Guiding the Blind with Bluetooth and JAVA

GRADUATE PROJECT TECHNICAL REPORT

Submitted to the Faculty of
The Department of Computing Sciences
Texas A&M University - Corpus Christi
Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By

Rockwell Martin
Spring 2008

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. David R. Thomas
Committee Member

Dr. Dulal Kar
Committee Member
Guiding the Blind with Bluetooth and JAVA

GRADUATE PROJECT TECHNICAL REPORT

Submitted to the Faculty of
The Department of Computing Sciences
Texas A&M University – Corpus Christi
Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

By

Rockwell Martin
Spring 2008

Committee Members

Dr. Mario Garcia
Committee Chairperson

Dr. David R. Thomas
Committee Member

Dr. Dulal Kar
Committee Member
ABSTRACT

With the advancement in technology, life is becoming increasingly easier for the physically challenged individuals. Technology is being used to help, the cripple to walk, the deaf to hear and the mute to speak in some way. Research is already in progress, to assist and help blind people “read” off of a label in a super market. It is a giant leap, when with the help of technology; blind people are directed to find the way to their destination.

The Bluetooth technology is getting very more popular and accessible. This project uses Bluetooth technology along with combination of cellular phone technology to “direct” a blind person to their destination. A blind person possessing this Bluetooth enabled cell phone is helped to walk around campus. When he approaches key locations, where there are Bluetooth enabled device installed. This installed device starts to recognize this cell phone held by blind person, as he comes in within radius of ~10 meters of the installation point. The device then starts to communicate with the cell phone. Also it lets him/her know where is he/she is currently positioned, as well as, to which building is coming up ahead. Since Bluetooth network systems always remains actively on, all this help is accomplished without any interaction of the blind person.
TABLE OF CONTENTS

Abstract..........................................................................................................................ii

Table of Contents.........................................................................................................iii

List of Figures...............................................................................................................v

1. Background and Rationale.......................................................................................1
   1.1 Introduction....................................................................................................1
   1.2 Brief History of Cellular Phone.................................................................2
   1.3 Bluetooth Technology....................................................................................4
   1.4 Java in Portable Devices................................................................................9
   1.5 Integrating Java, Bluetooth and Wireless Technology.........................11
   1.6 Project Objective.........................................................................................12

2. Narrative................................................................................................................15

3. System Design and Implementation.....................................................................18
   3.1 Getting Started............................................................................................18
   3.2 Hardware/Software.......................................................................................18
   3.3 High-Level System Diagram.......................................................................20

4. Test Cases and Evaluation...................................................................................23

5. Future Work............................................................................................................28
   5.1 Future Work...............................................................................................28

6. Conclusion..............................................................................................................29
## LIST OF FIGURES

| Figure 1.1 | Nokia Mobira Senator and Motoroal Dyna-Tac | 3 |
| Figure 1.2 | Bluetooth Protocol Stack | 5 |
| Figure 1.3 | Scatternet Comprising Three Piconets | 8 |
| Figure 1.4 | Master and Slave Formation in a Scatternet | 8 |
| Figure 1.5 | Two Computers Communicating via Bluetooth | 13 |
| Figure 1.6 | Cellphone and a Computer Communicating via Bluetooth | 13 |
| Figure 1.7 | Simulated Environment Within a Computer | 14 |
| Figure 2.1 | Braille | 15 |
| Figure 2.2 | Portable GPS Device | 16 |
| Figure 2.3 | Bluetooth Signal Sent Via Cellular Device | 17 |
| Figure 3.1 | Bluetooth Use Cases | 21 |
| Figure 3.2 | Bluetooth Server and Client Activities | 22 |
| Figure 4.1 | Test Case 1: Program scans and evaluates the absence of any buildings nearby | 24 |
| Figure 4.2 | Test Case 2: Program scans and communicates with the center for instructions | 25 |
| Figure 4.3 | Test Case 3: Program scans/searches and discovers student services center | 26 |
1. BACKGROUND AND RATIONALE

1.1 Introduction

We all wake up each day with our sound mind and body. Have you ever imagined what it must be like to wake up blind? Or have you ever wondered how must it feel like to go through the day without being able to see? I can’t even begin to imagine what kind of struggles and challenges blind people go through each day to accomplish simple tasks which we all take for granted. For a long time people have been developing ways to help blind people lead an ordinary life. In 1821 Luis Braille invented a system to help a blind person read and write. It revolutionized the written communication for the blind [Wikipedia 2007]. Ever since then people have been coming up with devices, gadgets and applications which help improve the quality of life of the sight impaired.

Imagine a blind person walking to their destination without the help of another person. A project already underway is helping a blind person shop without the help of another human being. The project is named “Trinetra” [Bluetooth 2007]. Trinetra aims to develop cost-effective, Smartphone-enabled assistive technologies to provide people with an enhanced quality of life in their daily activities. The project is in its infancy but a prototype is already implemented in the shopping center in Carnegie Mellon University. All the products in the store are bar coded and the blind person has a Bluetooth enabled cellular phone and a pen shaped bar code reader. When the barcode reader scans the barcode on the product, the information is transmitted to the Bluetooth enabled cell phone. The cell phone has access to the Internet. The cellular phone
accesses a database which has the bar code information of the product. This information is then transmitted back to the phone. The information then using a software program is converted from a text format to a speech format. Thus, finally in audio format, the information then is relayed to the blind person. So all this is accomplished without human interaction.

If a blind person can “read” using the latest technology available, then the same technology can surely help him/her walk to their destination. This project assesses, what can actually be done in terms of how feasible, user-friendly and effective it is. The prototype was something on campus. Bluetooth enabled devices are located at some key location and or at the building entrances. The Bluetooth enabled cell phone was registered on the campus Personal Area Network (PAN). The area of communication for Bluetooth is approximately 10 meters. When a blind person possessing cell phone comes within that area; the device installed at that location starts automatically to communicate with the cell phone. It lets the person know, their current positioning and also as to which location they are approaching. The device starts sending this message via text format, which is then translated into speech using some text-to-speech conversion software. Products by different companies, such as NeoSpeech, Digitalfutresoft and Nuance were looked into but eventually, a database was used consisting of predetermined locations which were stored in the local file system as audio files.

1.2 Brief History of Cellular Phones

Of course all this technology would have been useless if the cell phone had not transformed from heavy bulky thing to carry to a small device weighing few ounces which can be carried in a pocket. One of the earliest cell phone by Nokia weighted almost 21 pounds,
making it not a real portable phone. It was really designed to be used in cars. Motorola came out with their cell phone in 1973. It weighed around 2.5 pounds, had 30 circuit boards, talk time of 35 minutes and a recharge time of 10 hours. Figure 1.1 shows the early models of the cell phones [MSN.com 2007].

![Figure 1.1 Nokia Mobira Senator and Motoroal Dyna-Tac](https://www.msn.com/en-us/2007)

Cellular phones can be divided into two big classes, 2\textsuperscript{nd} and 3\textsuperscript{rd} Generations phones. Commonly these phones are referred as a 2G or 3G phone. Most of the phones since the 1990’s come under the 2\textsuperscript{nd} generation. 2G phones work on the Global Systems for Mobile communications (GSM). They use digital circuitry for their communications. The old cellular phones worked on the analog signals. The digital system gave the network more quality and security. The 2G phones were much smaller and lighter than their older counterparts, thus making them truly portable. The older phones were heavier and needed larger batteries.

The third generation (3G) is the latest technology in the cellular phones. Standards of 3G are usually different depending on the network. It mostly depends on the network and the cell phone providers. The two most important requirements for a 3G include: 2 megabits of maximum data
rate indoors and 384 kbits for outdoor use. Currently certain audio visual content available for download on a cell phone requires high speed data rate transfer. [Tech-faq.com 2007].

When Cell phones first came out they were not accessible to common people since they were very expensive. So only the business community was being catered by that technology. As the years progressed and the technology advanced, they started to become cheaper, smaller and more powerful in terms of what they could do. Today the technology has developed so advanced that you are only limited by your imagination as to what can be done with the cell phone.

1.3 Bluetooth Technology

The word "Bluetooth" is taken from the 10th century Danish King Harald Bluetooth. King Bluetooth had been influential in uniting Scandinavian Europe during an era when the region was torn apart by wars and feuding clans. The founders of the Bluetooth SIG felt the name was fitting because Bluetooth technology was first developed in Scandinavia, and Bluetooth technology is able to unite differing industries such as the cell phone, computing, and automotive markets. Bluetooth wireless technology simplifies and combines multiple forms of wireless communication into a single, secure, low-power, low-cost, globally available radio frequency (Bluetooth Protocol Stack) [BlueTomorrow.com 2007]. The high-level view of the architecture of the Bluetooth protocol stack looks like figure 1.2
The responsibilities of the layers in this stack are as follows:

- The **radio layer** is the physical wireless connection. To avoid interference with other devices that communicate in the ISM band, the modulation is based on fast frequency hopping. Bluetooth divides the 2.4 GHz frequency band into 79 channels 1 MHz apart (from 2.402 to 2.480 GHz), and uses this spread spectrum to hop from one channel to another, up to 1600 times a second. The standard range is 10 cm to 10 m, and can be extended to 100 m by increasing transmission power.

- The **baseband layer** is responsible for controlling and sending data packets over the radio link. It provides transmission channels for both data and voice. The baseband layer maintains Synchronous Connection-Oriented (SCO) links for voice and Asynchronous Connectionless
(ACL) links for data. SCO packets are never retransmitted but ACL packets are, to ensure data integrity.

SCO links are point-to-point symmetric connections, where time slots are reserved to guarantee timely transmission. A slave device is allowed to respond during the time slot immediately following an SCO transmission from the master. A master can support up to three SCO links to a single slave or to multiple slaves, and a single slave can support up to two SCO links to different slaves.

Data transmissions on ACL links, on the other hand, are established on a per-slot basis (using slots not reserved for SCO links). ACL links support point-to-multipoint transmissions. After an ACL transmission from the master, only a slave addressed specifically may respond during the next time slot; if no device is addressed, the message is treated as a broadcast.

- **The Link Manager Protocol (LMP)** uses the links set up by the baseband to establish connections and manage piconets. Responsibilities of the LMP also include authentication and security services, and monitoring of service quality.

- **The Host Controller Interface (HCI)** is the dividing line between software and hardware. The L2CAP and layers above it are currently implemented in software, and the LMP and lower layers are in hardware. The HCI is the driver interface for the physical bus that connects these two components. The HCI may not be required. The L2CAP may be accessed directly by the application, or through certain support protocols provided to ease the burden on application programmers.
- The Logical Link Control and Adaptation Protocol (L2CAP) receives application data and adapts it to the Bluetooth format. Quality of Service (QoS) parameters is exchanged at this layer. [Mahmoud 2003]

Bluetooth wireless technology is a short-range communications technology intended to replace the cables connecting portable and/or fixed devices while maintaining high levels of security. The key features of Bluetooth technology are robustness, low power, and low cost. The Bluetooth specification defines a uniform structure for a wide range of devices to connect and communicate with each other. Bluetooth technology has achieved global acceptance such that any Bluetooth enabled device, almost everywhere in the world, can connect to other Bluetooth enabled devices in proximity. Bluetooth enabled electronic devices connect and communicate wirelessly through short-range, ad hoc networks known as piconets. Each device can simultaneously communicate with up to seven other devices within a single piconet. Each device can also belong to several piconets simultaneously. Piconets are established dynamically and automatically as Bluetooth enabled devices enter and leave radio proximity [Bluetooth.com 2007]. Figure 1.2 and Figure 1.3 illustrate the formation of piconets.

A fundamental Bluetooth wireless technology strength is the ability to simultaneously handle both data and voice transmissions. This enables users to enjoy variety of innovative solutions such as a hands-free headset for voice calls, printing and fax capabilities, and synchronizing PDA, laptop, and mobile phone applications to name a few [Bluetooth.com 2007].
Figure 1.3 Scatternet Comprising Three Piconets [Sun.com 2000]

Figure 1.4 Master and Slave Formation in a Scatternet [Lam 2004]
1.4 Java in portable devices

Connected, personalized, intelligent information appliances are becoming increasingly important in our business and private lives. These appliances, which include devices such as cell phones, two-way pagers, personal organizers, screen phones, and point of service terminals, have many things in common. But they are also diverse in features, form, and function. They tend to be special-purpose, limited-function devices, not the general-purpose computing machines we have known in the past. The number of these information appliances is increasing rapidly. Today, there are roughly 2.2 billion cell phone subscribers worldwide. Experts predict that will jump to 3 billion by the end of this year [Reardon 2007]. Compare this to the installed base of personal computers, which at the beginning of 2000 was around 311 million worldwide [Sun.com 2000].

To meet the demand for information appliances in the rapidly developing Consumer and embedded markets, Sun has extended the scope of Java technology with the introduction of Java™ 2 Platform, Micro Edition (J2ME™). The versatility of the Java application development environment is now enabling the development of many new and powerful information appliance products. Java technology enables users, service providers, and device manufacturers to take advantage of a rich portfolio of application content that can be delivered to the user’s device on demand, by wired or wireless connections. The main benefits of CLDC (Connected limited device configuration) devices involve:

• Cross-platform work is transferred between CLDC and other devices.
• Dynamic content is determined by user experience, and information transfer between CLDC and other devices.
• Security

• Developer community

The developer talent needed for these devices already exists and is readily available for CLDC devices [Sun.com 2000]. Users want the ability to purchase economically-priced products with basic functionality and then use them with ever-increasing sophistication. In order to support this kind of flexibility and customizable deployment demanded by the consumer and embedded market, the J2ME architecture is designed to be modular and scalable. This modularity and scalability are defined by J2ME as three layers of software built upon the Host Operating System of the device:

• **Java Virtual Machine.** This layer is an implementation of a Java virtual machine that is customized for a particular device’s host operating system and supports a particular J2ME configuration.

• **Configuration.** The configuration is less visible to users, but is very important to profile implementers. It defines the minimum set of Java virtual machine features and Java class libraries available on a particular “category” of devices representing a particular “horizontal” market segment. In a way, a configuration defines the “lowest common denominator” of the Java platform features and libraries that the developers can assume to be available on all devices.

• **Profile.** The profile is the most visible layer to users and application providers. It defines the minimum set of Application Programming Interfaces (APIs) available on a particular “family” of devices representing a particular “vertical” market segment. Profiles are implemented “upon” a particular configuration. Applications are written
“for” a particular profile and are thus portable to any device that “supports” that profile. A device can support multiple profiles [Sun.com 2000].

1.5 Integrating Java, Bluetooth and Cellular Technology

The Java2 MicroEdition (J2ME) platform joins the Bluetooth technology to be one of the most exciting offerings in the wireless industry. J2ME was defined by Sun Microsystems to meet the new needs of Java developers working on consumer and Small embedded systems. J2ME itself is not a specification, but a group of them Defining how Java technology is upon devices with few resources compared to a PC. This platform is portable, so applications follow the Java philosophy “once written run Anywhere”. It appears as a tool to let us write custom applications and run them on Mobile Bluetooth enabled devices. Devices using this Java Platform are: PDAs, cell phones, television set top boxes, remote telemetry units, many other such application devices and also other embedded devices. These are heterogeneous devices regarding processor power, memory, persistent storage and user interface. It is difficult to provide an optimal functionality for all these embedded devices due to this heterogeneity.

There was a first division of the devices into two sections considering the resources above mentioned, but not taking into account the function or use of the device. For the latter reason, there was a subdivision into classes of devices. There are two J2ME configurations corresponding to the former division above mentioned: Connected Device Configuration (CDC) and Connected Device Limited Configuration (CDLC), each one providing a JVM (Java virtual machine) and different libraries and APIs in order to create a run-time programming
environment for a group of devices. Configurations provide a common denominator subset of 
Java suitable for the characteristics of devices to which they are intended. Within a 
configuration, there is still much heterogeneity in aspects such as user interface, function and 
usage. Configurations do not specify the user interface toolkit and persistent storage APIs. This 
is the task of profiles [Zarceno 2004].

A profile is a set of JAVA APIs for a particular class of device, considering it’s 
Function, such as mobile phones. It is built over the platform provided by the 
Correspondent configuration. It is meant to couple this configuration. The Foundation 
Profile and the Mobile Information Device Profile (MIDP), correspond to CDC and 
CDLC respectively. For example, MIDP 2.0 joins CLDC to provide a run-time 
Environment. J2ME defines a small core API defines a core API to be implemented in every 
J2ME compatible device, deployed in different configurations. The next sets of Java classes are 
to be in every configuration: java.util and java.lang [Zarceno 2004].

1.6 Project Objective

The object of the project was to develop a prototype or a mimicking working piconet or a 
pair of Bluetooth enabled devices which are able to communicate with each other. Since this 
project can grow to be a monster, it was started on a small scale and assessed for its feasibility 
and effectiveness before expanding it on a large scale.

The prototype was a couple of Bluetooth/JAVA enabled devices. One of the devices was 
a cell phone while the other was a computer with Bluetooth (Figure 1.5 and Figure 1.6). Both 
devices successfully communicated with each other. Hence, any 2 such devices, which have 
Bluetooth and JAVA, will be able to communicate with each other.
Figure 1.5 Two Computers Communicating Via Bluetooth

Figure 1.6 Cellphone and a Computer Communicating Via Bluetooth
Figure 1.7 Simulated Environments Within a Computer
2. NARRATIVE

People have been coming up with countless ways with technology to help physically challenged people cope with day to day life. With the development of Braille, blind people can read. The markings outside rooms in buildings have Braille (Figure 2.1) which blind people can touch with their fingers and “read” what that room is.

![Braille Chart](Figure 2.1 Braille [Quido 2007])

Projects are underway which are helping blind people shop without anyone’s help. The blind person carries with him a Bluetooth enabled cell phone and a Bluetooth enabled bar code reader. When he/she scans a product in a store with the code reader, the reader sends the data to the cell phone. The cell phone accesses the database on Internet to find out the details of the product and relays that to the blind person using text-to-speech software.

Another technology which helps people to navigate is GPS (Global Positioning Systems). In addition, we now have portable GPS (Global Positioning System) devices, which tell the position/location based on the latitudes and longitudes (Figure 2.2). Devices like these can certainly be utilized to help a blind person with directions, but these systems have their
limitations. These devices work hand in hand with road maps. When destination information is input, these devices are able to navigate as to how to get there by relevant road directions. Being on some big university campus and wanting to get directions from building to building, this GPS device won’t do much good. The functionality of this project begins, where the functionality of a GPS device stops or diminishes.

![Portable GPS Device](image)

Figure 2.2 Portable GPS Device [Buy 2007]

True functionality of GPS devices diminishes once you’re inside a campus. GPS devices work on street addresses. For example, if input information is fed into the GPS device as to how to go from 5151 staples to 6300 ocean drive (TAMUCC), it gives turn by turn direction as to how to get there but once inside the campus it does not tell, as to how to navigate from one building to next or which building is being approached, and this is where this project takes over.

This project navigates a blind person in a whole new way. Imagine the campus has some kind of Bluetooth devices at key locations. The Bluetooth enabled cell phone of the blind person is registered on the campus Personal Area Network (PAN). As soon as the blind person comes on campus and within 10– meters of the device, the device then sends a message in a text form to the cell phone communicating the position of the blind person of his current location. The cell
phone after receiving the text message, translates the received text message into an audio
message using text-to-speech software. Database containing pre-recorded audio files is being
used to accomplish this. Either way the blind person “hears” about his current position/ location.
For example, he/she might be approaching the building which houses the admissions offices; in
that case the device located at the entrance will scan/detect the cell phone of the blind person and
will transmit an audio “message” telling him, “that he is approaching the admissions office”
(Figure 2.3). Depending on as to how accurate the Bluetooth network is, the “device” may also
let them know how far they are from the approaching building. These devices when placed very
strategically send the accurate information to the person.

Figure 2.3 Bluetooth Signal Sent by Cellular Device.
3. SYSTEM DESIGN AND IMPLEMENTATION

3.1 Getting Started

To get started, it was decided on which scale was the system going to be implemented. If not careful, it can certainly get out of hand and become undoable. Since this required a lot of integration between different kinds of hardware and software products, the devices were chosen very carefully. For the project to work, different hardware/software needed to be compatible with each other. For the project to be successful the devices have to be able to successfully communicate between a cellular phone/PDA and a PC, both of which are Bluetooth and JAVA enabled. The decision to choose for a prototype depended upon number of factors, including money, scalability, availability, feasibility and compatibility.

After much research a Motorola phone was picked for the project. It is a Motorola Q9H. It is a 3\textsuperscript{rd} Generation phone with all the bells and whistles, more importantly it has the JAVA Bluetooth API JSR-82. The operating system of the phone is windows 6.0. It has numerous Bluetooth profiles which can be used to develop various other applications but for this project only JSR-82 was used. The other system which communicates with the Motorola phone is the PC (with Bluetooth chip by DELL) running windows XP.

3.2 Hardware/Software

For the cellular phone/PDA to work it has to be Bluetooth enabled and JAVA compatible. Not only it has to be JAVA compatible but it has to be running JSR-82 (Java Bluetooth stack). Hence the Motorola Q9H was selected for the project. The PC is a windows XP machine which has a Bluetooth chip. For the PC to be able to use the JAVA JSR-82 API, JSR-82
library was installed from a third party (freeware/opensource) which works with the Microsoft Bluetooth stack (XP Bluetooth drivers).

JSR-82 is a standard defined by the Java Community Process for providing a standard to develop Bluetooth applications in Java. It is an open and non-proprietary standard for developing Bluetooth applications. The JSR-82 API hides the complexity of the Bluetooth protocol stack, by exposing a simple set of Java API’s [Mark 2007].

To develop a Bluetooth application in Java using the JSR-82, the following were needed

1. **JSR-82 Compliant Bluetooth Stack.**

   Need a JSR-82 Implementation to successfully develop and test a Java Bluetooth application. Either needs to have a simulated environment like the Sun Java Wireless Toolkit, or a real Java Bluetooth Stack like ElectricBlue or Avelink. The simulator allows testing the Bluetooth application in a simulated environment, with no access to real Bluetooth device. Since this project works on real devices instead of a simulator BlueCove was installed on the PC. BlueCove is a JSR-82 J2SE implementation that currently interfaces with the Mac OS X, WIDCOMM, BlueSoleil and Microsoft Bluetooth stack found in Windows XP SP2 and newer. Originally developed by Intel Research and currently maintained by volunteers [BlueCove 2008].

2. **Local Bluetooth Device.**

   When using a simulator, this is not needed. But to test the application in real world over a java Bluetooth stack, a Bluetooth enabled system is needed. This can either be a
Bluetooth USB Dongle plugged in the system, or the internal built in Bluetooth device.

The java application that was written runs over this Bluetooth device, and this Bluetooth device is called as the ‘LocalDevice’ in JSR-82 [Mark 2007]

The JSR-82 API provides you the options to do the following

1. Manage the Local Bluetooth Device settings.
2. Scan to discover other Bluetooth devices in the neighborhood.
3. Search for Bluetooth devices on the discovered Bluetooth devices.
4. Connect to any of those Bluetooth service and communicate with it.
5. Register a Bluetooth service on the Local Bluetooth Device, so that other Bluetooth devices can connect to it.
6. Manage and control the communication connections.
7. Provide the security to all of the above options [Mark 2007].

Once the Java and Bluetooth is configured to talk to each other. Next step is to figure out as to how to convert text to speech. Third party software product by companies such as NeoSpeech, Digitalfutresoft and Nuance were reviewed but none were compatible with our hardware. So for the project it was decided to use pre-defined audio files with information about the different buildings. The predefined audio files were made with the help of ATT research labs TTS (test-to-speech) software

3.3 High-level System Diagram

The system was divided primarily in 3 parts (Figure 3.1 and Figure 3.2)

Initialization - Bluetooth-enabled application, server or client, will first initialize The Bluetooth stack.
**Client** - Client consumes remote services. It first discovers any nearby devices, then for each discovered device it searches for services of interest.

**Server** - Server makes services available to clients. It registers them in the Service Discovery Database (SDDB), in effect advertising them. It then waits for incoming Connections, accepts them as they come in, and serves the clients that make them.

Finally, when the service is no longer needed the application removes it from the SDDB [Ortiz 2005].

---

![Figure 3.1 Bluetooth Use Cases [Ortiz 2005]](image)
Figure 3.2 Server and Client Activities [Ortiz 2005]
4. TEST CASES AND EVALUATION

The test environment was a Smart phone and a PC. The test cases tested the connectivity of the programs to see if these two were able to connect and exchange data with each other. Obviously, the precise “range” of the connectivity was not calculated. The range of a typical Bluetooth connection is approximately 10 meters (~30 feet), which was critical in making this a practical solution for the blind.

To start the testing procedures, it was made sure that the Microsoft stack was installed on the XP station. Then the Bluetooth library JSR-82 on the PC was installed correctly and was working fine. Lastly, the JAVA program detects the Smartphone, when the phone was set on Bluetooth discoverable mode. The devices successfully detected and communicated with each other within 10 meters, hence it will communicate with any other Bluetooth JAVA enabled devices within its vicinity.

**Test Case 1**: Program scans and discovers no device (no buildings nearby)

1. Booted up the computer and brought up the console to run java programs. At the prompt made sure that the correct classpath was set to run the java program. (At this point the cell phone power switch was turned to “off” position).
2. Ran the program by giving the command: `java SearchForBuilding`
3. The program scanned for the device but could not find it (since the phone was turned “off“Figure 4.1).
Test Case 2: Program scans and discovers Motorola Q9H (center for instruction Building)

1. Turned the power switch to “on” position of the cell phone AND in the Bluetooth settings changed the setting to “make cell phone discoverable”

2. Initiated the java program again on the computer with the command: java SearchForBuilding

3. The program scanned and found the device. Also it returned back with the hardware address and the name of the device.
4. The java program then proceeds to check a file which has a list of classes in which the student is registered.

5. The program matches the name of the building returned by the device and the name in the local file. It then returns an audio file associated with it. Basically telling the person in audio message as to which building he is approaching and telling them the floor and classroom information, as to where/if they have a class.

**Test Case 3**: Program scans/searches and discovers Motorola V330 (student services Center)

1. Disabled the discoverable bluetooth mode in Q9H (center for instruction)
2. Enabled discoverable mode in Motorola V330 (student services center).
3. Started the program again with java: SearchForBuilding
4. The program found the student services center building approaching.

Figure 4.3 Program scans /searches and discovers student services center

5. The java program then proceeds to check a file which has a list of classes in which the student is registered
6. The program matches the name of the building, returned by the device and the name in the local file and returns a audio file associated with it, basically telling the person an audio message as to which building he is approaching and telling them the floor and classroom information as to where/if they have a class
The pre-recorded messages were created with the help of ATT research labs (text to speech- TTS). They were then stored locally on the device and then called from the invoking procedure [Att.com 2008].
5. FUTURE WORK

5.1 Future Work

Since the Bluetooth technology is fairly new, there is still a lot of work to be done. Every day, there are new innovations on how to incorporate the use of Bluetooth in our daily life. The new cars are advertising the use of voice commands in the car, which are realized due to the Bluetooth technology. Bluetooth applications should not be used in some GPS applications, along with in conjunction with java. Bluetooth gives an approximation but not an exact distance. On Similar grounds, this project only gives an approximation and a general idea of where the person’s position/location is, but not an exact distance from the building. For example this application was not able to tell a person as to how far away in feet he/she is from the building.

As the future brings out more and more enhancements in Bluetooth technology we might see a big jump in the use of Bluetooth in more devices. Till that happens there is much room for the development in this area.
6. CONCLUSION

6.1 Conclusion

With the project working and installed in place, it was able to navigate a sight impaired person around the campus. The prototype was instrumental to make this project a viable one. When the JAVA program was initiated, it scanned and detected the Smartphone and received its hardware address and name. The program then initiated an Audio stream with which was heard on the speakers.

With technology advancing by leaps and bounds, people who are physically handicapped are reaping the benefits. Especially people who are sight impaired face new challenges everyday. With the development of technology the electronic gadgets are getting smaller and smaller and are becoming very powerful tools. The cellular phones are becoming a necessity in today’s life. These little devices are instrumental in helping a blind person these days. This project successfully takes this a step further. The blind person equipped with a phone, which has Bluetooth and is JAVA enabled technology can walk around a campus with little more ease. As soon as he comes within the radius of the device, which would be located at key positions and at building entrances, his phone receives a Bluetooth signal along with the information of the approaching building (sent by java via Bluetooth). That text information is then converted to audio format which they can listen to. The success of the project depended on many factors. Some questions still remain to be addressed. Was the speed with which information be transmitted is fast enough? Would the phone process and convert that information from text to
speech, fast enough to be meaningful? Would the signal from neighboring devices affect the communicating devices?

All these questions are important and only advancing forward with the project to an advance stage will determine if this would be a feasible, affordable and a viable solution for a sight impaired person. If nothing else this project at the least spear heads a direction leading towards finding a solution which “is definitely going” to be very helpful in the near future.
BIBLIOGRAPHY AND REFERENCES


import java.util.Vector;
import javax.bluetooth.*;
import sun.audio.*;
import java.awt.*;
import java.io.*;

public class SearchForBuilding {

    public static final Vector devicesDiscovered = new Vector();

    public static void main(String[] args) throws IOException, InterruptedException {
        final Object inquiryCompletedEvent = new Object();

        devicesDiscovered.clear();

        DiscoveryListener listener = new DiscoveryListener() {
            public void deviceDiscovered(RemoteDevice btDevice, DeviceClass cod) {
                String building = null;
                String buildingschedule = null;

                System.out.print("Bluetooth Device found \" + btDevice.getBluetoothAddress() + \"");
                devicesDiscovered.addElement(btDevice);

                try {
                    building = btDevice.getFriendlyName(false);
                    System.out.println(" Approaching " + building);

                    if (building.equals("center for instruction")) {
                        building = "cibuilding.wav";
                        buildingschedule = "cischedule.wav";
                    }
                }
            }

            public void inquiryCompleted() {
                devicesDiscovered.clear();
                inquiryCompletedEvent.notify();
            }
        }

        DiscoveryAgent agent = new DiscoveryAgent();
        agent.setInquiryMode(DiscoveryAgent.GENERIC_WRITE);
        agent.setDiscoveryListener(listener);
        agent.startInquiry();
    }
}
if (building.equals("Student Services Center")){
    building="ssc.wav";
    buildingschedule="sscschedule.wav";
}

if (building.equals("Science and Technology building")){
    building="sandt.wav";
    buildingschedule="sandt.wav";
}

if (building.equals("University Center")){
    building="uc.wav";
    buildingschedule="uc.wav";
}

if (building.equals("Faculty Center")){
    building="fc.wav";
    buildingschedule="fc.wav";
}

if (building.equals("The Library")){
    building="library.wav";
    buildingschedule="library.wav";
}

if (building.equals("University Health Center")){
    building="uhc.wav";
    buildingschedule="uhc.wav";
}

//building="cibuilding.wav";
playAudioFile(building);
playScheduleFile(buildingschedule);

//System.out.println(" Approaching " + btDevice.getFriendlyName(false));
} catch (IOException cantGetDeviceName) {
}
}

public void inquiryCompleted(int discType) {
    //System.out.println("Device Inquiry completed!");
}
System.out.println("\n");
synchronized(inquiryCompletedEvent){
    inquiryCompletedEvent.notifyAll();
}

public void serviceSearchCompleted(int transID, int respCode) {
}

public void servicesDiscovered(int transID, ServiceRecord[] servRecord) {
}

public void playAudioFile(String building) {
    String buildingName = building;
    AudioDataStream buildingAudio;

    try{
        FileInputStream fileinput = new FileInputStream( new File( buildingName) );
        AudioStream audiofileinput = new AudioStream( fileinput );
        AudioData audioIn = audiofileinput.getData();
        buildingAudio = new AudioDataStream( audioIn );

        AudioPlayer.player.start( buildingAudio );
        try{
            Thread.sleep(3500);
        } catch (Exception e){}
        AudioPlayer.player.stop(buildingAudio);
    }catch (IOException e){}

    public void playScheduleFile(String buildingschedule){
        String schedule = buildingschedule;
        }
AudioDataStream scheduleAudio;

try{
    FileInputStream fileinput = new FileInputStream( new File(schedule) );
    AudioStream audiofileinput = new AudioStream( fileinput );
    AudioData audioIn = audiofileinput.getData();
    scheduleAudio = new AudioDataStream( audioIn );

    AudioPlayer.player.start( scheduleAudio );

    try{
        Thread.sleep(5000);
    } catch (Exception e){}

    AudioPlayer.player.stop(scheduleAudio);
}

synchronized(inquiryCompletedEvent) {
    boolean started =
    LocalDevice.getLocalDevice().getDiscoveryAgent().startInquiry(DiscoveryAgent.LIAC, listener);

    if (started) {
        System.out.println( " ");
        inquiryCompletedEvent.wait();

        if (devicesDiscovered.size()>0){
            System.out.println(devicesDiscovered.size() + " Building(s) Found nearby...");
        }
    }
    else{

System.out.println("No Buildings Found nearby...");

String fileName = "nonefound.wav";
AudioDataStream noBuilding;

FileInputStream fileStream = new FileInputStream(new File(fileName));
AudioStream audiofileinput = new AudioStream(fileStream);
AudioData audioIn = audiofileinput.getData();
noBuilding = new AudioDataStream(audioIn);

AudioPlayer.player.start(noBuilding);
Thread.sleep(3500);
AudioPlayer.player.stop(noBuilding);

System.out.println("\n");