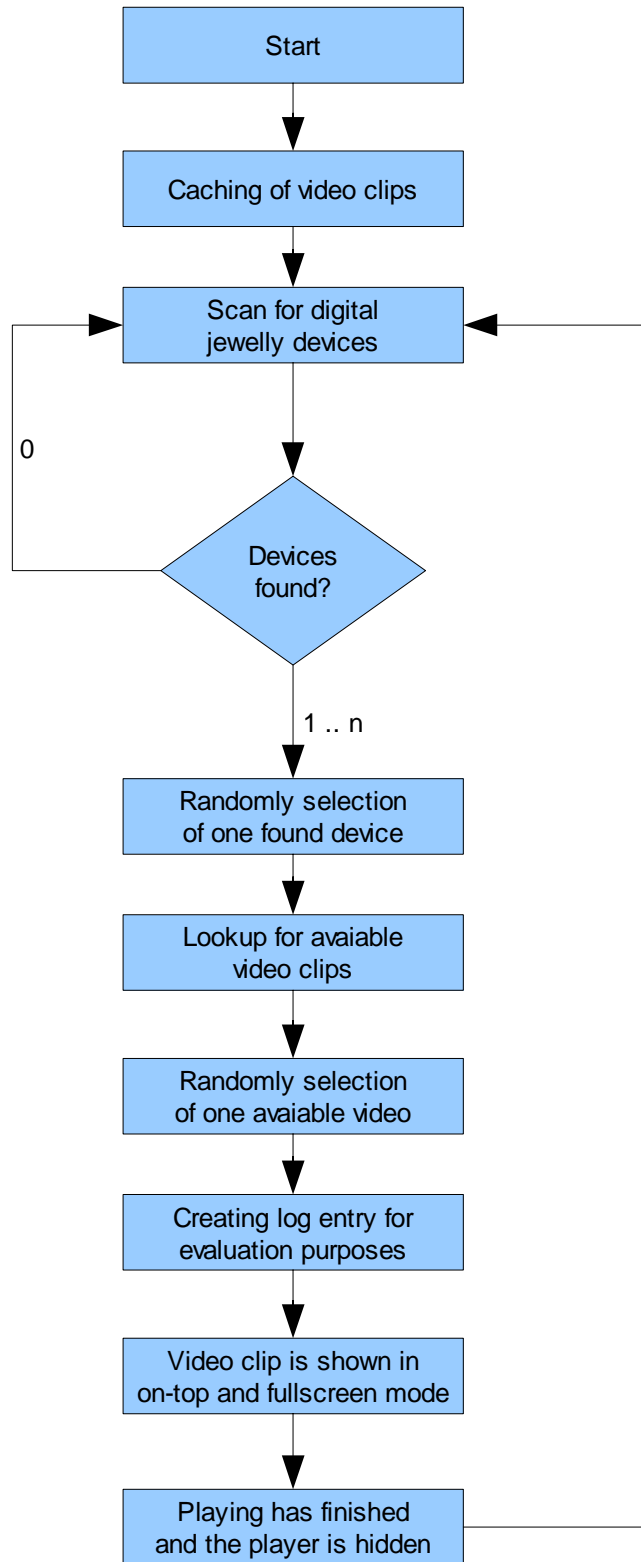
The evaluation conclusion of the whole project will be presented in another paper after the evaluation has finished in august 2007.

The bluetooth scanner and video player developed during this project are able to play video clips independent from any codecs based upon found bluetooth devices in full screen mode on the iDisplay systems at the Institute for Geoinformatics. During a short test period the software runs 24h without any problems on the iDisplays.

## REFERENCES

1. Fremann, N., Jentsch, M., Wilmsmann, D. Interaction with public displays. <http://www.denniswilmsmann.de/studium/lbs/doc/InteractionWithPublicDisplays-Ausarbeitung.pdf>
2. iDisplays. <http://www.idisplays.info/>
3. Java Media Framework API (JMF) <http://java.sun.com/products/java-media/jmf/>
4. FOBS <http://fobs.sourceforge.net/>
5. BlueCove <http://code.google.com/p/bluecove/>

## PROGRAMM SEQUENCE



**Figure 2:** The program sequence