

Bluetooth broadcasting

Johan Ho

Master of Science in Computer Science

Submission date: June 2006

Supervisor: Alf Inge Wang, IDI

Problem Description

The purpose of this project is to research Bluetooth broadcasting and what usage areas it can be used for. Other wireless communication technologies will also be compared to Bluetooth, and a test system will be developed for testing the research results.

Assignment given: 20. January 2006
Supervisor: Alf Inge Wang, IDI

a Abstract

Background: The wireless technology Bluetooth has rapidly become more commonly supported by electronic devices like mobile phones and PDAs. Several companies are currently developing Bluetooth broadcasting systems to use for marketing. This report is a result of researching the use of Bluetooth broadcasting for delivering information for more general purposes, how well Bluetooth actually works for broadcasting, and also on the topic of user privacy.

Results: Broadcasting with Bluetooth did work with a few devices at the same time, since Bluetooth allows up to seven connections to one Bluetooth radio at the same time. By making a passive system where the user is the one which requests information, it also will not affect users' privacy. However, my research also found a few issues with the Bluetooth which might affect a broadcast, the most noticeable of them being the somewhat low transfer rate, an issue with device discovery not always working when a lot of users are doing device discovery at the same time. The fact that it only supports seven connections is also a limitation. Basically, while it is possible to use Bluetooth for broadcasting, it might be problematic to use it for targeting a large audience.

Conclusions: Even with the problems mentioned, Bluetooth broadcasting provides quite a unique way of broadcasting, and with the newer versions of Bluetooth the issues mentioned might be less of a problem. Bluetooth broadcasting definitely has some potential.

b Preface

This report is the result of the work done by Johan Ho in the masters degree at the Institute for Computer and Information Science (IDI) at the Norwegian University of Science and Technology (NTNU) Spring 2006.

I would like to thank my supervisor, Alf Inge Wang, for providing resources, ideas and feedback on this project.

Trondheim, 07/06/2006

A handwritten signature in black ink that reads "Johan Ho". The signature is written in a cursive style with a horizontal line underneath the name.

Johan Ho

c Contents

a Abstract.....	i
b Preface.....	ii
Part I: Introduction.....	1
1 Background.....	2
1.1 Problem definition.....	2
1.2 Project context.....	3
2 Research methodology.....	4
2.1 Research questions.....	5
2.2 Research agenda.....	6
3 Reader's guide.....	7
Part II: Prestudy.....	9
4 Why Bluetooth broadcasting?.....	10
4.1 Area broadcasting instead of group broadcasting.....	10
4.2 Sending information where people are.....	10
4.3 Portable information.....	11
4.4 “Cost free” distribution.....	11
4.5 More specific examples of usage.....	11
4.6 Broadcasting station.....	12
5 Existing projects.....	13
5.1 Filter UK – BlueCasting.....	13
5.2 LondonDev – BroadTooth.....	14
5.3 Midletsoft – Jellingspot.....	14
5.4 Peekablue.....	15
5.5 Alterwave.....	15
5.6 BlueBlitz – MagicBeamer.....	16
5.7 Thoughts.....	16
6 Target devices.....	18
6.1 Mobile phones.....	18
6.2 PDAs (Personal Digital Assistants).....	19
6.3 Laptops.....	20

6.4 File types.....	20
7 Wireless communication technologies.....	22
7.1 Bluetooth.....	22
7.2 IrDA (Infrared Data Association).....	25
7.3 Wi-Fi.....	25
7.4 RFID (Radio Frequency Identification).....	26
7.5 Mobile phone network.....	27
7.6 ZigBee.....	28
7.7 Comparison chart.....	29
7.8 Thoughts about technologies.....	30
8 User privacy.....	31
8.1 Alternatives to automatic sending.....	32
9 Functionality choices.....	34
9.1 Custom user client.....	34
9.2 On demand versus automatic sending.....	34
9.3 Two-way communication.....	35
9.4 Logging.....	36
9.5 File types.....	37
9.6 Summary.....	37
Part III: Requirements specifications.....	39
10 Introduction.....	40
11 Functional requirements.....	41
12 Quality requirements.....	43
Part IV: Architecture and design.....	45
13 Introduction.....	46
14 Stakeholders and concerns.....	47
15 Selection of architectural viewpoints.....	48
16 Quality tactics.....	49
17 Architectural pattern.....	53
18 Views.....	54
18.1 Scenario view.....	54

18.2 Logical view.....	55
18.3 Process view.....	57
19 Changes.....	58
19.1 Quality tactics.....	58
19.2 Architectural pattern.....	58
19.3 Views: Scenarios.....	58
19.4 Logical view.....	59
Part V: Implementation.....	61
20 Requirements.....	62
21 Baloo's GUI.....	63
21.1 Main configuration screen.....	63
21.2 Managing broadcasted resources.....	66
21.3 Statistics.....	67
22 Accessing the broadcast.....	69
22.1 Using a Symbian Series 60 phone.....	69
22.2 Using a Palm device.....	71
22.3 Using a Pocket PC device.....	73
Part VI: Results and conclusion.....	79
23 Answers on the research questions.....	80
23.1 Research Answer 1.....	80
23.2 Research Answer 2.....	81
23.3 Research Answer 3.....	83
23.4 Research Answer 4.....	84
24 Conclusion.....	87
Part VII: Further work.....	89
25 Further work.....	90
Part VIII: Glossary.....	93
26 Glossary.....	94
Part IX: References.....	97
27 Information resources.....	98
28 List of illustrations and tables.....	99

Part I: Introduction

1 Background

As anyone who has used computers surely knows, things can easily get messy around the computer. Wires and cables are tangled together all over the desktop, or if you are lucky, hidden under and behind it. The number of peripheral units for the computer is increasing, with keyboards, mice, printers, microphone headsets, scanners, digital cameras and MP3 players, to mention some. And with each new unit, the number of wires also increases even more, since each device usually requires at least both their own power cable and a cable to the computer.

To avoid this problem, developers have been researching wireless technologies to reduce the number of wires needed for electronic devices to communicate together. Bluetooth[®] is one of these technologies. With Bluetooth, users can use devices like headsets, printers or mice with their computers, without needing to connect them with a cable. Bluetooth is also supported by an increasing amount of mobile phones and other handheld devices, allowing users to share files with their friends and use wireless add ons like headsets with their phones.

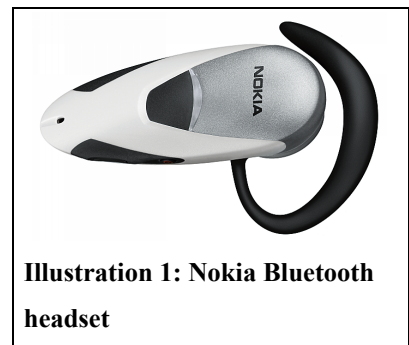


Illustration 1: Nokia Bluetooth headset

However, not until recently has Bluetooth as a broadcasting channel for spreading information been researched. What are the possibilities with such a system? That is what this project aims to find out.

1.1 Problem definition

The Bluetooth technology [1] allows for wireless communication between devices, and can potentially be used to broadcast digital information in a different way than it has been possible with other messaging technologies. A number of companies are already working on developing broadcasting systems to be used for advertisements and marketing via Bluetooth. This project, however, will look at what other usage



areas Bluetooth broadcasting can be used for, through research of the technology and development of a prototype for testing.

1.2 Project context

This project is part of the MOWAHS [2] project (MOBILE Work Across Heterogeneous Systems). The MOWAHS project aims to improve work processes in virtual organizations by providing systems and solutions which aid in communications and information sharing between various devices, from computers to PDAs and smart phones. The MOWAHS project has three goals:

G1. Helping to understand and to continuously assess and improve work processes in virtual organizations.

G2. Providing a flexible, common work environment to execute and share real work processes and their artifacts, applicable on a variety of electronic devices (from big servers to small PDAs).

G3. Disseminating the results to colleagues, students, companies, and the community at large.

The results of this project will contribute to G2, by researching how Bluetooth broadcasting can be used for delivering information to mobile devices. The supervisor for this project is Alf Inge Wang, who will provide guidance for this project.

2 Research methodology

There are many different research methodologies for different types of research. For a software engineering project, which differs quite a bit from research related to production and other areas, there are certain research methodologies that are more appropriate than others. Without following one of these methodologies, one risks that a software project turns out to be just a development project, rather than a research project. Basili [3] described three different research methodologies associated with software engineering:

- **The engineering method:** *observe existing solutions, propose better solutions, build/develop, measure and analyze, and repeat the process until no more improvements appear possible.*
- **The empirical method:** *propose a model, develop statistical/qualitative methods, apply to case studies, measure and analyze, validate the model and repeat the procedure.*
- **The mathematical method:** *propose a formal theory or set of axioms, develop a theory, derive results and if possible compare with empirical observations.*

The first two methods are variations of the scientific method, in which you propose a model, measure and analyse it, and then use the results to validate the model. But while the engineering method focuses on improving existing solutions, the empirical method assumes that a new model is proposed, not necessarily based on any existing models. The third method, the mathematical method, is an analytical method focusing on analysing and improving models based on mathematical models rather than on building a system for testing out hypotheses.

For our project, the method used will be the engineering method. The reason for this is that this project focuses on using existing systems in a new way. As mentioned in the introduction, Bluetooth broadcasting has already been tested by some companies for marketing campaigns, so the technology itself is nothing new. The topic that remains to research is how to use this technology in different ways for benefiting the users. To do this we will, in accordance to the engineering method:

1. investigate existing Bluetooth broadcasting systems
2. propose new usages for the technology and come up with example scenarios and eventual system improvement suggestions

3. develop a prototype application for testing proposed scenarios and suggestions
4. analyze and validate the propositions

2.1 Research questions

The following are the research questions that we want to answer in this project.

RQ1: Is Bluetooth suited for broadcasting information?

We want to find out whether Bluetooth is at all suited for broadcasting information. Are there any major drawbacks to using Bluetooth to broadcast information? What other technologies can be used in a similar way? Are any of those a better choice?

RQ2: What kind of information can be sent with a Bluetooth broadcast?

Here we want to look at different file and data types, and how these can be transferred as a Bluetooth broadcast, without requiring additional software on the receiving devices. We want to see how these file types can be used for conveying information. Maybe some common file types can be used in original ways, or in ways that they have not been used much before?

RQ3: Can Bluetooth broadcasting be done in a way which does not compromise users' privacy?

Not everyone would want incoming Bluetooth connections every time they are within the range of a Bluetooth broadcast. Some people fear that Bluetooth will become another annoying spam channel. Is there any way to avoid this reputation for Bluetooth broadcasting, and yet without making receiving the broadcasts too much of a hassle for normal people to bother with?

RQ4: What usage area(s) would Bluetooth broadcasting be best for?

Are there any usage areas that Bluetooth broadcasting would fit especially well for? What would be the benefits of using Bluetooth broadcast in those usage areas?

2.1.1 Approach

These research questions will be answered using the results from the research and development done in this project, and the answers will be used to form a conclusion. (Testing was originally also to be used for answering the research questions, but since testing had to be cancelled, this is not possible now.)

2.2 Research agenda

There are mainly three phases in this project:

1. **Research and ideas phase**

In this phase we will do research on the technologies that are relevant to this project. We will mainly look at the Bluetooth technology, but we will also describe other similar technologies to see how they compare. We will also look at some relevant user devices that implement Bluetooth, and how they can interact with a Bluetooth broadcasting station. We will then develop a prototype application based on the results of the research.

2. **Development phase**

Here we will develop the architecture for the proposed prototype application based on our requirements specifications, and then implement the application.

3. **Testing and results**

Lastly we will test the prototype application and analyze the results. These results, in addition to the research done in the first phase, will be used for answering the research questions.

Unfortunately, the testing had to be skipped because of lack of time and resources at the end of the project. The last phase will therefore just contain the results based on the prestudy, development and the testing done by the developers during development.

3 Reader's guide

This chapter will describe each part of this report briefly, so the readers can decide which parts may be interesting to them.

- **Part I: Introduction**

The part you are currently reading. This part describes the project, its background and how it will be executed.

- **Part II: Prestudy**

This part shows the result of the research done in this project. It contains information about the different technologies, ideas and solutions suggested for the project.

- **Part III: Requirements specifications**

This part defines the requirements suggested for a prototype application that is to be developed to test the ideas from this project.

- **Part IV: Architecture and design**

This part describes the architecture and design choices for the prototype application. The architecture is a description of how the application is built up in order to fulfil the requirements given in the requirements specification.

- **Part V: Implementation**

This part contains a description of how the application is implemented.

- **Part VI: Results and conclusion**

This part shows the results of the project, based on the experiences during development, and describes the final thoughts about the results of this project.

- **Part VII: Further work**

This part shows what we think could be done further with the topic of this project.

- **Part VIII: Glossary**

A list of expressions and terms used in this document.

- **Part IX: References**

A list of the information sources used in this project.

Part I: Introduction

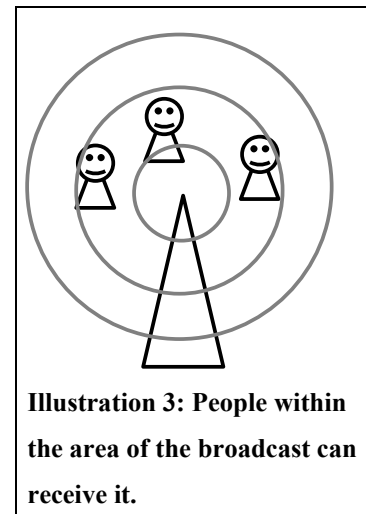
Part II: Prestudy

4 Why Bluetooth broadcasting?

Using the Bluetooth technology for broadcasting information allows for a way of broadcasting that is quite different from any other currently common broadcasting technologies. Bluetooth broadcasting will most likely not be able to replace broadcasting systems like television and radio broadcasts, or even message transmission technologies like e-mail, SMS/MMS and instant messengers (one example on instant messengers is MSN Messenger). However, Bluetooth broadcasting can be used in situations where the other technologies are not as suitable. The following subchapters will describe what kind of usages Bluetooth broadcasting will be most suited for.

4.1 Area broadcasting instead of group broadcasting

Unlike other digital channels for sending information, Bluetooth broadcasting allows for directing broadcasted information towards all people that are in the area, rather than a group of known people. This can be compared with a short range radio or TV broadcast, where everyone within the area of the broadcast can receive the signal, as long as they have a receiver. The reason we say short range is because implementations of the Bluetooth technology usually have quite limited range.



When compared to digital messaging services like SMS, MMS and e-mail, the most noticeable difference is that these services require a list of known users in order to send messages. Bluetooth broadcasting is therefore more useful compared to those services when the broadcasts are meant for unknown or temporary recipients. One example is people that are visitors of a place, for example a museum. In those cases the visitors does not need to be known beforehand in order to be able to access the broadcasts, they just need to be within range.

4.2 Sending information where people are

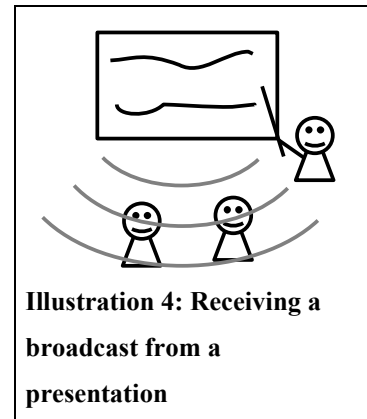
Sometimes information posters or notices may have links to website, or some other kind of contact information which interested people would find useful. Writing it down may be too much

of a hassle for some people though, especially long visiting addresses or website addresses. If this information could instead be broadcasted from a small Bluetooth station placed close to the information notices, it could broadcast the contact information so that it is automatically stored as a contact or just as text. Meeting schedules could be automatically added to the calendar, with an alarm to remind people of meetings.

4.3 Portable information

When sending information as e-mail or putting it on the Internet, the information would only be accessible from a computer, which in most cases is not easily portable. To view the information away from home or work, people would have to print it out or write it down. However, information sent to mobile phones would be easily portable and viewed at any time.

In the example in the previous subchapter, the broadcast could send some or even all of the information that the notice contains, so the user can view it at any time. If the notice contains address and phone information, this could be added to the user's address book so that he easily can locate or contact the owner of the notice.



4.4 “Cost free” distribution

While SMS and MMS messages could to some extent fill the same purpose as Bluetooth broadcasting, using Bluetooth has the additional advantage of being free to use. There is no need to pay for each message sent to users. It is not completely free of course, since it will still cost money for the hardware used for broadcasting, and for maintenance.

4.5 More specific examples of usage

The following list contains more examples on things that Bluetooth broadcasting could be used for:

- Public transportation schedules: A Bluetooth station could broadcast information about the bus or train schedules. This information could then be stored on their mobile phones so they do not need to be near the stations to look up the information.

Another way to use this is to send the departure times for the buses or other transportation

nearby for the next hour or so. This would be useful when broadcasted in a place where a lot of people are, without being right next to the stations, like shopping centres, schools/universities and company buildings. People would then know if they need to rush to the station to catch the next bus, or if they still have time to spare.

- Concert halls, theatres and similar places could broadcast calendar events to inform about upcoming events. These events could be added to the calendar software in user's devices.
- Museums could broadcast information about opening hours and special events. Inside the museum it could broadcast information about museum items or areas.
- Trains and buses could broadcast information inside the vehicles, for example the list of bus/train stops, travel time, modified vacation schedules, and other information.
- University/school classes could broadcast calendar events for lectures and important deadlines, and for example information about the syllabus and the books to be bought for a class.
- And of course, it could be used for advertisements, but this will not be discussed in this project since most of the other projects on Bluetooth broadcasting are focusing on this.

4.6 Broadcasting station

In this project we will use the term “Bluetooth broadcasting station” for the devices which do the actual broadcasting. A broadcasting station may consist of a computer with one or more Bluetooth radios, and running a broadcasting application. However, it is not always practical to use a computer for broadcasting, depending on where one wants the broadcast to be. For example, if someone wants a Bluetooth broadcast near a poster or notice on a bulletin board, they may not be able to put a computer near it. In those cases it would be necessary to have a smaller device specialized for storing information and broadcasting it. These devices will also be called Bluetooth broadcasting stations. In this project, the use of small Bluetooth devices for broadcasting will not be researched: this project will focus on the possibilities of Bluetooth broadcasting as a system. It is in any case quite possible to produce small Bluetooth stations, as a few of the companies mentioned in “5 Existing projects” already sell such products. The prototype application which will be developed in this project will however be tested on a laptop with an Bluetooth USB adapter.

5 Existing projects

This part of the report will look at existing projects using Bluetooth broadcasting in some form, and compare the choices each project has chosen for their solutions. There are quite a few projects on this topic, so we will keep information about each project brief.

5.1 Filter UK – BlueCasting

BlueCasting [4] is developed by Filter UK, and is described as “The Proximity Marketing System”.

The system bases itself on adding small Bluetooth station boxes on for example advertising posters.

When mobile devices with Bluetooth enabled are in range, the device information is sent to a central server, which decides what contents to send, and

sends it to the device via the Bluetooth station. The website also states that a transaction history will be stored for each device. The transaction history, together with other rules like what time of day it is, or how many users have already received the broadcasts, can be used to adjust the information sent to each device.



The website lists the file types supported by their system:

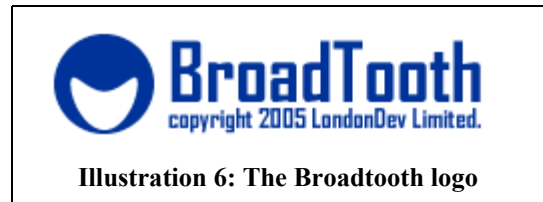
- Text [as TXT files]
- Still images [as GIF or JPG files]
- Animated images [as animated GIF files]
- Audio [as WAV, RMF, MP3, MP4 or ring tone files]
- Video [as RM, 3GP or MP4 files]
- Java Applications [as JAR files]
- vCard [Business card files]
- vCal [Calendar event files]

This solution has already been used in a few campaigns, one of them being an advertising campaign for the band Coldplay, where they set up stations on the posters broadcasting a promotional video of the band to users' mobile phones. This campaign was described in an article

in Time magazine in April 2006 [5], and was also mentioned in IEEE Pervasive Computing in October 2005 [6]. Other companies that were mentioned on their website to have used their system are Volvo and British Airways. It therefore seems like this might be the most mature Bluetooth broadcasting project, or at least the most successful one so far.

5.2 LondonDev – BroadTooth

BroadTooth [7] is a Bluetooth broadcasting system developed by LondonDev Limited. There is not much information about the project yet, and their website only has one page, which lists the features of their system.



The BroadTooth system is described as a marketing solution able to send “multiple advertisements and messages to passing phones and review the success of the broadcast in real time”. The website states that their software will be run on a laptop with Bluetooth support.

5.3 Midletsoft – Jellingspot

Jellingspot [8] is a Java Bluetooth broadcasting application developed by the Midletsoft Corporation. This product is already released and usable (a free fully working beta version can be



downloaded from their website), but requires the users to install a client application on their handheld device to be able to connect to the broadcast stations. They have a selection of client software for a wide range of different mobile phone models on their download page.

There are a few news articles on the company website [9] about Jellingspot being used in various places. However, the newest articles are from early 2004, so the current status of the project is uncertain, although their developers' forum is still somewhat active.

The website informs that Jellingspot is able to send “electronic coupons, informational text, audio, video, corporate information, and more”. The demo states that the commercial version can

send “MP3, 3GP, JPG, GIF, PDF, TXT files and more”, apparently it can send any file types, regardless of whether the receiver can receive it or not. In an e-mail from them, they said this may change to only support file types which are known to be usable on the portable devices.

One thing to notice here is that since Jellingspot's solution requires users to load up a client program and connect to the broadcasting stations manually, it will not affect user privacy by repeatedly trying to send messages to people walking by.

5.4 Peekablue

This product is slightly different from the other ones mentioned here. **Peekablue** [10] is a mobile phone program which allows the user to broadcast images from

their mobile phones to other Bluetooth mobile phones that are in range. It also allows the user to create a “Friends list” in order to avoid strangers to access the pictures. The current version is only available for 17 different mobile phone models, most of them Nokia models. Unfortunately, we can not test it since we do not have any of the models listed.



5.5 Alterwave

Alterwave [11] has developed Distri-Lite and Distri-Full, which are two slightly different Bluetooth broadcasting systems, though both use the same infrastructure.



Distri-Lite works by broadcasting a few messages (welcoming messages, ads) automatically, and then allowing distribution of further information on demand, allowing users to request content by sending messages to the Bluetooth station. It also stores statistics about system activity for the store owners to see the broadcasting traffic.

Distri-Full requires users to download a client application in order to access the contents of the broadcast. This client is broadcasted to the users in range, though how their system differs

between different types of devices or even if they differ between them at all is uncertain.

Anyway, the client can then be used to browse through and request content from the broadcast, and the content is sent to the user devices' inbox.

The website informs that their product has been used in Popkomm 2005 (the largest Music Festival in Germany) for sending MP3 files to mobile phones over Bluetooth.

5.6 BlueBlitz – MagicBeamer

BlueBlitz [12] has made a solution called MagicBeamer, which supports four services:

1. broadcasting messages and media
2. sending information and media by request (the users send messages to the Bluetooth station to request content)
3. receiving messages and media and forward these via e-mail
4. allowing users to connect to the Bluetooth station and use an Internet connection through it

They also sell broadcasting stations that use their broadcasting software. These stations can be set up and then deployed without having to be connected to a computer.

Their software can be run on a computer with Bluetooth.

The software comes on a bootable CD and acts as a simple operating system, so the computer has to be booted with the CD in order to run the software.

5.7 Thoughts

Most of the projects described here are solutions aimed towards marketing. The fact that so many companies already has developed solutions for advertising via Bluetooth broadcasting, makes it even more important for this project to research the usage of this technology in other ways. The Peekablue product is an example of a use of Bluetooth broadcasting which benefits the users.



Illustration 10: BlueBlitz logo



Illustration 11: BlueBlitz broadcasting station

Jellingspot also stands out because it can support a wide range of services, by allowing developers to add services by creating “Jellingspot Service Beans”, which can be added to Jellingspot server. The disadvantage is that the Jellingspot solution requires a user client for it to work, but this could also be an advantage, depending on how you look at it. For example this would mean that users will not get requests for incoming transmissions from the Bluetooth stations, instead they will have to start up the client application and manually connect to them to download information.

The last two services listed in BlueBlitz' product information also stand out, since those two services, forwarding messages and providing Internet access, will allow two-way communication for the users. With these services the users can not only fetch information, but also potentially provide information to the broadcasting station.

As these examples show, there are already a variety of different implementations of Bluetooth broadcasting: some require user clients to work while others do not, some send most of the information automatically while others send most of the content on demand. In “9 Functionality choices” the advantages, drawbacks and possibilities for different methods will be discussed.

6 Target devices

For a Bluetooth broadcasting system, the main targets for the broadcast will be devices that move within range of the broadcast, without necessarily residing there. In most cases this will be mobile phones and PDAs. However, Bluetooth-enabled laptops could also possibly be a target. While people usually do not leave their laptops powered on while traveling around with it, they could still receive broadcasts in places where they can sit down and use their laptops, like lecture halls, cafés and libraries.

This chapter will describe the different target devices and their Bluetooth capabilities, and discuss what needs to be taken into consideration for the different devices.

6.1 Mobile phones

Mobile phones are portable phones that communicate with other phones via wireless mobile phone networks. While originally only usable for phone calls, most modern mobile phones can also send and receive text and multimedia messages, and many phones also add extra functionalities like camera and music playback. Some mobile phone models are a hybrid between mobile phones and PDAs, making them practically PDAs with mobile phone functionality. In this project these phones can be treated as PDAs, as the Bluetooth support for these devices will be quite similar to PDAs.

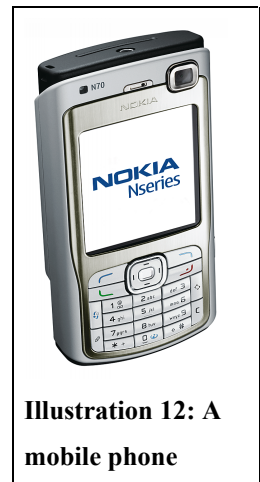


Illustration 12: A mobile phone

Due to the large variations in what is supported by different mobile phone models, it is difficult to say what kind of file types can be used in a broadcast targeted at mobile phones. Some of the simplest phones are not able to accept much except for image files, while the more advanced phones can accept close to everything (although this does not necessarily mean that they can show all the files without additional software).

One type of mobile phones that has advanced functionality, are mobile phones using Symbian OS, which is an operating system which is actually more like a PDA OS than a mobile phone OS. It allows users to install additional software on the phones, and features a user interface with an

icon-based menu system for browsing through installed software. Symbian OS is owned by a number of companies, including Nokia and Sony Ericsson.

6.2 PDAs (Personal Digital Assistants)

PDAs are handheld computers with a touch screen for user interaction. They were originally just meant to replace personal organizers, but all PDAs today are powerful enough to run a vast number of different programs, making them practically small personal computers.

The two most used operating systems for PDAs are Palm OS (developed by PalmSource) and Windows Mobile (developed by Microsoft), formerly known as Pocket PC (note that the name Pocket PC is now used for PDAs that fulfill certain criteria, one of them being that it must run using the Windows Mobile OS). Both operating systems supports Bluetooth (if the device itself has a Bluetooth radio), but the way they handle it differs greatly:

- Palm OS will receive any file via Bluetooth, however, if no application is set to handle the received file type, the Palm OS device will state that the file is of an unknown type, and discard it.
- Windows Mobile (or rather, Pocket PC 2002, which was what was tested) seems to receive all file types, but if the file type is not either a calendar event, a todo item or a address book contact, it will just be left in the temp folder. No messages are given when the device has finished receiving the file either, leaving it up to the user to open File Explorer and find the file manually when he thinks it has finished.

In a way Windows Mobile devices are more flexible, since they will at least store all files that are received, while Palm OS devices discards unknown files. However, Palm OS is more user friendly, since it will store recognized files in the right place for that specific file, and load up the program which is registered to show that file.

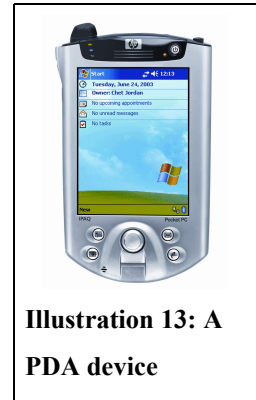


Illustration 13: A PDA device

Either way, it seems like Bluetooth support on PDA devices is not quite as good as one would expect. Most likely it is best if files that are targeted towards PDAs are usable by one of the commonly preinstalled programs in the receiving device.

6.3 Laptops

Laptops are portable computers, with a keyboard and a flat screen that can be folded over the keyboard. While laptops used to be twice as expensive as stationary computers in the early days, the price has dropped significantly, making them a lot more common for personal usage.

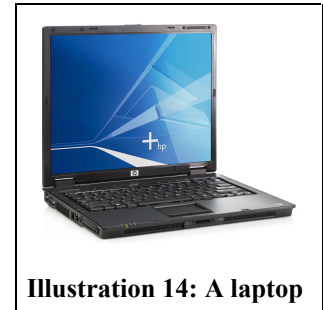


Illustration 14: A laptop

While laptops have become quite common, they were usually not sold with built in Bluetooth support until recently. With the increasing amount of available Bluetooth peripheral devices like headsets, printers and keyboards, more laptops are now shipped with Bluetooth support built in.

Laptops with Bluetooth can usually accept anything sent to them, allowing users to accept also uncommon file types which may be used by software they have installed. Sending files to laptops should therefore not be much of a problem.

6.4 File types

Since both PDAs and mobile phones will have large limitations in the type of files that can be sent, this has to be taken into consideration in a broadcasting application. Three ways to solve this problem is:

1. Send only file types that we are certain all (or close to all) devices can handle.
2. Try to recognize what device we are sending to first, before sending anything.
3. Let the users choose what files to fetch.

Solution 1 is the most straightforward way to solve this. In “5 Existing projects”, one of the existing projects had listed up a number of file types that were supported:

- Text [as TXT files]
- Still images [as GIF or JPG files]

- Animated images [as animated GIF files]
- Audio [as WAV, RMF, MP3, MP4 or ring tone files]
- Video [as RM, 3GP or MP4 files]
- Java Applications [as JAR files]
- vCard [Business card files]
- vCal [Calendar event files]

This list does indeed contain most, if not all of the most commonly supported file types on mobile phones. PDAs should support most of these file types and more, although some PDAs might not support videos. Not all of the mentioned file types will work on all mobile phones either. JAR files in particular are a bit tricky. Even if a phone supports Java applications, the applications usually need to be adapted to that phone model or series in order for the application to work properly. The reason is that there are differences in the Java implementations and screen sizes on different phone models.

Solution 2 is used by a few of the existing projects. The BlueCast solution does this by recognizing the handset based on its Bluetooth ID. This is possible because ranges of Bluetooth IDs are given to producers of Bluetooth devices, which then assigns subranges to different models of their devices. The third solution would also certainly work, assuming that the users have enough knowledge to know what files their cellphones support.

The different solutions will be discussed in “9 Functionality choices”.

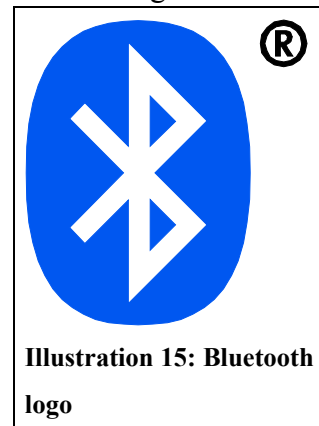
7 Wireless communication technologies

This chapter will describe a number of wireless communication technologies that are available today. Although this project focuses on using the Bluetooth technology, it is still of importance to take other technologies into account, to see if any of those technologies could be a better choice than Bluetooth for the purpose of broadcasting. Other technologies will therefore be described and compared to Bluetooth.

7.1 Bluetooth

Bluetooth [1] is a specification for wireless transfer of data between devices using radio waves.

While it is not nearly as fast as the standards used for wireless local area networks (Wi-Fi), it is cheaper to implement, and is designed for lower power consumption, making it ideal for portable devices like PDAs and mobile phones. The specifications for Bluetooth version 2 was released in 2004, though Bluetooth version 1.x (1.0, 1.1 and 1.2) are still more common in devices used today.



Each Bluetooth device has a unique address, also called Bluetooth ID or device ID. This ID is usually not shown to users, as the more user friendly customizable Bluetooth name is shown instead.

One popular use of Bluetooth is for wireless headsets. A Bluetooth headset would have a Bluetooth radio built in, allowing it to connect to mobile phones or computers, so that people can use the headset without connecting it with a cable. Many mobile phones also let users use Bluetooth for transferring data with other phones or with computers. Other computer peripherals that have Bluetooth versions include printers and input devices like mice and keyboards (Illustration 16). There are also several projects researching the use of Bluetooth in other usage areas, like mobile payment [13] and audio streaming [14], proving that there is indeed quite some interest in the Bluetooth technology.

Bluetooth operates with radio waves using the 2.45 GHz band. This band is license free, and is therefore also used by other appliances such as microwave ovens and baby monitors. To avoid interference with other devices, the Bluetooth protocol divides this band into 79 channels, each of them 1 MHz wide. A



Illustration 16: Nokia Bluetooth keyboard

Bluetooth transmission changes between these channels up to 1600 times per second.

The transmission range of Bluetooth ranges depending on the power class of the implementation. It has three power classes, where class 2, with a range of about 10 meters is the most common. The others are class 1, up to 100 meters, and the rarely used class 3, which has a range of 10 cm, up to a maximum of 1 meter. The maximum speed of Bluetooth transfers is 723.1 kilobit/s in version 1.1 - 1.2, and 2.1 megabit/s in version 2.0 [15].

Devices with Bluetooth can usually set it in one of three modes: Off, hidden and discoverable. When set to discoverable, other devices will be able to see it when searching for devices, and can attempt connecting to it. When set to hidden, other devices will not be able to see it when they search for devices. Devices will have to know the Bluetooth ID in order to connect to a hidden device. This can be done by pairing the devices, by using a pass code which the users agree on (or in the case of Bluetooth add-on devices like headsets: a factory preset or user-configurable code on one of the devices). This has to be done when at least one of the devices is discoverable. When the code is recognized, the two devices will store the other device's Bluetooth ID, so that they can connect to each other even when they are set to hidden.

7.1.1 Performance with multiple simultaneous connections

It is possible for a Bluetooth device to connect to up to seven other devices at the same time. The main device is called the master, and the seven connected devices are called slaves. A network group with up to 8 devices like this is called a *piconet*. Not all devices can connect to several other devices at the same time, for example most mobile phones can only connect to one device at the time. Such devices can not be masters in a piconet, but they can be slaves.

Part II: Prestudy

A master device can only communicate with one slave device at a time, but it switches quickly between them to allow “simultaneous” transfer. The transfer rate will obviously be lower when connecting to several devices, lowering the speed of a broadcast as the number of users increase. It is to a certain degree possible to counter this and the 7 connections limitation by using multiple Bluetooth radios on the broadcasting station. However, this would require a more complex hardware and/or software solutions, since Windows XP by default does not support more than one Bluetooth radio at a time [16]. Also, since there are a limited amount of channels, more Bluetooth radios would increase the risk of connections interfering with each other, lowering the performance.

Another factor that affects connection performance is the discovery time, or the time that a Bluetooth device uses when scanning for devices that it can connect to. This is a lengthy operation which is one of the flaws of the Bluetooth protocol. In a small test done using Bluetooth version 1.x devices in another research project [17], the time just for the discovery was found to be at least 16 seconds, and then additional 3-4 seconds were required for the handshake and connection. This would severely reduce the amount of broadcasts possible within a given time span. It is possible that this has been improved in Bluetooth version 2, though no information about this has been found. Either way, most mobile devices that are common today are using Bluetooth version 1.x.

What if the Bluetooth broadcast remains passive, and instead requires the user to search for it to request information? Unfortunately, when many users search for Bluetooth devices at nearly the same time (in the same place) this will also affect the discovery. Peterson [18] has in a study found that it is possible that with many users, discovery would be delayed for some users. In the worst case scenario, they would not be able to find the Bluetooth broadcast at the first try at all. This risk increases with the number of people using device discovery at the same time.

While the transfer speed might not be such a big problem, as long as the files sent are of reasonable size, the discovery problem could be a problem if it is really serious. What about other technologies? The next subchapters will look at other technologies and discuss whether or not

they can be used for broadcasting to mobile devices in the same way as with the Bluetooth technology.

7.2 IrDA (Infrared Data Association)

IrDA is an association that has defined a standard for using infrared light for short range communication between devices [19]. Infrared was the first wireless transfer technology that was implemented in mobile phone devices. Since light is used for communication, a direct line of sight is required between devices (or rather, between the devices' infrared port) for data transfer to work.

The IrDA specification defines the communication range as smaller than 1 meter, although some devices may have longer range. The transfer rate varies from 2.4 kbps to 16 Mbps, making it possibly a lot faster than Bluetooth. Still, the short range, the drawback of needing a direct line of sight, and the fact that it can only transfer to one client at a time, makes it rather inappropriate for broadcasting.

7.3 Wi-Fi

Wi-Fi is the name for a collection of standards defined by the Wi-Fi alliance [20]. The standards are defined for use in a local area network (LAN), commonly used by personal computers. It is based on the IEEE 802.11 specifications, which is the only specification used for Wi-Fi for now, although new ones are under development.



Illustration 17: Official Wi-Fi logo

There are different versions of the 802.11 specification [21], and the currently newest version, 802.11g, supports transfer speeds up to 54 Mbps and a range of about 30 metres, while being backward compatible with older versions. Some companies claim that their products support higher speeds, though those products use proprietary extensions of the specification that are not part of the standards. The next version of the specification, 802.11n, will theoretically support a transfer rate of up to 540 Mbps.

While Wi-Fi has longer range and has higher transfer rate than Bluetooth, it also consumes more power and is more expensive to implement, and is therefore not as common on portable devices.

There are a wide variety of services that allow for transferring files between two devices over a Wi-Fi connection, but most of them require communication between specific software, and no standards exists for just sending files over a Wi-Fi connection.

7.4 RFID (Radio Frequency Identification)

RFID is a standard for identification of objects with RFID tags. RFID tags are small microchips with antennas, containing identification information about the object it is attached to (or just an identification code, leaving the information to be found in external databases). Typical RFID tags usually can not store more than 2KB of data [22], limiting the amount of information that they can send.

The range of RFID tags range from just a few millimeters to several meters using passive tags (tags without their own energy source), to 100 meters or more with active tags (tags that are powered by an energy source). The range also depends on other factors, like the RFID reader used and interference. The transfer rate is also highly varying, depending on the implementation, frequency used, whether active tags or passive tags are used, and possibly other factors.

Very few consumer devices have built in RFID support. Nokia has recently released some mobile phones with NFC (Near Field Communication), which is a standard based on RFID. The NFC standard was developed by Sony and Philips with focus on user privacy, in order to avoid the bad reputation that the RFID technology has gotten. Since NFC is defined to be short range, it is hardly useful for broadcasting.

If RFID becomes common, it may be somewhat possible to broadcast information using this technology. If active RFID tags are used, they may get enough range that users can get information without having to put the phone right next to the tag, assuming that the RFID readers on the phone also are powerful enough to pick up the signals. The limitation of storage space could be circumvented by creating tags that have more space. The main reason that RFID tags

generally are made with as little storage and processing power as possible is that it should be cheaper to mass-produce them. If used for broadcasting, however, the number of tags does not need to be as high, making it possible to use more expensive tags. This is theoretic though, and using RFID in public situations is hardly an issue before devices with RFID readers start getting common, which may not be very soon, with the technology being as controversial as it is today.

7.5 Mobile phone network

The mobile phone network is the wireless network that mobile phones connect to for phone functions like calls and messages. There are several types of networks, with the most popular one being GSM (Global System for Mobile communications). GSM is a second generation (2G) type of mobile phone network, meaning that the radio signals used are digital instead of analog.

There are a few ways that the mobile phone network can be used for broadcasting messages. One way is to use the SMS (Short Message Service) or MMS (Multimedia Messaging Service) services to broadcast messages, and this is already used for example by the mobile phone companies to send information or campaign messages to all their customers. SMS is the service which allows users to send short text messages to each other's mobile phones. MMS is a similar service, but allows sending images, audio clips and video clips in addition to text.

The advantage of using SMS to send messages is that practically all mobile phones today have at least SMS support. Messages are therefore bound to be received correctly. Broadcasting MMS messages has also gotten more common with MMS enabled phones getting more popular. The drawback for using these services for broadcasting is that they cost money for each recipient of a message, making it quite costly if messages are regularly sent to a large amount of people. The messages also cannot target an area, they have to be targeting a known group of users.

Another way to use the mobile phone network for broadcasting is a technology called cell broadcasting [23], which is a function of mobile phone network that is rarely used. This technology allows for broadcasting text messages to all mobile phones connected to the broadcasting cell. However, in many countries there are some controversy over whether this type of broadcasting will be used at all. Even if it is, it will most likely be reserved for government use

in emergencies. That aside, this technology is also of a quite larger scale than Bluetooth broadcasting, where we are talking about covering geographical areas rather than covering one or a few rooms. All in all, cell broadcasting is not very relevant to this project.

One could also broadcast things over a GPRS connection. GPRS (General Packet Radio Service) is a service that allows mobile phones users to access various over the GSM network [24]. GPRS acts as a gateway to the Internet, allowing users to download web pages or content specifically made for mobile devices. The transfer rates are usually around 30–80 kbps, significantly lower than Bluetooth. This way of broadcasting would also not be able to support the “area broadcasting” that Bluetooth supports, and is also not cost free to use for the users. (Usually the costs of accessing GPRS are calculated per kilobyte transferred.)

7.6 ZigBee

ZigBee [25] is a specification for wireless communication based on the IEEE 802.15.4 standard for wireless personal area networks. It is designed for small size and low power consumption, and is described as using even

less power than Bluetooth, although it also has lower transmission speeds than Bluetooth.

ZigBee uses several radio bands for wireless communication: 2.4 GHz, 915 MHz and 868 Mhz, and the maximum speed is different on each band, with the highest speed being 250 kbps in the 2.4 GHz band. The range is between 10 and 75 meters. [26]

The ZigBee technology is designed mainly for monitoring and control purposes, for example in building automation. It supports a significantly higher amount of simultaneous connections, 65 000 or higher according to their website. A connected node can go into a sleeping mode, and can go into active mode for transmissions before going back to sleep. This is to lower the power consumption without requiring a reconnection every time a data transfer is needed, and allows for the large number of simultaneous connections that the ZigBee specification supports, without using up all available bandwidth.



This technology is not really meant for transferring more than a minor amount of data between nodes/devices, and is therefore not really meant for broadcasting in the sense this project is aiming for. Devices using this technology are also mostly specialized devices used for the purposes that were mentioned earlier: automation, monitoring and control. Like with RFID, there are no common user devices that implement this technology. ZigBee is therefore not quite relevant in this context.

7.7 Comparison chart

<i>Technology</i>	<i>Transfer rate</i>	<i>Range</i>	<i>Availability</i>
Bluetooth	v1.x: 723.1kbps v2.0: 2.1 Mbps	Usually around 10 meters	Many handheld devices, and also computers
IrDA	2.4 kbps - 16 Mbps	Around 1 meter	Available on some phones and computers, though less common after Bluetooth was introduced.
Wi-Fi	Up to 54 Mbps	Up to 30 meters	Mostly used on computers. Some PDAs and smart phones can also use Wi-Fi.
RFID	Varying	Passive tags: Up to a few meters. Active tags: Several hundred meters. NFC: A few centimeters.	Mostly used in industry/business environments, though some mobile phone companies have recently started making NFC enabled phones.
SMS/MMS	(Not applicable)	Wherever there is mobile phone coverage	SMS is supported by practically all mobile phones now. MMS is supported on most newer phones.
Cell broadcasting	(Not applicable)	Wherever there is mobile phone coverage	Should be supported by all mobile phones.
GPRS	30 - 80 kbps	Wherever there is mobile phone coverage	Supported by most newer phone models.
ZigBee	Up to 250 kbps	10-75 meters	Mostly used for monitoring purposes like building automation.

Table 1: Wireless technologies comparison chart

Table 1 lists the technologies that have been described in this chapter and some of their specifications that are relevant to this project.

7.8 Thoughts about technologies

While some of the other technologies that were described also can be used for broadcasting, their range, cost and availability makes broadcasting with these technologies quite different from when broadcasting with Bluetooth. Therefore, Bluetooth broadcasting is quite an unique way of broadcasting.

In “7.1 Bluetooth”, it was mentioned that Bluetooth has a few flaws when it comes to broadcasting. One is the limitations on multiple transfers: a Bluetooth radio can only send to 7 people at the same time, and the transfer rate might be too low for it to be practical to send larger files to multiple people at the same time. The other is the risk of the Bluetooth station not being discovered if a lot of users are doing device discovery at the same time.

These two flaws would mean that using Bluetooth broadcasting targeting large audiences might be difficult. This is however quite hypothetical, and it could be possible that in practical use, these flaws will not affect the performance as much as feared. For example, as long as the files broadcasted are of reasonable size and do not take too long to send, user connections will not last very long. This would leave room for quite a large number of downloads per time unit, even with only 7 connections at a time. In any case, using a Bluetooth broadcast in a situation where it is not expected to be many simultaneous users should not be a problem.

8 User privacy

In research question 3 in “2.1 Research questions“, a concern was raised regarding whether Bluetooth would become a spam-infested communication channel like e-mail has become.

Imagine this scenario:

A group of friends are sitting in a café in a shopping mall. One of them decides to send his friends a funny picture he has taken earlier, and his friends start setting their Bluetooth settings to “Discoverable” on their mobile phones. Right after they do that, they start getting incoming files, not from their friend, but from a dozen Bluetooth broadcasting advertises. Even after denying one request, another one pops up right afterwards. In the end they end up not being able to send anything to each other, having to turn off Bluetooth on their phones.

There are a few notes to this scenario. One is the fact that it is possible to avoid this, if the friends had stored each others' Bluetooth IDs beforehand, so that they can send stuff to each other without setting their phones to discoverable. Another note is that there might not be as many anonymous advertisements sent via Bluetooth as on e-mail, and if known brands and stores set up broadcasts, users could easily go and complain. Last thing is that this example is quite exaggerated, and should hopefully never become reality, even if automatic sending Bluetooth broadcasts become common. Still, it is theoretically possible that a situation like that would occur.

In any case, users might not like broadcasting systems which automatically sends messages or files to nearby discoverable Bluetooth devices. Even if incoming transmissions has to be accepted by the users before they are received, a situation like the one described in the example scenario would certainly be annoying.

On the topic of unwelcome Bluetooth connections: A phenomenon which has a grown small community of followers on the Internet is Bluejacking. This is a term for sending contacts or messages to unknown people who have set up their phones to be discoverable. The communities do stress the point of not sending anything harassing or offensive, though, and they also have some amusing example stories sent in from the members. Another not so innocent term related to

Part II: Prestudy

Bluetooth is called Bluesnarfing, which denotes stealing information from a device via the Bluetooth connection without the user of that device knowing. This is done by exploiting bugs in the Bluetooth implementations on some mobile phones [27]. Bluesnarfing is illegal in many countries since it breaches user privacy.

It would seem like leaving Bluetooth devices on discoverable by default may be a security risk. Is it then justifiable to ask users to accept incoming requests from a Bluetooth broadcast, or even ask them to have their devices discoverable in order to receive broadcasts? And even if the users do that, should a broadcast try and send information to any Bluetooth device within range which has Bluetooth enabled? If not, what are the options?

Another issue is the logging of broadcasting activity. A few of the projects mentioned in “5 Existing projects” mentioned that their systems could remember the user activity and use it to customize user experience, something that is quite possible because of the unique Bluetooth ID each device has. While some people will not find this alarming at all, there are some people who are skeptical to systems which can log user activity like this. Adding in the fact that it is possible to log the location of each user by logging which broadcasting station they are accessing (although none of the projects states that they do this), these people may not like the thought of being logged by Bluetooth broadcasts.

In a broadcast that automatically sends requests to users within range, it is obvious that it needs at least some form of logging users so it does not try sending the same thing several times to the same user. But how much logging is it reasonable to do?

8.1 Alternatives to automatic sending

To avoid users feeling irritated by incoming requests, one could make the broadcasts passive. Instead of the broadcasting station continuously searching for and sending out information, the users should be the one to take initiative. When a user sends a text message to a broadcasting station, it would then respond by sending out the information that is requested by that text message. Some of the projects that were described in Chapter 5 used this method, although most of them used this in addition to automatic sending.

Another way to do this is to only broadcast to users with a specific prefix or suffix in their device names, and inform about this on posters near the broadcast. This is a cumbersome solution, however, and users might not want to change their Bluetooth names just for receiving broadcasts.

A third way would be to require users to register for the broadcast before he can receive it. Since there is a unique Bluetooth ID for each Bluetooth device, users could register their ID and phone name on for example a website, or just by sending a registration message to the broadcast. When the broadcasting station discovers a Bluetooth device, it could then look in the database to see if the device ID is registered with the correct phone name, before sending anything to the device.

The different alternatives will be discussed in the next chapter.

9 Functionality choices

The report so far has described several solutions for different parts of the functionality of a Bluetooth broadcasting system. The following chapter will discuss the different solutions and the advantages and drawbacks of each of them, and select the solutions that will be used for developing the test system.

9.1 Custom user client

Requiring the users to install a user client to accept broadcasts has the advantage of making the system more predictable. However, it will make the system less user friendly, and not all users are acquainted with installing software on their mobile devices, more specifically mobile phones (of those using PDAs, the percentage of people who knows how to install software is probably higher). So is it best to require users to use a custom user client, or is it best to avoid this?

Most likely it is best not to depend on a custom user client for a broadcast. In addition to the issue already mentioned, another problem is that there are quite a vast amount of models for mobile phones and PDAs. Making sure that there is a version of the user client that works for each device will be quite difficult, with new models being released every month.

9.2 On demand versus automatic sending

Should the broadcasts automatically send information to users within range, or should they be passive, requiring users to request information before it is sent? Or should it be a combination, where basic information is sent first, and letting users then request more information as they wish?

For people who are well acquainted with using Bluetooth on their mobile phones, having to send requests to a Bluetooth station should not prove much of a problem, and may be preferable for them. For people who are less tech-savvy, however, would having to request information be too much of a hassle? Since most of the people with mobile devices would know how to send text messages, it may not be that much more difficult to send Bluetooth messages for requesting information. But it would still be a bit more troublesome than if the information was sent more or less automatically, so two questions would be:

1. How important is it for the broadcaster that as many people as possible should get the information that is broadcasted?
2. Would enough users be interested in using broadcasts at all if they are not automatic? Will it be too inconvenient?

To answer question 1: With what we have discovered so far in this project, it is quite obvious that the Bluetooth technology as it is, is not a good medium for broadcasting important messages. The main issue is that the Bluetooth technology has a few flaws that makes it work inefficiently when broadcasting to a large number of people. Another thing is the fact that even though Bluetooth is getting increasingly common, not everyone has Bluetooth devices yet. So even if Bluetooth broadcasting is used as a supplement to provide important information, it is hardly a reason in itself to push information to every device within range.

For question 2, the answer can quite possibly be answered in the same way as any other technology: It depends on whether or not any good services are created using this technology. Of course, since it is free for the users to receive information via Bluetooth, that alone will probably help immensely on getting interest from people in the start phase. But whether it gets popular or not still depends on the first answer we gave. The question is then: will requiring users to request information be inconvenient enough for them to not use even good services? This question cannot be answered just by logical reasoning. In general, however, it is probably best to avoid automatic sending, in order to avoid getting a scenario like the one described in “8 User privacy“.

9.3 Two-way communication

Should the Bluetooth broadcasting station also be able to receive files? This is a possibility, and could be quite useful for a broadcast that is for internal use, like in a company or university. The files received could be added to the broadcast, allowing other users to access it, or the file could be forwarded to a file server or e-mail address. For public broadcasts it could be used for things like surveys or other type of user feedback, or for example photo competitions where the users can send in their photos.

While this is possible, inputting text on handheld devices in general is not quite as effective as on computers. It is therefore not likely that people would write large amounts of text in reply to a Bluetooth broadcast (although it would be possible with a laptop with Bluetooth). Received data would therefore more likely be previously created data, or media that are easily recorded on handheld devices like photos, audio and video clips.

In any case, even if not every broadcast would require two-way communication, leaving at least the option available would be a good thing.

9.4 Logging

When it comes to the question about how much logging of user information there should be, the answer would heavily depend on the target for the broadcast, and the type of broadcast. If the broadcast is passive, and the users are the ones who request information, then it is not really necessary for the broadcasting application to store any information about the transmissions at all. And if the broadcast is used for sharing information internally in a company, then logging is obviously not an issue at all, and may even be necessary for security reasons.

For a public broadcast that is automatic, however, it should optimally not store more information than what is needed to avoid repeated sends to one person. This information should be deleted within reasonable time, and should not need to be accessible for other than the system itself.

Logging the traffic of the Bluetooth broadcast without connecting it to user information might be more acceptable. A log like this can be useful in discovering things like at what times the broadcast is most commonly accessed, and what services are most popular. A problem would be if it is desired to log unique requests, so the broadcaster can see how many different people have accessed the broadcast. For this it would be necessary to log the Bluetooth ID. But for how long should this be stored? Just for an hour, to get unique visits per hour? Or store it for a day, to get unique visits per day? How long is it justifiable to store it, if at all? To address this problem, one solution would be to store an encrypted version of the Bluetooth ID instead of the ID itself. That way it will be harder to connect the Bluetooth ID to the device it is from. Eventually this hash can be deleted at the end of the day, after the number of unique visits has been calculated.

9.5 File types

Which file types are the most useful to send via a Bluetooth broadcast? In most cases, this will probably be the file types that most people will be able to show immediately on their devices. The file types that are most certain to be viewed properly would be text, contact cards and calendar events. Images should also be supported by a majority of Bluetooth devices, though Palm devices without an image viewer will not be able to receive images.

However, it might not be necessary to limit the file types that it is possible to send in a broadcasting system. The documentation or broadcasting system should probably still show information about how well supported each file type is. Other than that, allowing for the broadcaster to determine what file types to broadcast would allow them to use the system for more specialized purposes.

The solution mentioned in “6.4 File types” about determining what files to send by recognizing the handheld device, could also be a possibility. However, doing it this way requires using a central database which has to be updated with every new model released, and the data needs to be gotten from all (or at least all the major) producers of Bluetooth devices. This is therefore not possible at least for this project, though it is a valid solution which ensures that users will not have to spend time on downloading content that they can not view.

9.6 Summary

The following is a summary of the choices that were made in this chapter:

- The Bluetooth broadcasting system should not depend on a custom user client, but should instead use the standard Bluetooth file sending functionality.
- Broadcasts should generally wait for user requests before sending information.
- Two-way communication is not strictly necessary, but could be useful in certain situations, and should therefore be supported (if possible).
- Logging of user information should be avoided. Logging of traffic can, however, be quite useful.

Part II: Prestudy

- The broadcasting application should not need to limit the file types that can be delivered, but it should still inform those who set up the broadcasts about which file types are more commonly viewable on different types of devices.

While these rules are not necessarily the best ones in every situation, they are the ones that seems to be best in general when taking user privacy into account, and to some degree user friendliness. The prototype that is going to be made in this project will therefore follow these choices. In the next part, “Requirements specification”, a set of requirements will be described for a prototype broadcasting system based on these choices.

Part III: Requirements specifications

10 Introduction

This part of the document describes the functional requirements that are proposed for the prototype broadcasting application that will be developed in this project. The name of the broadcasting application will be Baloo, and this is the name that will be used in the rest of this report when referring to the test broadcasting system.

Since this is a prototype application, the most important aspects of the application are:

- **Functionality:** The prototype should do what it is meant to do. However, it should not need to do much more than that, as new functionality can rather be added in any further development of the application in future projects.
- **User-friendliness:** It should be somewhat easy for people to test the application by setting up their own broadcasts. This will make it easier to raise interest.
- **Stability:** While not the most important aspect of a prototype, Baloo should still be stable enough not to crash very easily.
- **Finishing before the deadline:** The deadline for this project is immutable, therefore it is quite important for development of this application to not take longer time scheduled.

The functional requirements are based on the choices made in the conclusions of part II: Prestudy, and is a list of functions that we expect the final product to have. The quality requirements are based on the functional requirements and describes how we expect different parts of the functionality to behave in specific situations. Both lists of requirements will be used as guideline for development and testing.

11 Functional requirements

FR 1 Baloo should be able to deliver any type of information that the broadcaster wants to send.

FR 1.1 The information should possibly be dynamic, for example time of day could be accounted for when sending requested information.

FR 1.2 Receiving the information should not require a custom user client, but use the standard file receiving functionality of the Bluetooth devices.

FR 2 Passive broadcasting: Baloo should not send anything until the user requests it.

FR 2.1 There should either be a command for listing what information is available for requesting, or such a listing can be sent when a user pairs his/hers device with the broadcast.

FR 2.2 Commands should be sent as text messages to the broadcasting station. Baloo should then reply by sending the requested resource, or if the command is not recognized, it should send an error message and the list of available resources.

FR 3 Receiving files: It should be possible to set up Baloo to receive files, and treat these in various ways.

FR 3.1 It should be possible to enable and disable this functionality, and the functionality should not interfere with receiving commands/information requests.

FR 4 Logging: Since we are going for a completely passive system, no logging of user information is necessary. Baloo should still log information about traffic, so the results can be used for test results. The logging system should:

FR 4.1 Log each request with time of the request, which resource was requested, plus a hash of the Bluetooth ID in order to calculate unique requests.

FR 4.2 Delete the hash at the end of the day, after calculating unique visits that day and per hour.

FR 5 Baloo should be able to use several Bluetooth radios (if more than one is available on the computer running Baloo)

FR 6 There should be a GUI (graphical user interface) for setting up Baloo. The interface should allow the broadcaster to:

FR 6.1 add, edit and remove commands for requesting information, and the resources they are connected to.

Part III: Requirements specifications

FR 6.2 edit the list of commands that is sent to the users. (This way some commands can be “hidden”.)

FR 6.3 enable/disable receiving files, and make it possible to choose what to do with received files.

FR 6.4 limit specific commands to registered users, or describe how to do the limiting.

12 Quality requirements

Based on the aspects described in the introduction and on the functional requirements, these quality requirements were made:

QR 1 Testability

QR 1.1 Baloo should always reply to a information request.

QR 1.1.1 If a recognized command is received by Baloo, the associated resource should be sent to the user right away.

Measure: From the user sends resource request an until he receives it, the time passed should be less than 2 minutes (although larger files like graphics and audio will take longer, at least for text it should not take longer time than that).

QR 1.1.2 If an unrecognized command is received by Baloo, an error message should be sent stating that the command was not recognized. The error message should also contain a list of available resources and their commands.

Measure: The error message should be sent in 100% of the cases where an unrecognized command is received.

QR 1.2 Baloo should log each transmission and incoming requests. This way if Baloo crashes because of a transmission, it should be possible to find out what command caused the crash.

Measure: 100% of the transmissions should be logged.

QR 2 Usability

QR 2.1 Adding or editing resources and their associated commands should not take longer than the time it takes to type the information needed for each resource.

Measure: The time it takes for adding or editing one resource (assuming that one already knows the information needed for the resource) should not exceed 2 minutes.

QR 2.2 Deleting, enabling or disabling resources should not take longer than the time it takes to select the resource and clicking on the relevant button.

Measure: The total time for doing one of these actions should not exceed 10 seconds.

QR 3 Availability

QR 3.1 Under normal usage, Baloo should not crash.

Measure: Baloo should be able to run for at least 1 hours without crashing.

Part III: Requirements specifications

QR 3.2 If the resource that a command is connected to is not available for any reason, an error message stating this should be sent when this resource is requested.

Measure: The error message should be sent in 100% of the cases where the resources requested are not available.

QR 4 Modifiability

QR 4.1 The code for Baloo should be well organized so it is easier to locate parts of the code which needs to be changed.

Measure: Finding a specific part of the code should not exceed 5 seconds.

Part IV: Architecture and design

13 Introduction

This part of the document describes the architecture of the prototype application developed in this project. The architecture serves two purposes:

1. It is planned in order to be a guideline for the implementation. A well planned architecture will:
 - make implementation easier, making a quick completion more likely.
 - ensure a stable and consistent application.
2. It is updated during the implementation to ensure that the architecture described reflects the actual architecture in the developed application, so that it can be used as a reference for maintenance or improvements in any future projects. The changes and why they were done are described in the Changes chapter.

The main architecture driver for the prototype application will be “Testability”. The reason is that the purpose of the prototype application is to test it against the hypotheses in this project, in order to see if our answers to the research questions hold.

The language that will be used for developing Baloo is C#, using the .NET framework. The library 32feet.NET [28] will be used for the Bluetooth connectivity. The requirements for the computers running Baloo will thus be:

- Windows XP with SP2 installed (SP2 is required for Microsoft Bluetooth Stack)
- .NET framework version 2.0
- Bluetooth radio, using Microsoft's Bluetooth Stack drivers (the drivers that usually install automatically when SP2 is installed, without needing a CD)

14 Stakeholders and concerns

Since the application to be developed is a prototype, there are not really that many stakeholders involved in this project. The main stakeholders are:

- **The developer and acquirer:** these two are basically the same people in this case. Their concerns are:
 - Having a guideline for implementation of the application.
 - Getting a stable and consistent end result.
 - Finishing implementation within the time frame given.
- **Testers and users:** these are also the same as the developers and acquirer, but also any eventual external people that will do testing. Their concerns are:
 - Knowing how to use the system, and what the expected results are.
- **The maintainers:** anyone who eventually works on this application in the future. Their concerns are:
 - Being able to locate what parts of the code to edit in order to add the changes they want.
 - Knowing what exactly each part of the code works.

15 Selection of architectural viewpoints

There are a few different viewpoints from which one can describe an architecture. Some of the relevant ones from “Architectural Blueprints—The '4+1' View Model of Software Architecture” [29] were chosen to be included in this project:

- **Scenarios:**

Shows a few examples of how the application is expected to behave externally, in other words how a user interacts with it and the expected results of the interaction. This viewpoint is useful for testers and users, and can be used in addition to other documentation of the application.

- **The logical view:**

Describes the logical architecture of the application, in other words how it is divided up into parts and how each part interacts with each other. The method used for describing this viewpoint will be UML class diagrams and class descriptions, and this viewpoint is mainly meant for developers and maintainers.

- **The process view:**

Describes how the engine of the application is expected to work, showing a chronological order of events. The method used here will be UML activity diagrams, and this viewpoint is also meant for developers and maintainers, although testers and users may also be interested in this viewpoint to further their understanding of the application.

16 Quality tactics

The following is a listing of the tactics that will be used in order to ensure some quality attributes that are important to this project.

QT 1 Testability

QT 1.1 Baloo should save the following information in a SQL database for generating statistics:

- time of incoming request
- command used
- the reply sent
- hash of Bluetooth ID to calculate unique users

QT 1.2 Baloo should log the following events to a plain text file:

- incoming request
- fetching and preparing requested information
- sending requested information
- saving the transmission data to the database

The difference between the log file and the statistics database, is that the log file contains more technical information used for debugging, while the statistics only show the events, and no more details than necessary.

QT 1.3 Every information request should be replied with some type of feedback.

QT 1.3.1 If the command in the request is recognized, the relevant resource should be sent to the user who sent the command.

QT 1.3.2 If the command is recognized, but the resource is unavailable, an error message stating this, eventually also the reason for the unavailability, should be sent to the user.

QT 1.3.3 If the command is not recognized, an error message should be sent, with a list of available resources and the commands for requesting them.

QT 2 Maintainability

QT 2.1 Dynamic content will be provided by an URL.

Instead of making a scripting engine or a complex schema for making dynamic content in the prototype application, it has been decided to rely on using a web server's PHP scripting engine for dynamic content. This should heavily reduce the

time it takes to develop the prototype application, although this will probably affect performance negatively. By using a local web server on the same computer that is running the broadcast, the performance should, however, not be affected much.

QT 2.2 The code should be well structured:

QT 2.2.1 Code should be put in functions in a logical way even if that particular piece of code is only used once, to avoid code cluttering.

QT 2.2.2 Functions and long code sequences should be well formatted and commented for readability.

QT 2.2.3 Code that depends on the same library/libraries should be placed together in the same code file(although this file can be divided if it is so big that it is turning overly complex).

QT 3 Usability

The GUI descriptions that follow are shown as lists instead of sketches of the GUI. This is because the GUI will be designed using the graphical user interface of Microsoft Visual C#, making it somewhat unnecessary to make a draft. To see what the GUI looks like, read “Part V: Implementation”.

QT 3.1 Baloo should have a user friendly GUI for setting up a broadcast. The GUI should be divided into three parts which can be switched between using a tabbed panel:

QT 3.1.1 Main configuration.

QT 3.1.2 Resources management

QT 3.1.3 Statistics

QT 3.2 Certain elements should always be shown:

QT 3.2.1 **A title** of Baloo.

QT 3.2.2 **Button(s) for starting/stopping the broadcast**, and eventually text showing the status (unless the button is designed to show this).

QT 3.2.3 **Buttons for saving and loading the configuration** to/from a file that the user chooses, using default Windows file dialogs.

QT 3.2.4 **A button for exiting Baloo**, placed a bit away from other buttons.

QT 3.3 The main configuration part should have the following elements:

QT 3.3.1 **A text field for inputting name of broadcast**, which should be the name that users see when searching for Bluetooth devices. (Note: It turns out this is not usable, since the 32feet.NET API is actually not able change the broadcasted name.)

QT 3.3.2 **A text field for writing feedback messages, and a combo box to select which message to be edited**. These messages are sent to the user in case of errors or other events.

QT 3.3.3 **Radio buttons for choosing what to do with received non-text files**.

(Received text should be interpreted as commands.) Options can be:

QT 3.3.3.1 Deny the files (delete them)

QT 3.3.3.2 Store in a specified folder.

QT 3.3.3.3 Forward to a predefined e-mail address. (This option is left out for now, because it is not crucial for Baloo, and does not help testing.)

QT 3.3.4 **Text field for filtering received file types** so that certain file types will not be stored.

QT 3.3.5 **A text field for setting the debug log file** that the logging should be saved to, using a Windows file dialog for selecting a file, and creating the file if it does not exist. (Eventually creating it once it starts logging.)

QT 3.3.6 **Text fields for inputting data about the MySQL database** which is used to store statistics:

QT 3.3.6.1 Server address

QT 3.3.6.2 User name

QT 3.3.6.3 Password (plain text)

QT 3.3.6.4 The MySQL database to use. The tables in the database should be created by Baloo if they are not created already.

QT 3.3.7 **A button for showing the statistics screen**.

QT 3.3.8 **A text area showing the system log**. This text does not need to be removed when the broadcast is stopped.

QT 3.3.9 **A list showing the currently running user threads**, to make it easier to see how many users that are currently connected to the broadcast.

QT 3.4 The resources managing part should have these elements:

QT 3.4.1 **A button for adding a new resource.**

QT 3.4.2 **A list with the resources that have been added**, the user should be able to click on this list to select resources.

QT 3.4.3 **A text field for writing a name** of the resource

QT 3.4.4 **A text field for writing the command** for requesting this resource.

QT 3.4.5 **A text field for writing a description** (to be used in the command listing sent to users)

QT 3.4.6 **A check box for enabling/disabling the resource.**

QT 3.4.7 **A check box for selecting whether this resource should be listed or not** in the list of commands that is sent to the user.

QT 3.4.8 **Four radio buttons for selecting type of resource:**

QT 3.4.8.1 **List of commands**

QT 3.4.8.2 **A specific file** (chosen by using Windows file dialog)

QT 3.4.8.3 **A specified folder** of files. When only a command for the folder is sent, the folder listing is sent. The files can then be fetched with command + filename.

QT 3.4.8.4 **An URL** (for dynamic content)

QT 3.4.9 **A button for deleting** the resource. The button should show a confirmation dialog asking if the user really wants to delete the resource.

QT 3.5 The statistics part should contain the following elements:

QT 3.5.1 **Buttons for selecting hourly, daily or total statistics.**

QT 3.5.2 **The statistics** based on the selected option (hourly, daily, total). For each hour/day or in total, show:

- number of requests
- number of unique requests
- each resource and the number of times they have been requested, ordered by the number of requests

QT 3.5.3 **Button for saving the generated statistics.** (This was left out due to lack of time.)

QT 3.5.4 **Button for clearing statistics older than a given amount of days** from the database.

17 Architectural pattern

The architectural pattern that will be used for Baloo, is the “Layered architectural pattern”. This pattern fits very well with the way that Baloo will work.

Bluetooth layer: handles connections with user devices
Resources managing layer: This layer handles the resources, and includes a GUI for mapping resources to commands and a class for storing resource data. It also tells the Bluetooth layer what to return to the user.
Resources: The resources that are returned to the user. There are two types: files and web content. If the requested resource is an URL, the web server handles the request and returns the result, otherwise a file is returned.
PHP engine: The PHP engine is used for generating dynamic content for the web server, when this is needed.

Table 2: Baloo's layered architecture (the bold layers are the layers that are within Baloo)

Table 2 shows the layers that Baloo will have. The layers that are shown in bold are the ones that are within the Baloo application. The other two layers are the external layers for resources which will be used in this project.

This pattern is chosen of two reasons:

1. The Bluetooth class needs to be constantly running in a thread in order to receive incoming requests, while the resources handling class only needs to be called once an incoming request has been received.
2. The Bluetooth handling is separated from the handling of resources, making it easier to eventually upgrade the Bluetooth handling routines without affecting resource handling (or vice versa – upgrading the resource handling classes without affecting Bluetooth handling).

This pattern is therefore the pattern that will be used in the development of the test system.

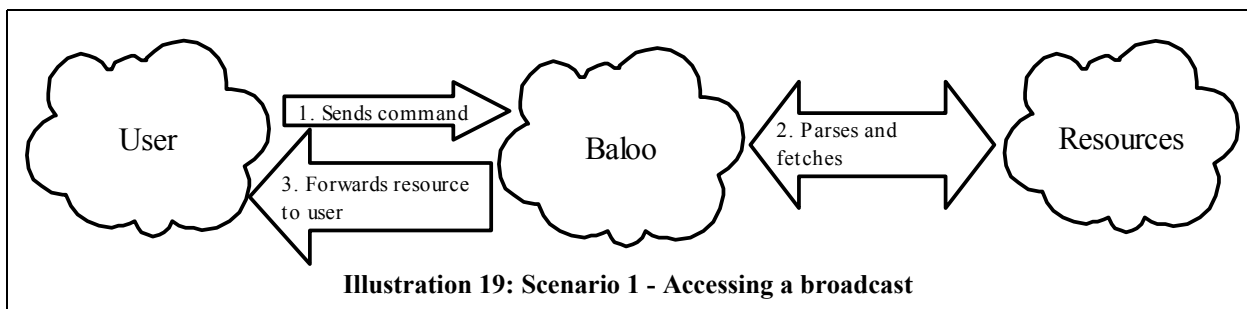
18 Views

This chapter describes the architecture from each of the viewpoints selected in “15 Selection of architectural viewpoints”.

18.1 Scenario view

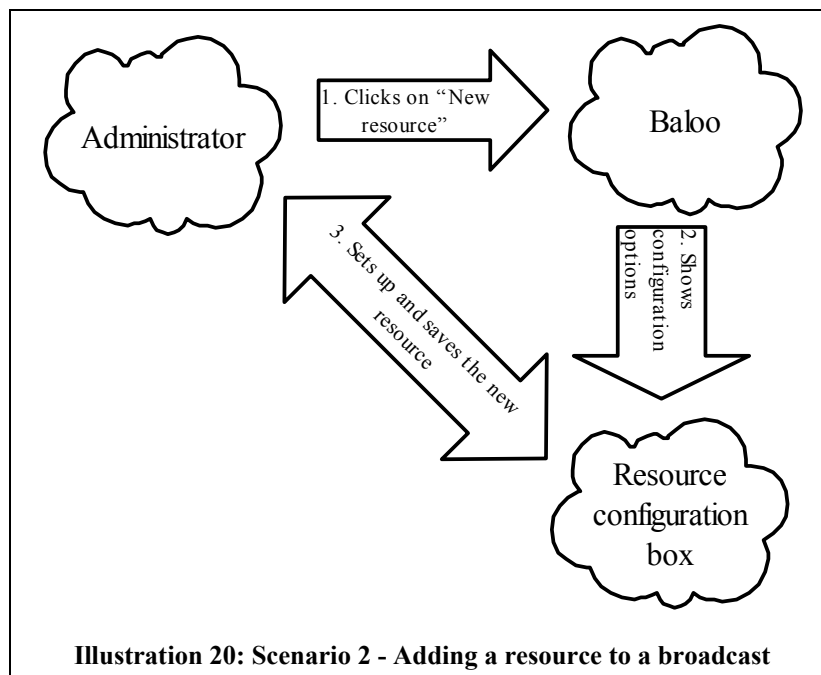
This subchapter describes a few example scenarios for how the broadcasting application should be used.

18.1.1 Accessing a broadcast



This scenario shows how Baloo acts as a mediator between the users and the resources, allowing users to access resources through Baloo. The user in this case is a person accessing the broadcast.

18.1.2 Adding a resource to a broadcast



When add new resource is clicked, a box containing the options that can be set for a resource will be shown (Illustration 20). This box is hidden when no resources are selected. The changes made to a resource is saved automatically. Note that “Administrator” in this illustration is the person in charge of setting up the broadcast. This term is used instead of “User” to avoid confusion with the “User” mentioned in the previous scenario.

18.2 Logical view

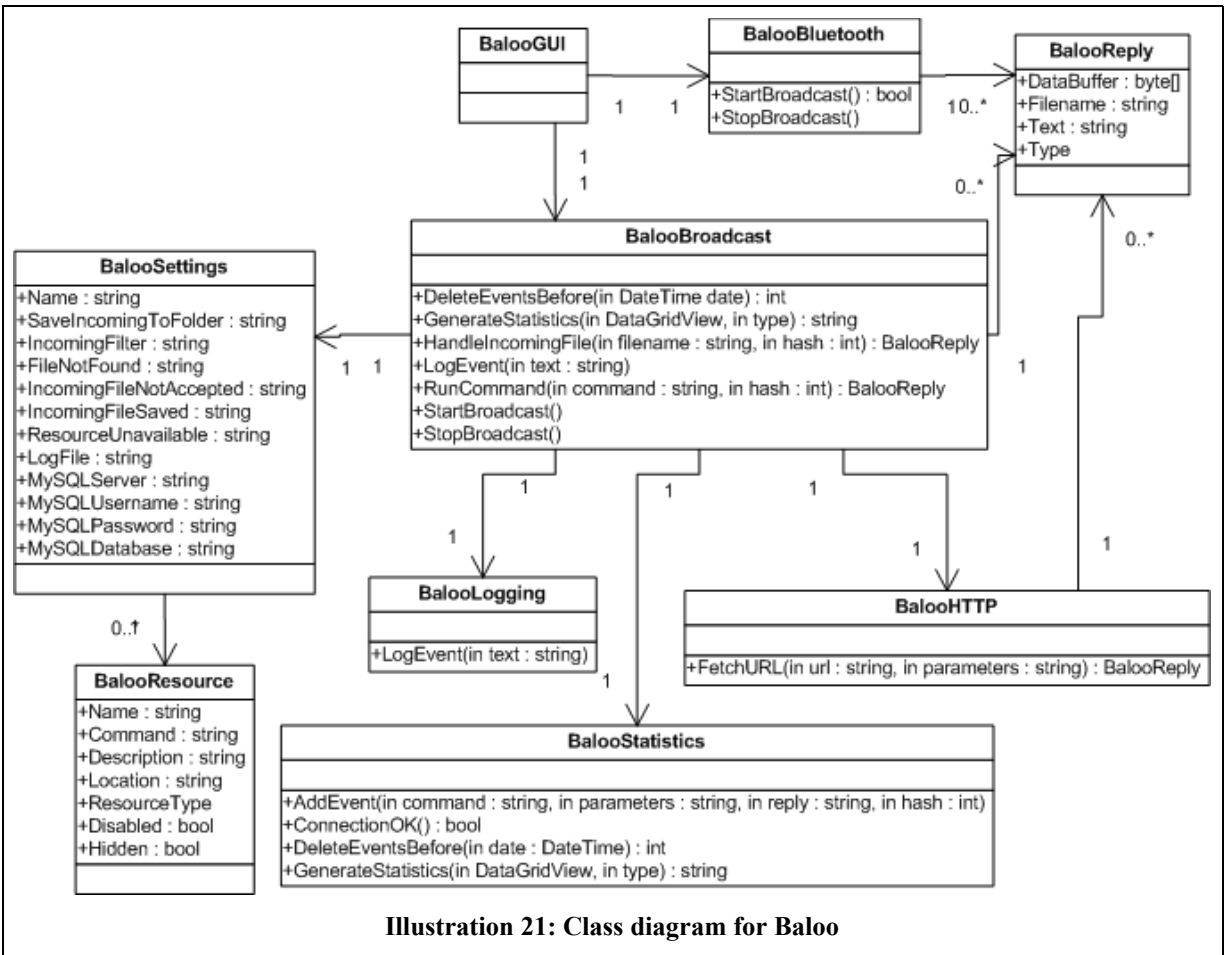


Illustration 21 shows the planned classes for Baloo, and how they interact with each other. Most of the public functions and properties/variables are shown. (Helper variables like constants are not shown.)

With the exception of BalooResource and BalooReply, all classes are intended to only have one instance. BalooGUI should be the main class which is started when Baloo is started, and the other classes are initialized by BalooGUI either directly or indirectly (indirectly: when the classes that

are initialized by BalooGUI, initializes their own children classes). For BalooResource, a new instance of the class is created for each new resource added. An instance of BalooBluetooth should be initialized when starting the broadcast, and that instance should be deleted when the broadcast is stopped.

In accordance with the “Layered architectural pattern” that was chosen as the pattern for Baloo, there are two main layers of the application. The classes shown in Illustration 21 are divided into the two Baloo layers like this:

- **Bluetooth layer:**
 - **BalooBluetooth:** Class for listening to incoming requests and returning replies.
- **Resources managing layer**
 - **BalooGUI:** The GUI for setting up, starting and stopping the broadcast.
 - **BalooResource:** Class for storing data about each resource.
 - **BalooSettings:** Stores the settings for the broadcast, and contains a collection of BalooResource objects.
 - **BalooBroadcast:** This class shares the collection of resources (BalooResourceData objects) with BalooGUI, and is the class that is called by BalooBluetooth to handle incoming requests (after the request is fully transferred).
 - **BalooReply:** Helper class for storing information about the reply that BalooBluetooth should return.
 - **BalooLogging:** This class is used for logging events to the debug log.
 - **BalooStatistics:** This class is used for logging events to the statistics MySQL database.
 - **BalooHTTP:** This class connects to HTTP resources and fetches results from them.

Note that there is not a class for fetching file resources. This will probably be built into BalooBroadcast or BalooBluetooth, since there are functions for directly sending a file with the Bluetooth API.

18.3 Process view

The GUI in itself should run in a normal, non-threaded mode, where events (button presses and check box selections) activate the functions which are connected to those events, and run until they are finished. The only exception is the start/stop broadcast button(s), which should start the broadcast in another thread, so that the main GUI is still usable while the broadcast is running. This is necessary in order to allow the button for stopping the broadcast to work while the broadcast is running. Other buttons should be disabled, to avoid changing settings while the broadcast is running.

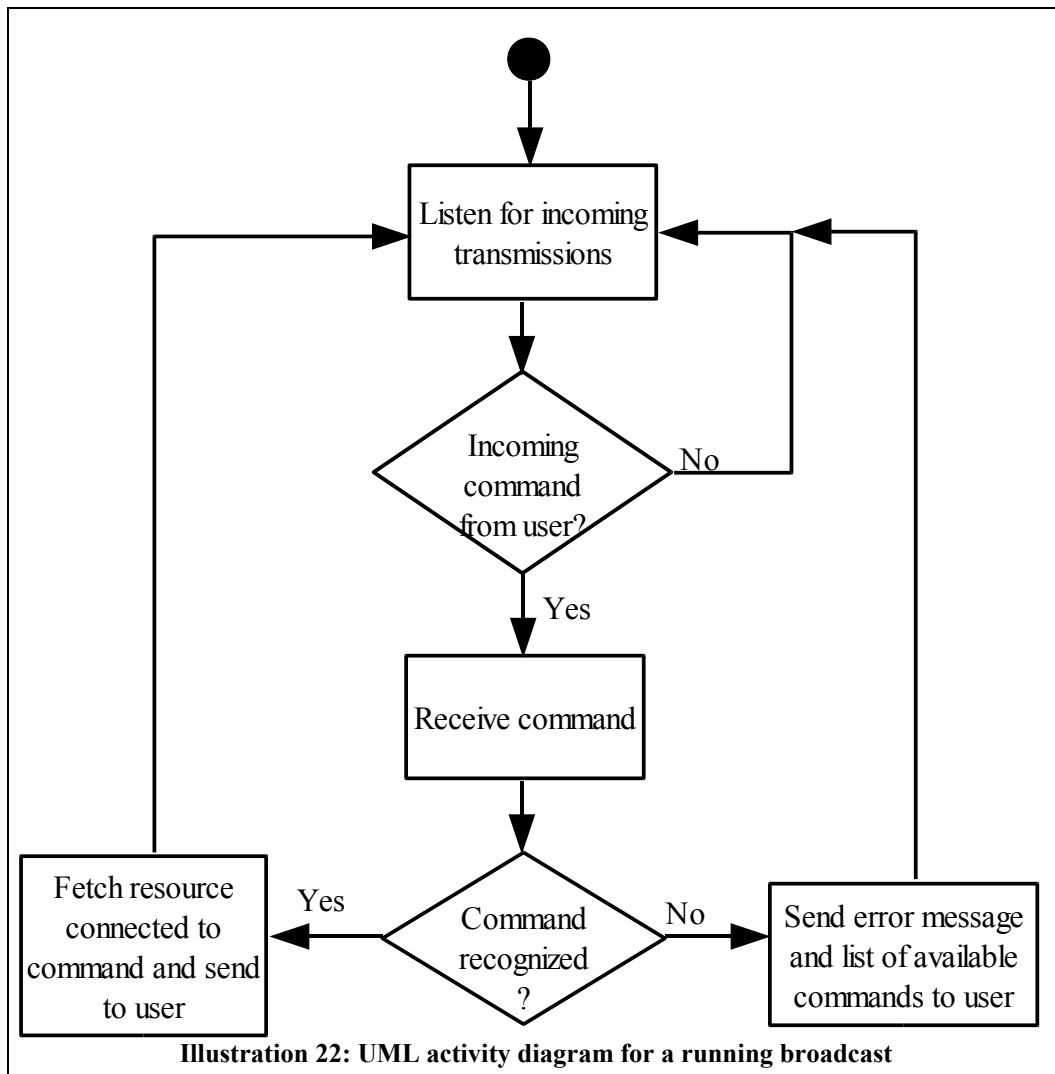


Illustration 22 shows the activity diagram for when the broadcast is running. The system is simple, so it should be quite stable.

19 Changes

This chapter describes any changes that were done to the architecture during development, and why they were found necessary.

19.1 Quality tactics

More elements were added to the design when implementing the GUI, and these elements were added to the quality tactics for completion. The added elements were:

- Text field for broadcast name.
- Text field for messages.
- What to do with incoming files: Added “Deny files” to the list.
- Options for selecting what to do with received files.
- Button for closing statistics window: Not needed because a tabbed interface was used instead of several windows.

This was removed:

- Button for saving a resource: Autosaving was implemented instead.

19.2 Architectural pattern

There was originally a “Resource fetching layer” below the “Resources managing layer” in “Table 2: Baloo's layered architecture (the bold layers are the layers that are within Baloo)“, however, this layer was removed.

The reason is that the Bluetooth API has functions for sending files directly, removing the file reading class that was in the layer. While the BalooHTTP class could still technically be left in the “Resource fetching layer”, it would be inaccurate to leave the layer there, when some of the resource fetching is done in the BalooBluetooth class too. It was therefore decided to move the BalooHTTP class to the “Resources managing layer”, and remove the “Resource fetching layer”.

19.3 Views: Scenarios

In “18.1.2 Adding a resource to a broadcast”:

- Changed the text inside the figure from “Opens configuration window” to “Shows configuration options”, since they were put in the same window in the implementation.

- Changed the name of the person who sets up the broadcast from “Broadcaster” to “Administrator”.

19.4 Logical view

- Added a class, BalooSettings. This made it easier to save the settings using XmlSerializer.
- Removed the functions in BalooBroadcast which were related to changing settings, since this is now done directly to the BalooSettings class instead.
- Changed broadcastname to name in BalooSettings, since it turns out the Bluetooth API is not able to change the broadcasted name.
- BalooReply was added to make it easier to communicate different replies from BalooBroadcast and BalooHTTP to BalooBluetooth.
- Reworked the whole diagram using Microsoft Visio (Instead of drawing directly in OpenOffice.)
- Added (almost) all public functions and properties for all classes.

Part V: Implementation

20 Requirements

Almost all functional and quality requirements were fulfilled. The ones that were not fulfilled are:

- **FR 4.1 Delete the hash at the end of the day**

This was not implemented because no tables were made for storing unique hits, so it has to be calculated each time unique hits is wanted. Also, the hash function was made to use a random new salt every time Baloo is started, so it is practically impossible to find out the Bluetooth ID from the hash from just the MySQL database. (When starting Baloo, two random characters are chosen, and these letters are added to the Bluetooth ID before it is encrypted, and the encrypted string is the string stored in the database.)

- **FR 5: Baloo should be able to use several Bluetooth radios**

This was not implemented because it turns out this is not supported in the current Bluetooth API. (More specifically, the 32feet.NET Bluetooth API uses Microsoft's Bluetooth stack, which does not support more than one Bluetooth radio at a time.)

- **FR 6.4 Limit specific commands to registered users**

This was not done because of time limitations, and because it is not critical for testing. A possible way of doing this would be to extend the URL resource type to also send the Bluetooth ID to the URL, and let a dynamic website handle limiting who can get information. However, this could also affect user privacy. Either way, since it was not critical for testing, it was left out.

- **QR 1.1 Baloo should always reply to a information request**

It seems there is a weakness in the Bluetooth sending functionality which sometimes returns a BadRequest for no apparent reason when trying to send a Bluetooth file to someone. The recipient will not even see that a file was attempted to be sent to him. It is possible to make Baloo retry sending when this happens, however, when a user cancels an incoming request, it also returns a BadRequest, and no way was found to distinguish between the two. Since the sending problem does not happen too often, it was therefore chosen to leave out retries in the test system for now.

21 Baloo's GUI

This part of the document will describe the functionality of Baloo, using screens from the application together with descriptions of common tasks. Note that we will use two terms for describing the people who use the broadcasts: The *administrator* is the one who sets up the broadcast using Baloo's GUI, while the *user* is those who access the broadcast for its services. The users should not have access to the GUI, they are only accessing Baloo via the Bluetooth connections.

21.1 Main configuration screen

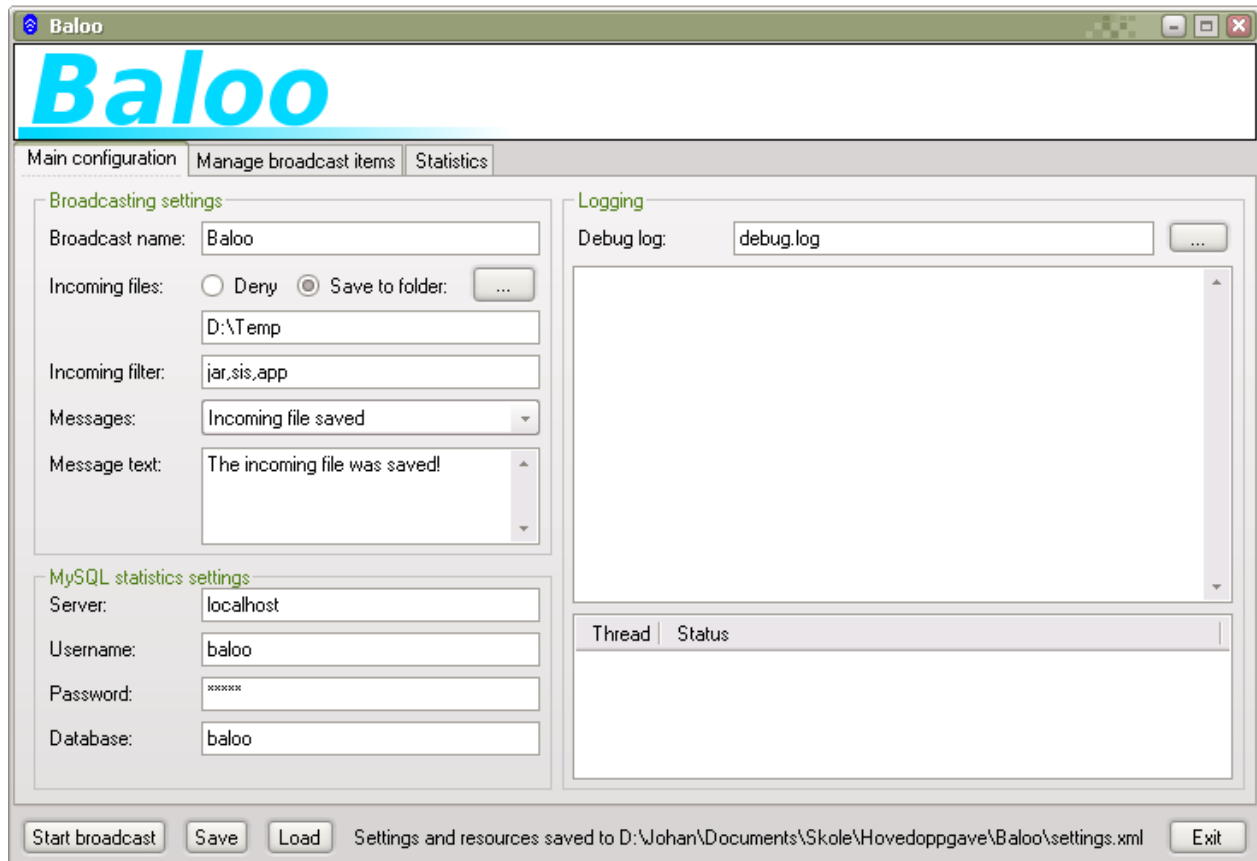


Illustration 23: Main screen

Illustration 23 shows the main configuration screen of Baloo. Here the administrator can change the settings for the broadcast. Most of the items shows a tooltip if the mouse pointer is held over the label, so it should be quite selfexplanatory.

Part V: Implementation

The buttons and status bar that are located at the bottom of Baloo are always shown. (The title image at the top is also always shown, though it does not actually do anything.)

- The “Save” and “Load” buttons saves and loads all settings, including all resources that are set up.

Note: If the file “settings.xml” exists, Baloo will automatically load this on startup. This was added to make testing easier, and has been left there. Saving has to be done manually though.

- The “Start broadcast” button starts the broadcasts if a Bluetooth adapters is found, and the buttons then change to “Stop broadcast”. No settings can be changed when a broadcast is running.

21.1.1 Broadcasting name

The broadcast name does not really do anything. It was intended to be shown as the Bluetooth name when broadcasting, but it turned out that it was not possible to change this in the code. This text field is still left there, in case newer versions of the 32feet Bluetooth API is able to change the broadcasted name.

21.1.2 Incoming files

This option determines whether incoming files that are not text files are accepted, and where to store them.

OBS: There is a bug in the 32feet ObexListener which makes it impossible to receive files larger than a certain size. Users trying to send too large files will find that the transfer halts midway without doing anything. Because of this issue, this option is a lot less useful than it could have been. Even if Deny incoming files is set, the same thing will happen when users try to send too large files! Hopefully this will be fixed in later versions of the 32feet API.

The custom listener that was made to work with the iPAQ device used for testing, supports receiving larger files, so sending larger files with iPAQ (and possibly other Windows Mobile devices) should work better. However, this listener does not seem to work as well with the other devices, so it will most likely only work with Windows Mobile devices.

21.1.3 Messages

The “Messages” part is used to set some feedback messages that are sent to the user, there are currently four messages that can be sent:

- Incoming file saved:
Used when the user sends a non-text file to the Bluetooth station, and it is saved.
- Incoming file not accepted:
Used when the incoming non-text file is not accepted, either because the broadcast is set to deny incoming files, or when the folder that is used for storing incoming files is inaccessible.
- Resource unavailable:
Used when a command that is received from a user is not recognized, or when a command is recognized, but the resource can not be fetched for some reason. (For example if a web page can not be accessed.)
- File not found:
Used when a file requested was not found.

21.1.4 MySQL statistics settings

The fields in this box sets the MySQL server and database which should be used for storing statistics: Each user request and the reply that was sent back will be saved, in addition to a hash of the user's Bluetooth ID, which is used to calculate unique hits. The MySQL table used for storing the statistics is automatically created if it does not already exist. If these fields are left blank (or contains invalid information), no statistics will be saved. An error message will be shown in the log when starting the broadcast.

21.1.5 Logging

The file selected here will be used for storing debug messages that are shown when the broadcast is running. The box under the debug log the same messages, but this box is emptied when Baloo is exited. If no file is given, only the text box will show the messages, and they will not be saved.

The list box under that shows the currently running reply threads and their statuses. When an incoming request is received, a reply thread is created to handle that request, and when replying is done, the thread exits and disappears from the list box.

21.2 Managing broadcasted resources

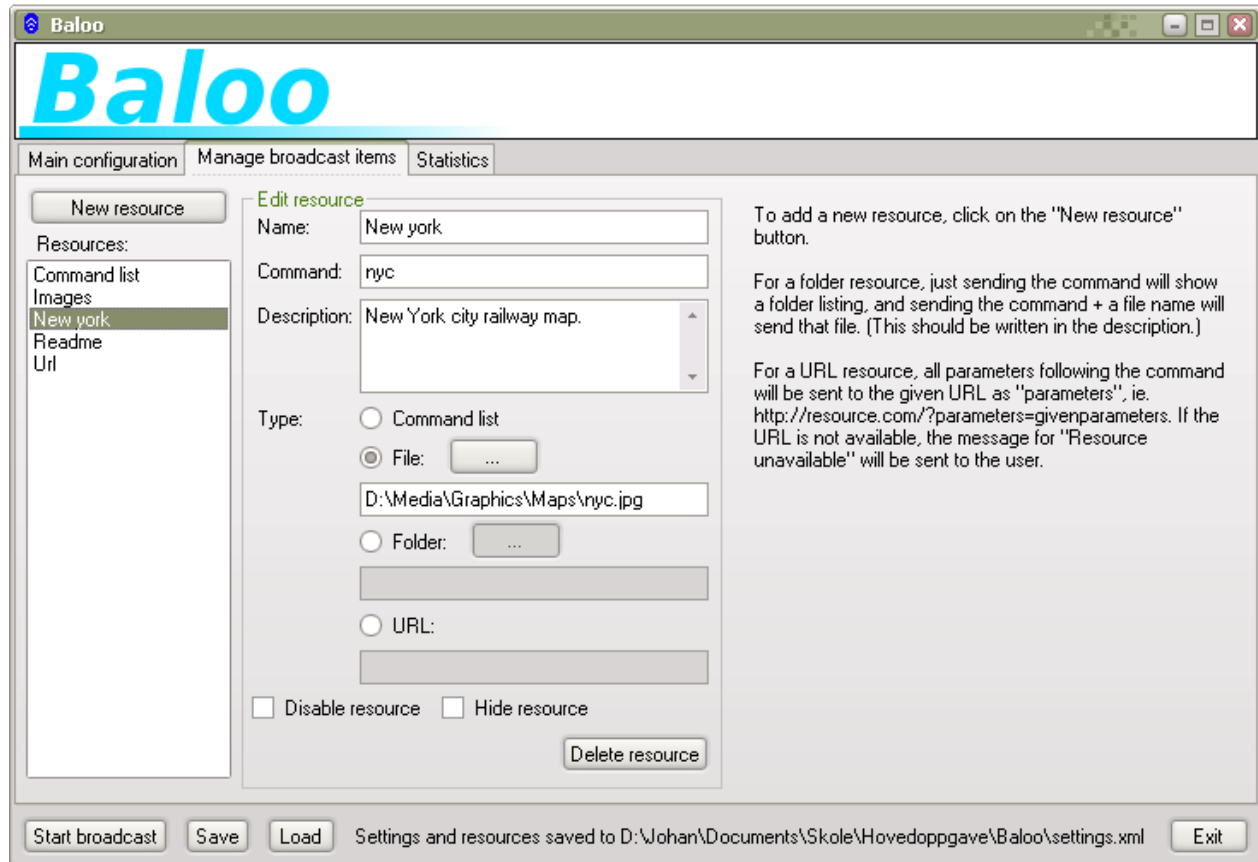


Illustration 24: Manage broadcasted resources

Illustration 24 shows the resource management screen. Here administrators can add, edit and remove the resources for the broadcast. New resources are added by clicking the “**New resource**” button. When a resource is selected, its settings are shown and can be edited. The resource **name** is only shown in the GUI, while the **description** is usually used in the command list, for describing what the **command** returns. There are four different types of resources:

- **Command list:** This returns the text in the “Description” text field, and then a list of all the resources that are not hidden and not disabled. The list shows each resource's command first, and then the description for that resource.
- **File:** This returns the file that is selected in the text field under the radio button.
- **Folder:** This returns a file listing of the files in the selected folder. If a filename is passed along with the command, the file is returned instead, if it exists.
- **URL:** This fetches a page from the given URL. If there are any parameters, these are sent as GET values to the page, and can be used for generating the reply.

Note that the **command** is case insensitive! Also, if you have several resources with the same command, only the first resource will be accessible to users, even though both will be listed in the command list.

“**Delete resource**” deletes the resource completely, and this can not be undone (though if you have saved the settings before deleting, you can still load the old settings). For making a resource temporarily inaccessible, it is better to use the “**Disable resource**” check box, which will make the broadcast treat that resource as if it did not exist. “**Hide resource**” will hide the resource from the command list sent to the users, but the resource will still be accessible by anyone.

21.3 Statistics

The statistics screen shows a variety of statistics using the current settings for the MySQL database. The types of statistics that can be shown are:

- “**Show all events**”: This just shows all the events in the database, with the events' id, date and time of the request, the command sent, any parameters, the reply sent, and the hash of the user's Bluetooth ID.
- “**Hourly statistics**”: This shows the number of hits and unique hits that the server has received per hour.
- “**Daily statistics**”: This shows the number of hits and unique hits per day.
- “**Commands popularity**”: This shows all the commands requested (including commands that do not exist as a resource), and the number of hits and unique hits for each of them.

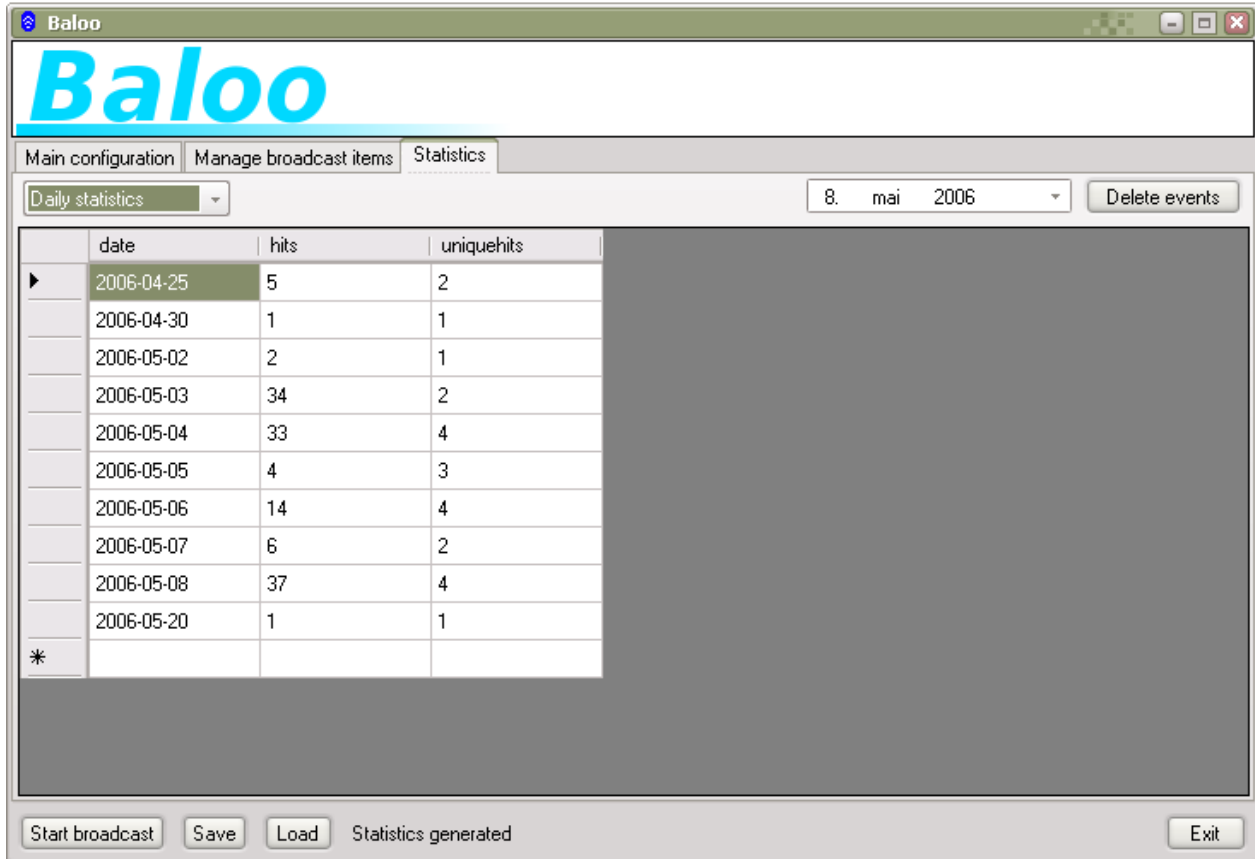


Illustration 25: Statistics example: Hourly statistics

The “**Delete events**” button deletes all events that are older than the date selected in the date selection box next to the button.

Note about the Bluetooth ID hash and the unique hits calculation: The Bluetooth ID hash is generated using a MD5 algorithm, with four random bytes added as a salt. The salt is randomly selected at the startup of Baloo. This means that if you start Baloo twice (or more), the unique hits for that day and hour will not be correct, since the hash will be different.

22 Accessing the broadcast

This chapter describes how to access the broadcast using a few common handheld device types.

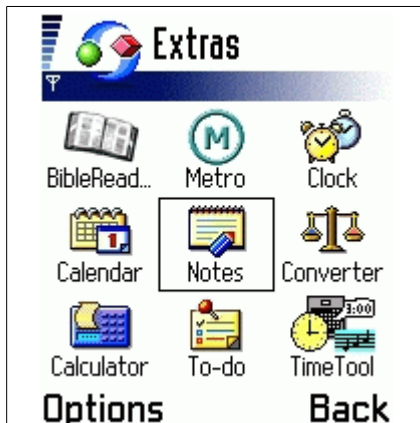
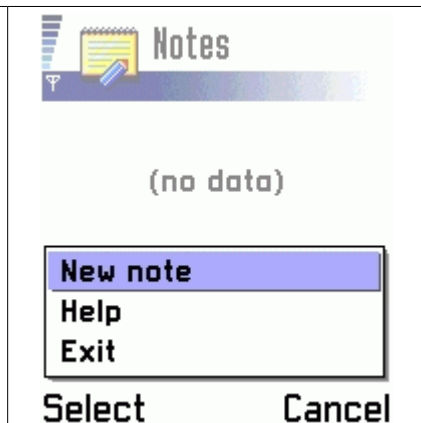
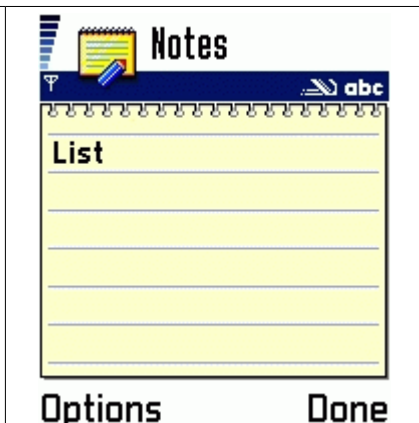
First some info about the broadcast that was used for testing:





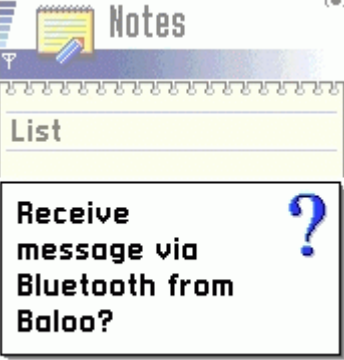

- The broadcasting station has been set up to use the Bluetooth name “Baloo”. This can however be changed in the Bluetooth settings on your computer. If you have set up a broadcast on your own computer without changing the Bluetooth name on it, you will most likely see your computer name when searching for Bluetooth devices, instead of Baloo.
- The “list” command is an example, and should return a list of other commands that can be used for the test broadcast that has been set up in this project. This command can also be different on different broadcast, depending on how Baloo has been configured.
- Not all commands return text, some can return files like images and sounds, depending on what resources you have set up on the broadcast.

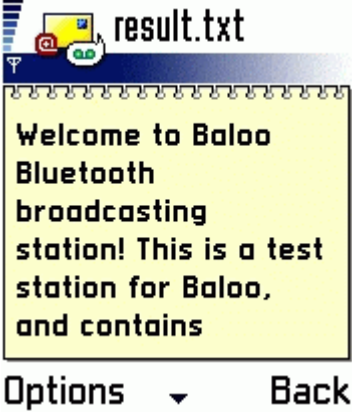
Also, note that Baloo is case insensitive, so the commands sent in can be capitalized in any way, and they will still be recognized.

22.1 Using a Symbian Series 60 phone

This chapter describes how to access a Baloo broadcast using a Symbian Series 60 phone. There might be differences in different phone models. The screenshots were taken on a Nokia 3650.

 <p>1. Open the menu, and select the “Notes” application.</p>	 <p>2. Create a new note (or open an old note).</p>	 <p>3. Write the word “list” in the note.</p>
--------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------

 <p>Select Cancel</p> <p>4. Select “Options”, then “Send”, and select the “Via Bluetooth” option.</p>	 <p>Yes No</p> <p>5. If you have not turned on Bluetooth, this message will show up. Select “Yes”.</p>	 <p>Select Cancel</p> <p>6. The phone should show a “Searching” message, and then list up the Bluetooth devices it found. Select “Baloo”.</p>
 <p>Cancel</p> <p>7. It should then connect to Baloo and send the message “list” to it.</p>	 <p>Yes No</p> <p>8. About 5 seconds after sending the message, you should be receiving a message from Baloo. Select “Yes”.</p>	 <p>Show Exit</p> <p>9. After the message is received, you will get the “Message received” sound from your cell phone. If you go to the main screen of the cell phone, you should get a “New message” box. Select “Show”.</p>



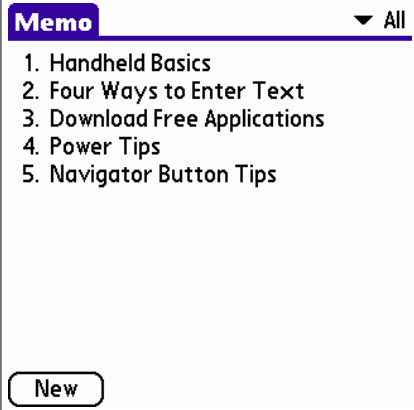

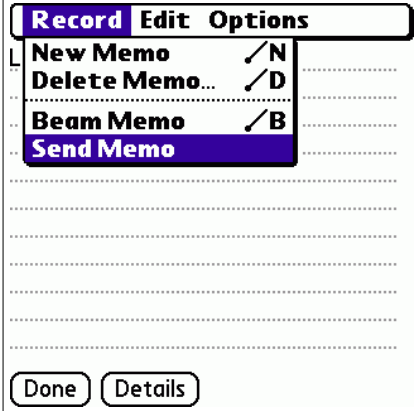
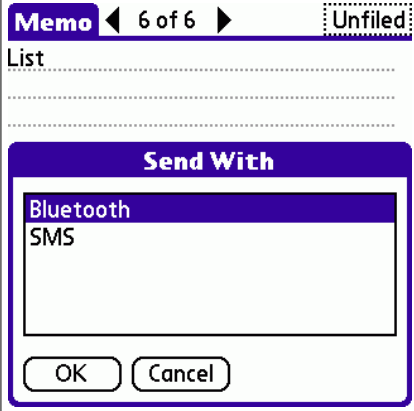



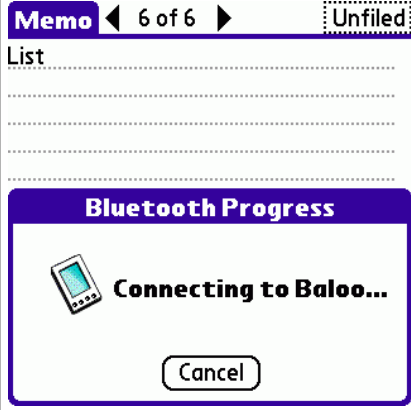
 <p>10. The message you received should then be showed using the Notes application.</p>		
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

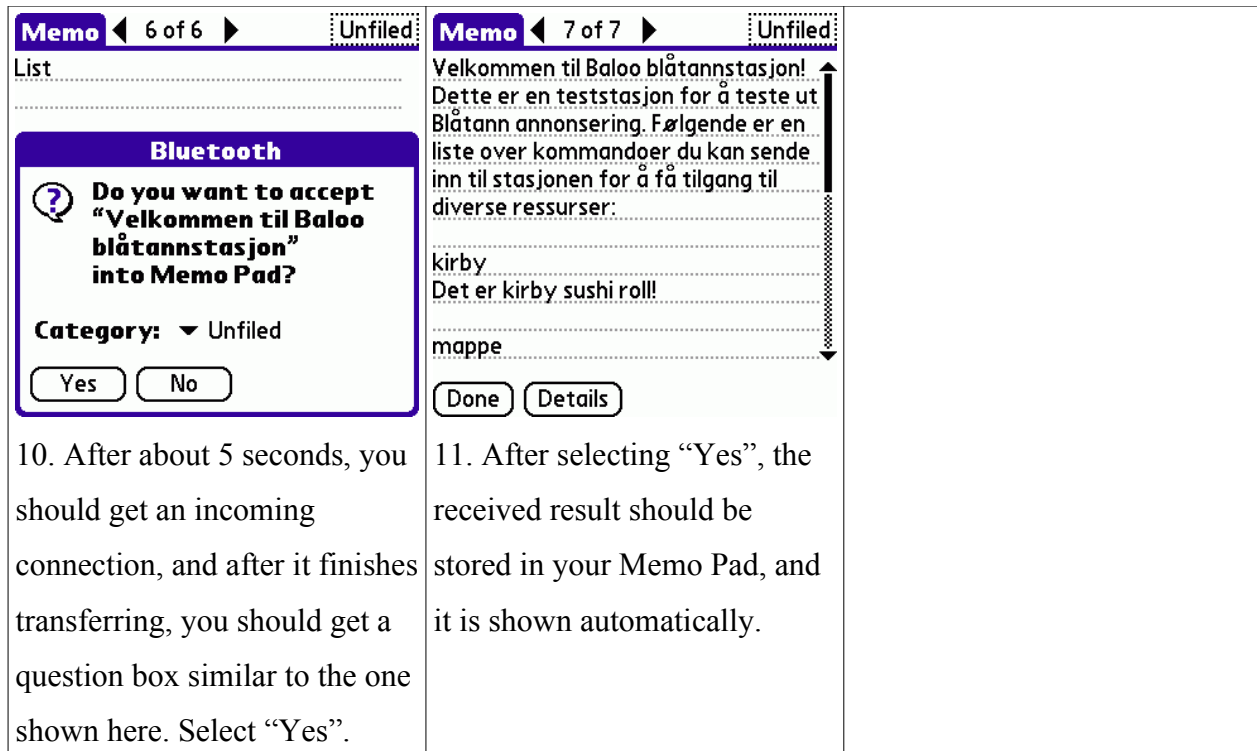
When using Symbian 60, all received files are stored in the inbox, whether they are text messages, pictures or sound files, and they will stay there until they are deleted by the user.

22.2 Using a Palm device

These are the instructions for using Baloo with a Palm device. The following screenshots were taken on a Palm Tungsten T (m550). There may be differences on different Palm devices.

Important note about Palm devices: If there is no default application set up for receiving specific file types, these file types will give the error “Data received in unknown format” and will be discarded. If you get this error, you may need to install additional applications on your Palm device to receive these types of files. Also note that not all applications are able to set themselves as the default application for a file type, so even if you install a image viewer, it is possible that you still will not be able to receive image files. TXT files should work with the Memo Pad by default, however.

 <p>12:51  All</p> <p>Address Calc Card Info Date Book Expense Graffiti HotSync Memo Pad Note Pad Prefs Snap To Do List</p> <p>1. Start the Memo Pad application on your Palm.</p>	 <p>Memo All</p> <p>1. Handheld Basics 2. Four Ways to Enter Text 3. Download Free Applications 4. Power Tips 5. Navigator Button Tips</p> <p>New</p> <p>2. Tap the “New” button at the bottom of the screen.</p>	 <p>Memo 6 of 6 Unfiled</p> <p>List</p> <p>Done Details</p> <p>3. In the new note, write the word “list”.</p>
 <p>Record Edit Options</p> <p>New Memo /N Delete Memo... /D Beam Memo /B Send Memo</p> <p>Done Details</p> <p>4. Select “Send Memo” from the “Record” menu.</p>	 <p>Memo 6 of 6 Unfiled</p> <p>List</p> <p>Send With</p> <p>Bluetooth SMS</p> <p>OK Cancel</p> <p>5. Select Bluetooth in the “Send With” dialog box.</p>	 <p>Memo 6 of 7 Unfiled</p> <p>List</p> <p>Bluetooth Disabled</p> <p>Bluetooth is currently disabled. Do you want to enable it?</p> <p>OK Cancel</p> <p>6. If you get a question dialog like this, just select “OK”.</p>
 <p>Memo 6 of 6 Unfiled</p> <p>List</p> <p>Bluetooth Discovery</p> <p>Searching for Bluetooth devices...</p> <p>Cancel</p> <p>7. Your Palm device should then start searching for Bluetooth devices.</p>	 <p>Memo 6 of 6 Unfiled</p> <p>Discovery Results</p> <p>Select one or more devices:</p> <p><input checked="" type="checkbox"/> Baloo</p> <p>OK Cancel Find More</p> <p>8. After the search is finished, it should show a list of devices. Select “Baloo” and press “OK”.</p>	 <p>Memo 6 of 6 Unfiled</p> <p>List</p> <p>Bluetooth Progress</p> <p>Connecting to Baloo...</p> <p>Cancel</p> <p>9. Your Palm devices should then connect to Baloo and send your memo to it.</p>



10. After about 5 seconds, you should get an incoming connection, and after it finishes transferring, you should get a question box similar to the one shown here. Select “Yes”.

11. After selecting “Yes”, the received result should be stored in your Memo Pad, and it is shown automatically.

Notes about using Palm devices with Baloo:

- If the user does not have software for handling a specific file type, the Palm device will reject the file after receiving it, even if it is a common file type like images.
- HTML files can be received and shown if the Palm device has WebPro (or other browsers) installed, however it seems like WebPro may not always refresh the page shown when it has the same file name. In this case, Balooreply.html is used for HTML replies generated from PHP urls. If WebPro opens and shows the old result, just click the Refresh button.





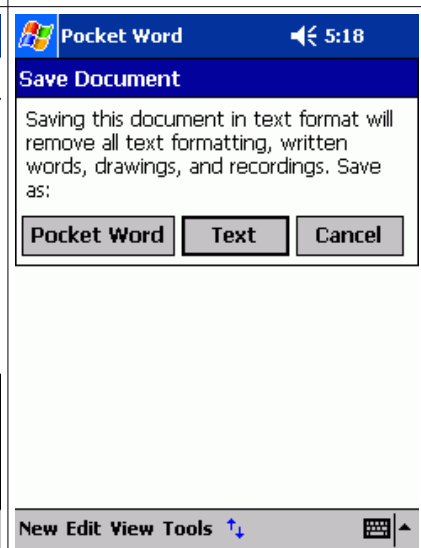
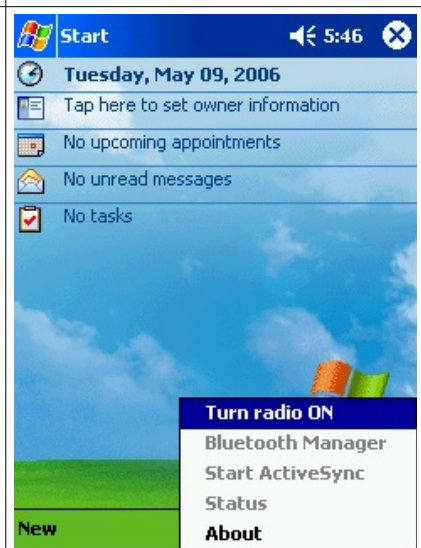
22.3 Using a Pocket PC device

The Pocket PC device used for testing was an iPAQ H3970. Unfortunately, it seems like Pocket PC version 3, which was the OS on that device, did not work well with object push transfers of other file types than calendar and contact files. Even though it works, it is not quite intuitive.

Even sending the request command is not quite that simple, because the Notes application that comes with Pocket PC does not store notes as plain text files (TXT). Since the Notes application supports adding drawings and sound recordings to the notes, it stores the notes using a custom

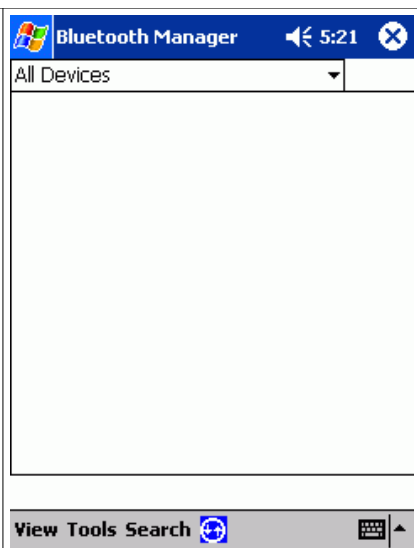
Part V: Implementation

file type (PWI). When testing with the iPAQ device, PocketWord was used for creating the notes instead, since it at least had the option of saving a plain text file.

 <p>1. First, use Pocket Word (or another text editor, preferably one which saves as TXT by default)</p>	 <p>2. Once started, create a new file by tapping “New”.</p>	 <p>3. Write the word “list” in the new file.</p>
 <p>4. Save the document, make sure it is saved as plain text file.</p>	 <p>5. You will get a confirmation box, select “Text”.</p>	 <p>6. On the Today screen, click on the icon in the lower right, and turn on Bluetooth.</p>



7. Start Bluetooth Manager.



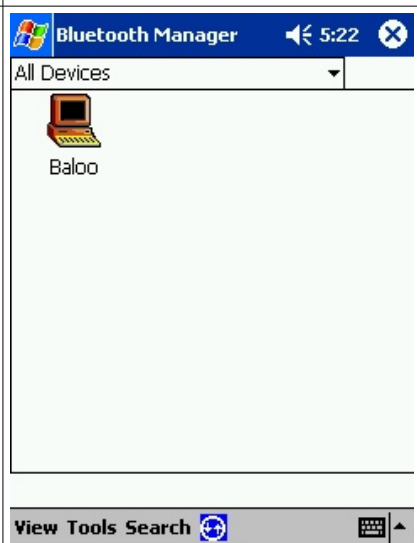
8. If you have not done this already, tap “Search”.



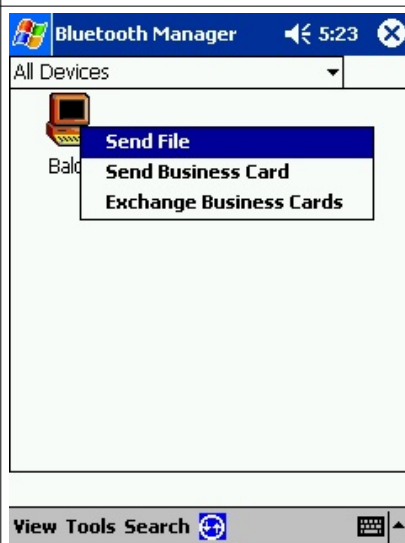
9. Baloo should show up, select it and tap “Save”.



10. Choose a group if you want, then tap “OK”, which will return you to screen 9. Tap “OK” in the upper right corner.


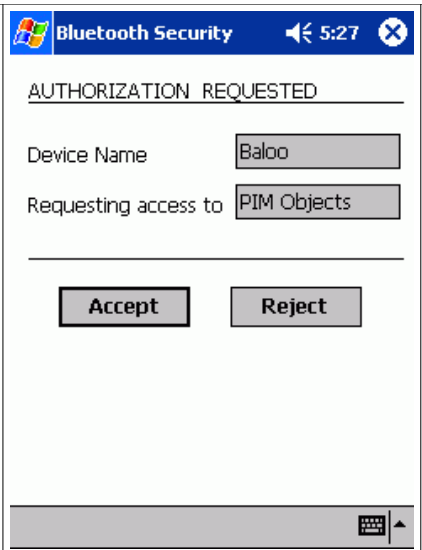
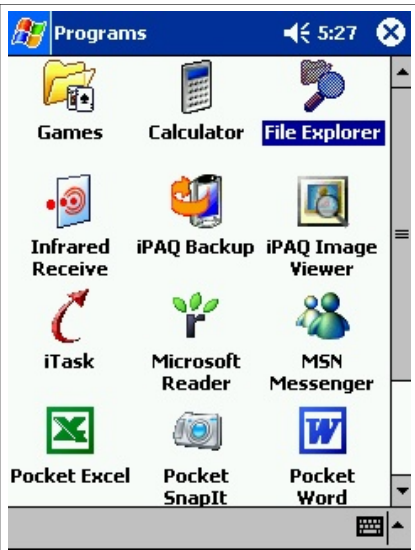


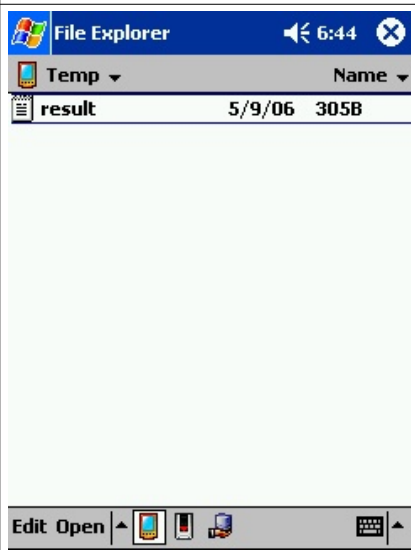


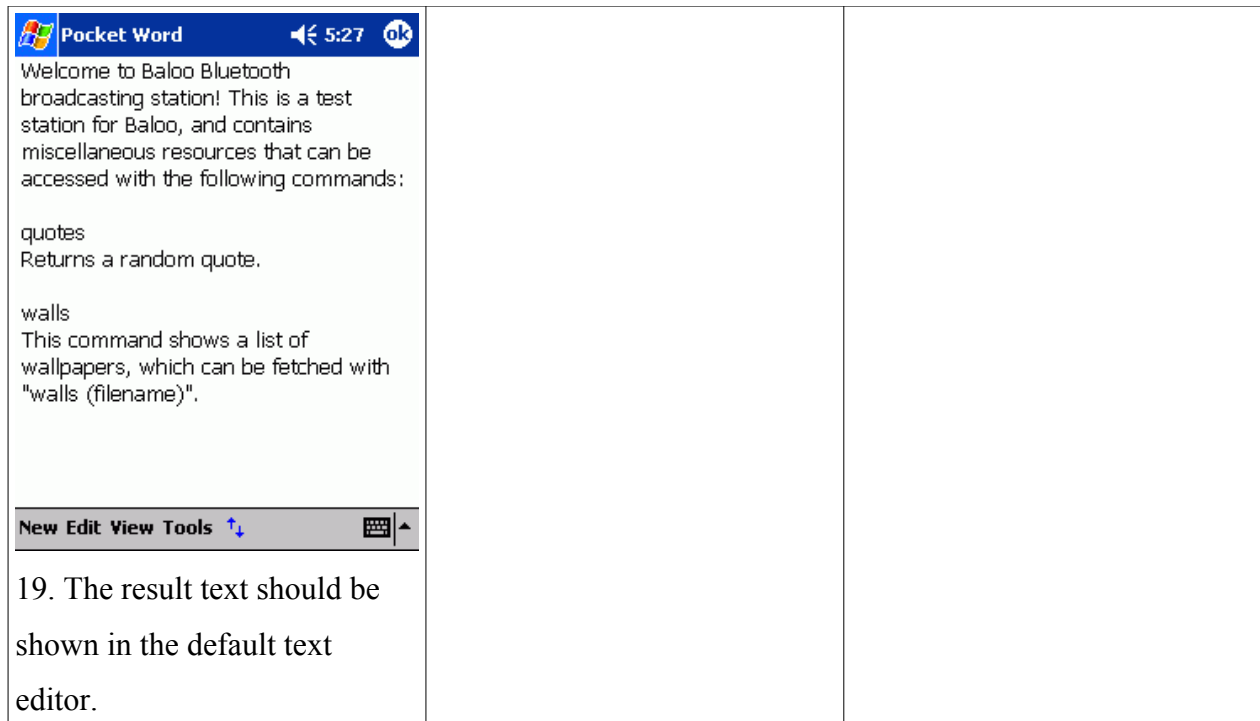
11. Baloo should now be on the list of devices. Tap and hold the pen over Baloo.



12. A menu should be shown, select “Send File”.

Part V: Implementation

 <p>13. Select “Browse...” and choose the file that was saved in point 3-5.</p>	 <p>14. About 5 seconds after sending, you should get an incoming request. Choose “Accept”.</p>	 <p>15. Start File Explorer.</p>
 <p>16. Select “My Device” from the “Show” menu.</p>	 <p>17. Go to “Temp”.</p>	 <p>18. Open the result file.</p>



Notes:

If the file received from Baloo is a contact or calendar event, it will not be stored in “Temp”, but it will be automatically inserted into the contact list or the calendar.

The most likely reason for why other files are stored in “Temp” is that Pocket PC stores files there for processing, but does not have any “default actions” for these file types, and therefore ends up leaving them there.

It is possible that Windows Mobile and other newer versions of Pocket PC will handle other file types more intuitively.

Part VI: Results and conclusion

23 Answers on the research questions

The following subchapters describe the answers for the research questions asked in chapter 2.1 “Research questions“. While it was originally planned to do a test of the broadcasting system with users, it has been necessary to leave this out, due to lack of time and resources for testing. For this reason, the answers will be based on the prestudy, the experiences gathered during development and the testing done during implementation.

23.1 Research Answer 1

RQ1: Is Bluetooth suited for broadcasting?

From the issues mentioned in the prestudy, it would seem that it might be problematic to use Bluetooth for broadcasting information to a large audience.

Chapter 7.1 “Bluetooth” stated that a Bluetooth device can send to seven other devices at the same time, and that the maximum speed is 723.1 kilobit per second, or roughly 90 kilobyte per second, about the same speed as the slower broadband Internet connections today. The speed would go down when more users are downloading at the same time, obviously. With seven users at the same time, the speed would go down to roughly 12-13 kilobyte per second, but that is the maximum speed. Interference and other influence might lower the speed even further, leaving roughly the speed of dial up Internet connections. For practical reasons, sending large files over Bluetooth should therefore probably be limited to situations where it is not assumed to be many users at the same time.

While multiple Bluetooth radios on a Bluetooth station could allow for more simultaneous users and also prevents the speed from going down as quickly with more users, there are some limitations to this. Since the radio frequencies used are limited, having many Bluetooth radios in near proximity would give an higher risk of interference, reducing performance. It also requires a more advanced hardware/software solution, since Windows XP by default does not support more than one Bluetooth radio at a time.

Another problem is the other issue mentioned, that a Bluetooth device might not be found by all the users if a lot of users are doing device discovery at the same time. This issue could be quite a problem for broadcasting to a large audience, depending on how serious it is in practice.

Still, it is uncertain how serious these issues will affect an actual broadcast system. Some of the issues might not be as big a problem as feared. For example, even with limited transfer speeds and connections, as long as the files sent are of reasonable size, connections should last briefly enough for it to be possible to transfer files to a large number of users per time unit.

Also, Bluetooth as a broadcasting technology is quite unique, and few other technologies can be used to broadcast files in the same way. Wi-Fi is the technology which comes closest, but it is not as commonly implemented in devices like mobile phones, and also requires specific software to be running in order to transfer files.

All in all, Bluetooth might not be optimal for broadcasting in all kinds of situations, but it would work for certain usage areas, and its uniqueness should bring new communication opportunities where other technologies might not work as well.

23.2 Research Answer 2

RQ2: What kind of information can be sent with a Bluetooth broadcast?

The Bluetooth technology by itself does not really have any limitations on the data that can be sent, except for the fact that Bluetooth transfers are somewhat slow, making it unpractical to transfer large files over Bluetooth.

The biggest limitation, however, lies in the receiving devices. Most mobile phones can only receive a small set of file types, and Palm OS devices will discard unknown file types. It therefore seems that when targeting a large audience, the best would be to only broadcast the file types which are most commonly supported, thus seriously limiting what kind of information (or rather, file types) can be sent.

Part VI: Results and conclusion

In fact, it turns out that even the task of sending a simple text message via Bluetooth might not be as easy as expected. In “22.3 Using a Pocket PC device“, it is revealed that to send a simple text message, one would first need to create the file, and then one would have to open the Bluetooth manager to send the file from there, making it less intuitive compared to both Symbian OS and Palm OS, where the user can just select “Send to Bluetooth” in the menu from their respective notes application. The fact that the Pocket PC device did not show any messages at all after receiving a file which it did not recognize also lowers the usability.

The fact that Palm OS devices would not store unknown file types was slightly surprising, though it should probably have been expected. Since this OS attempts to “cloak” the fact that there is a file system underneath the OS, it does not even have a file explorer application. Even if it accepted and stored unknown file types, users might not be able to access the file afterwards either way. For devices with memory cards, it would of course have been preferable if Palm OS would just store received files on the inserted memory card for later use, but apparently it is not possible without installing additional software for doing this.

Surprisingly, the Symbian device that was tested seemed to be the most usable: It was reasonably easy to write and send the text command, and it would receive any file type into the inbox. (It used the same inbox as the one used for receiving SMS and MMS messages, when receiving files over Bluetooth.) It obviously was not able to show all files, but it would be quite easy to transfer the received files to a computer for viewing.

In the end, it is somewhat disappointing that the two most common operating systems for PDAs did not work as well as expected, though Palm did work reasonably well as long as one made sure to only fetch files that it supported, and the tested did support the most common files. Regrettably, since user testing had to be canceled in this project, only a the few devices mentioned so far has been tested with the broadcast. Testing with a wider range of different mobile phones would have been most interesting, since mobile phones are the most common handheld device.

In any case, the file types that were found to be most certain to work properly seemed to be only calendar events, address book contacts and todo events. However, plain text files and images are well supported enough to also be considered for use in a broadcast, even with the minor issues that were described above for a few of the devices.

For a broadcasting system where the users can select what to download, it might not be necessary to have a limitation on the file types that can be sent. However, it is still a fact that many Bluetooth devices will be unable to receive a wide range of information and file types from a broadcast. Still, it is possible that this might change sooner than one can predict!

23.3 Research Answer 3

RQ3: Can Bluetooth broadcasting be done in a way which does not compromise users' privacy?

In Baloo, the broadcasting system developed in this project, the system was chosen to be completely passive, letting users be the ones who take initiative for the connections. This has three advantages related to user privacy:

1. It will not send requests to users without them asking for it first.
2. It does not need to log user details to avoid sending the same info again, making it possible to set up a broadcast which does not log any user details at all.
3. The users do not even need to set their devices to “Discoverable”. When the users send the command for requesting information, their Bluetooth ID will also be sent, and this Bluetooth ID is sufficient for the broadcasting station to send the reply.

Even if using a broadcasting system where the files are sent automatically, it is still possible to maintain users' privacy. One could make a system which requires a user to register his Bluetooth ID on a website or through the broadcast, before he will receive anything from it automatically. Also, even if user details are logged to avoid resending the same information, this information does not need to be stored anywhere except for in memory, making it disappear when the broadcasting system is shut down, or when the details is purged, for example one or more hours after last time the user received anything from the broadcast.

All in all, it is quite possible to make a Bluetooth broadcasting system which does not affect users' privacy.

23.4 Research Answer 4

RQ4: What usage area(s) would Bluetooth broadcasting be best for?

In “4 Why Bluetooth broadcasting?“, the following features for Bluetooth were described:

- Area broadcasting instead of group broadcasting
- Sending information where people are
- Portable information
- Cost-free distribution

The first three points in the list of features actually overlap each other. With a Bluetooth broadcast, users that are in the nearby area can access and “take” the information with them, allowing them to view the information at any time. The difference from using Wi-Fi is that only computers or advanced PDAs or smart phones can use Wi-Fi, but a lot of newer mobile phones and PDAs can use Bluetooth. And the difference from normal phone services like SMS, MMS or GPRS, is that Bluetooth will be limited to the proximity, and is also cost free both to send and to receive.

When starting this project, there were mainly three usage areas that were considered for Bluetooth broadcasting:

1. File sharing, like sharing documents in a workspace like a university or company, or sharing multimedia files like video clips or photo albums with friends in a club or school society.
2. Sending compact information to (often anonymous) visitors of a place like a hospital or a company.
3. Send information to large groups of people in special happenings like fairs, festivals and similar events.

As mentioned in “23.1 Research Answer 1“, Bluetooth might not be optimal for broadcasting to a large audience, thus limiting the effectiveness of using Bluetooth in situations where it can not be

assumed that only a few people will try to access the broadcast at the same time. Using Bluetooth broadcasts in large events like described in the third usage area might therefore be difficult. However, Bluetooth would work when the information is assumed to be accessed by only a few people at a time, making the two other usage areas possible.

The other issue mentioned in that chapter is the issue of transfer speeds. The speed might not seem all that slow, but when transferring larger multimedia files and documents, the transfer time will be quite noticeable. Also, many Bluetooth devices does not show a working progress bar for incoming transfers, meaning that users will not know when the transfer is finished. These facts make it rather impractical to use Bluetooth broadcasts for larger files than text, calendar events and smaller images, making it less suitable for the first suggested usage area.

The usage area that remains is therefore the second one, broadcasting compact information. This usage area can be compared to information delivered as brochures. Instead of, or in addition to using the traditional paper based brochures, one could broadcast the same information over a Bluetooth broadcast. The advantages Bluetooth would have over paper based brochures are:

- Brochures can run out, but a running broadcast will always have information available.
- Information can easily be changed in the broadcast, while brochures needs to be reprinted.
- The users might misplace the brochures, but information stored on their handheld devices will stay there until they delete it.
- Content can be context-sensitive, for example the content could change based on date and time (like bus schedules).
- Allows for multimedia content like audio and even video, though some phones might not support this. Also, the size of the files needs to be considered. If the broadcast is assumed to be used by only a few people at the same time, larger files can be allowed, otherwise, files should preferably be smaller.

There are several reasons “information brochures” were chosen as the equivalent to information broadcasted in a Bluetooth broadcast. While advanced as handheld devices has gotten, it is still often not very comfortable to read large amounts of text on them. Information for mobile devices should therefore generally be short and concise. Another thing is that to keep file sizes low, the

Part VI: Results and conclusion

information sent has to be limited. Though this is not a problem for plain text files, once images and other media files is used, this becomes an issue. Brochures, being generally small with compact information, are therefore the most suitable comparison for information sent via a Bluetooth broadcast.

Moreover, in “4.5 More specific examples of usage”, the following usage areas were suggested for a Bluetooth broadcast:

- Bus stations and other stations could broadcast route schedules.
- Concert halls, theatres, museums and similar places could broadcast calendar events and information like opening hours.
- Trains and buses could broadcast information inside the vehicles, for example the list of bus/train stops, travel time, modified vacation schedules, and other information.
- University/school classes could broadcast calendar events for lectures and important deadlines, and for example information about the syllabus and the books to be bought for a class.
- Stores could broadcast advertisements and special offers.

The common point for all these suggestions is indeed that the information sent is short and concise, and that the information in most of these examples would traditionally be delivered as information brochures.

After considering the details gotten from the research done in this project, this usage area does seem to be the type of usage which fits Bluetooth broadcasting the most. Still, usage in other areas should not be completely ruled out, though testing out how well Bluetooth would actually perform in different situations will have to be left to future projects.

24 Conclusion

From the answers that were given in “23 Answers on the research questions“, it seems like Bluetooth broadcasting might not be quite as flexible as it was hoped to be in the beginning of this project. It is not possible to simply set up a broadcast to send a wide variety of file types and hope that “everyone” will be able to access them. Without the user test, the conclusion will have to be that Bluetooth broadcasting is not quite as flexible as expected, due to the variations in how Bluetooth devices handle incoming files. This inflexibility makes the usage areas for Bluetooth broadcasting somewhat limited.

Still, even using just the most supported files described in “23.2 Research Answer 2”, it should be quite possible to set up a quite functional Bluetooth broadcast. And the suggestion given in “23.4 Research Answer 4” about using it as a supplement or replacement for information brochures might be the place to start. With calendar events that will be added to a device's calendar, images that the users can view on their mobile device at any time, and contacts that are added to the phone book automatically, a Bluetooth broadcast can be quite a useful addition to the traditional information brochures!

And with devices continuously evolving and the Bluetooth technology being improved, it is possible that Bluetooth broadcasting can be used for delivering a wider range of information in the near future at higher speeds than now. Therefore, Bluetooth broadcasting as a channel for delivering information definitely has potential as a technological solution.

Part VI: Results and conclusion

Part VII: Further work

25 Further work

The first thing that any projects building on this one should focus on, is how much technology has progressed after the completion of this project. Has a new version of Bluetooth become common, and what improvements and possibilities does it bring? Are Bluetooth devices more common and more flexible? While the importance of this might depend on how much time has passed, technology tends to progress rapidly enough for it to have improved even if a new project is done on this subject within a year after this one.

A user test for finding out the public opinion on such a system should also be a focus, especially since it had to be skipped in this project. Some things that might be interesting to find out with a test group, would be:

- Would they use a broadcasting system like this if it was available?
- Is it convenient enough?
- Would they prefer to have a system where information is sent automatically?

One thing that has not been discussed much at all in this project, is the security issues with a broadcast. Would it be possible for attackers to send harmful files to people who are expecting a file from a Bluetooth broadcasting station? Is there any ways to prevent attacks like this, or other types of attacks, while still having a working broadcast? The reason this has not been discussed in this project, is that it is quite a large topic, and would probably have made this project too large for the time span given. Besides, security was hardly an issue before it was found whether Bluetooth as a broadcasting channel had potential or not. Now that it is concluded that Bluetooth broadcasting does have potential, however, security issues might need to be researched.

The test broadcasting system that was developed in this project could also be improved upon.

One thing that was requested by supervisor Alf Inge Wang was the option to set the broadcast to automatically send some information to users nearby having Bluetooth devices set to discoverable. This was not included due to time limitations, and is also generally frowned upon by owners of Bluetooth devices. However, it would probably still have some interest for testing purposes. One other thing that could be interesting, would be to allow the administrator to limit access to some (or all) resources to a certain group of people, by registering their Bluetooth IDs,

or by requiring pass codes or passwords. A last suggestion would be to make it possible to request more than one file with a command. For example some commands could send a whole image album or a subset of it, allowing users to download more files without needing to send a command for each file they want.

With further research and newer versions of the Bluetooth technology, hopefully Bluetooth broadcasting will be a useful addition to electronic communications in the future!

Part VIII: Glossary

26 Glossary

AAC – Advanced Audio Coding

A file type for storing audio files, similar to the widely known MP3. It was created to be a better alternative to MP3, but it has yet to gain popularity.

Baloo

This is the name used for the test broadcasting system that was developed in this project.

client

When used as a technical term, *client* means an application/software which allows users to connect to a *server* for fetching or giving information. Connections can be done over the Internet, or over other communication protocols like Bluetooth.

device discovery (Bluetooth)

The process of searching for other Bluetooth devices nearby. The devices must have turned Bluetooth on and must be set as “Discoverable”.

kbps (kilobit per second)

A measure on transfer speeds for digital information, 1 kbps equals 1 000 bit per second, or 125 bytes per second.

Mbps (Megabit per second)

A measure on transfer speeds for digital information, 1 Mbps equals 1 000 000 per second, or 125 000 bytes per second.

MMS – Multimedia Messaging Service

A service for sending text messages with multimedia content over the mobile phone network using mobile phones (or eventually other devices).

MP3 – MPEG-1 Audio Layer 3

A popular audio encoding format.

NFC – Near Field Communication

A wireless communication technology based on RFID, but with focus on security. Nokia, Sony and Philips

PDA – Personal Digital Assistant

Small handheld computers, usually with a touch screen for input.

RFID – Radio Frequency Identification

RFID is a technology for using radio waves for automatic identification. Usually, small RFID radios are built into RFID tags, and these tags can store a small amount of information, usually an identification code, and the tags can then be read using a RFID reader.

server

An application which allows *client* software to connect to it in order to transfer information.

Smart phone

An advanced mobile phone with PDA functionality.

SMS – Short Message Service

A service for sending short text messages between cell phones (or other devices) over the mobile phone network.

Part IX: References

27 Information resources

1. Bluetooth SIG: The Official Bluetooth Membership Site - <http://www.bluetooth.org> (Last updated: 2006)
2. Alf Inge Wang: MOBILE Work Across Heterogeneous Systems - <http://www.mowahs.com/> (Last updated: 2004)
3. Victor R. Basili - The Experimental Paradigm in Software Engineering (Experimental Software Engineering Issues: Critical Assessment and Future Directions, Proc. Int'l Workshop, Dagstuhl Castle, Germany, September 14th-18th 1992, pages 3–12)
4. Filter UK: BlueCasting website - <http://www.bluecasting.com/> (Last updated: 2006)
5. Dody Tsiantar - Getting on Board (Time, April 10 2006, bonus section A1-A4)
6. Eyal de Lara nad Keith Farkas - New Products (IEEE Pervasive Computing, Volume 4, Issue 4 (October 2005), pages 4-7)
7. LondonDev: BroadTooth website - <http://www.londondev.com/broadtooth/> (Last updated: 2005)
8. Midletsoft: Jellingspot website - <http://www.jellingspot.com/> (Last updated: 2006)
9. Midletsoft: Midletsoft website - <http://www.midletsoft.com> (Last updated: 2004)
10. Peekablue: Peekablue website - <http://www.peekablue.com/> (Last updated: 2005)
11. Alterwave: Alterwave website - <http://www.alterwave.com/> (Last updated: 2006)
12. BlueBlitz: BlueBlitz website - <http://www.blueblitz.com/> (Last updated: 2006)
13. Agnieszka Zmijewska - Evaluating Wireless Technologies in Mobile Payments – A Customer Centric Approach (Proceedings of the International Conference on Mobile Business (ICMB'05), July 11-13 2005)
14. Paolo Bellavista - The ubiQoS Middleware for Audio Streaming to Bluetooth Devices (Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous'04), August 22-25 2004, 138-145)
15. Wikipedia: Bluetooth Air Interface - http://en.wikipedia.org/wiki/Bluetooth#Air_interface (Last updated: 2006)
16. 32feet.net forum: Multiple Bluetooth radios on the same host - <http://32feet.net/forums/thread/1289.aspx> (Last updated: 2006-05-25)
17. Alf Inge Wang, Michael Sars Norum - Issues related to Development of Wireless Peer-to-Peer Games in J2ME (First Conference on Entertainment Systems (ENSYS 2006), February 23-25 2006, pages 3-4)
18. Brian S. Peterson, Rusty O. Baldwin, and Richard A. Raines - Bluetooth Discovery Time with Multiple Inquirers (Proceedings of the 39th Hawaii International Conference on System Sciences, 2006)
19. IrDA: IrDA website - <http://www.irda.org/> (Last updated: 2006)
20. Wi-Fi Alliance: Wi-Fi website - <http://www.wi-fi.org/> (Last updated: 2006)
21. Wikipedia: 802.11 specifications - http://en.wikipedia.org/wiki/IEEE_802.11 (Last updated: 2006)
22. RFID Journal: How much information can an RFID tag store? - <http://www.rfidjournal.com/faq/18/66> (Last updated: 2006)
23. International Herald Tribune: Mobile providers resisting SOS alerts - <http://www.iht.com/articles/2006/01/10/business/warnings.php> (Last updated: 2006-01-10)
24. Wikipedia: GPRS - <http://en.wikipedia.org/wiki/GPRS> (Last updated: 2006)
25. ZigBee Alliance: ZigBee website - <http://www.zigbee.org/> (Last updated: 2006)

26. Wikipedia: ZigBee protocols - <http://en.wikipedia.org/wiki/ZigBee#Protocols> (Last updated: 2006)
27. The Bunker: Bluetooth security flaws - <http://www.thebunker.net/security/bluetooth.htm> (Last updated: 2004-10-14)
28. In The Hand Ltd: 32feet.NET website - <http://32feet.net/> (Last updated: 2006)
29. Philippe Kruchten - Architectural Blueprints — The “4+1” ViewModel of Software Architecture (IEEE Software 12 (6), November 1995, pages 42-50)

28 List of illustrations and tables

Illustration 1: Nokia Bluetooth headset.....	2
Illustration 2: The Bluetooth logo.....	2
Illustration 3: People within the area of the broadcast can receive it.....	9
Illustration 4: Receiving a broadcast from a presentation.....	10
Illustration 5: The Broadtooth logo.....	12
Illustration 6: The Jellingspot logo.....	12
Illustration 7: Bluecasting logo.....	13
Illustration 8: Peekablue logo.....	14
Illustration 9: Alterwave logo.....	14
Illustration 10: BlueBlitz logo.....	15
Illustration 11: BlueBlitz broad-casting station.....	15
Illustration 12: A mobile phone.....	17
Illustration 13: A PDA device.....	18
Illustration 14: A laptop.....	19
Illustration 15: Bluetooth logo.....	21
Illustration 16: Nokia Bluetooth keyboard.....	21
Illustration 17: Official Wi-Fi logo.....	24
Illustration 18: The ZigBee Alliance.....	27
Illustration 19: Scenario 1 - Accessing a broadcast.....	54
Illustration 20: Scenario 2 - Adding a resource to a broadcast.....	54
Illustration 21: Class diagram for Baloo.....	55
Illustration 22: UML activity diagram for a running broadcast.....	57
Illustration 23: Main screen.....	62
Illustration 24: Manage broadcasted resources.....	65
Illustration 25: Statistics example: Hourly statistics.....	67
Table 1: Wireless technologies comparison chart.....	28
Table 2: Baloo's layered architecture (the bold layers are the layers that are within Baloo).....	53

Copyright notes

- The Bluetooth word mark and logos are owned by the Bluetooth SIG, Inc. Usage of such marks in this report was granted because they are used for internal purposes (University-related work).
- Images of various Bluetooth products are fetched from the Nokia Norway Media Archive (<http://www.bildearkiv.nokia.no/Nokia/loginAnonymous.do?lang=no>). These images are allowed to be used in public documents as long as the source is stated.
- Other logos are trademarks owned by their respective owners.