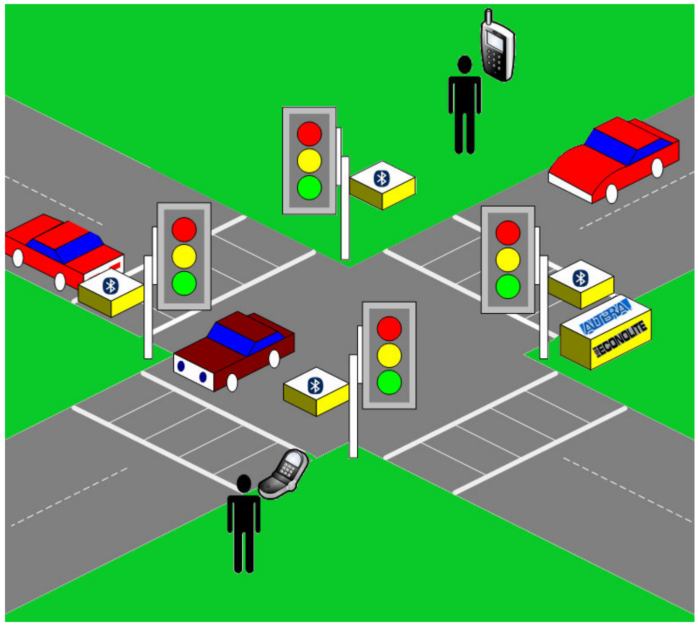
Universal Real-time Navigational Assistance

http://urna.projects.unoc.net/



Project Summary

Jack Baskin School of Engineering
University of California, Santa Cruz

Project Team

Shawn Bohonos
sbohonos@ucsc.edu
Computer Engineering
Multimedia

Alex Lee
alexlee@ucsc.edu
Computer Engineering
Networks

Amir Malik
amalik@ucsc.edu
Computer Engineering
Multimedia

Chung Thai
cthai@ucsc.edu
Electrical Engineering
Communications

Table of Contents

Project Overview

The Universal Real-time Navigational Assistance (URNA) design project is the second of a two-part project that has been proposed by Professor Roberto Manduchi. The first project deals with computer vision and entails the use of a mobile phone camera to allow a blind person to track physical symbols across a room or a street.

Our project deals specifically with mobile phone-based navigation using audio alerts as a navigational guide. The general idea is to use Bluetooth as an assistive technology for both visually impaired and sighted people. The system will consist of a series of short-range (between 5 and 20 meter radius) Bluetooth beacons that broadcast location-aware information to nearby Bluetooth devices. Any person equipped with a Bluetooth-enabled mobile phone will be able to take advantage of the system. Transmitters could be installed at traffic intersections to alert the visually impaired of the location of the crosswalk and when it is safe to cross. The goal of our project is to design and implement a functional pedestrian navigation system utilizing the ubiquity of Bluetooth technology in modern mobile phones.

We believe that a navigation system should benefit both sighted and visually impaired users. Due to the growing popularity of Bluetooth-enabled phones, our system could enable access to a wide audience. Ideally, we would like to see our system be capable of informing any user about pertinent information in a city, such as nearby bus routes and timetables, tourist attractions, and store directories. The technology need not be limited to only pedestrians, for example, even drivers could be alerted of nearby parking facilities.

The group was formally founded on October 5, 2006. We set up an online wiki (http://urna.projects.unoc.net/) early on to help our team track our research progress through the design phase of the project. The wiki also serves as a publicity site for our project, as well as internal documents, which are hidden from the public view. In addition, we use a CVS (Concurrent Versions System) repository to share code, documentation, and publicity materials such as artwork. We have overcome many challenges along the way, and anticipate many more.

System Hardware

Navigation Controller

The heart of our system is a PCB that we call NavCon, short for 'Navigation Controller' that will interface between the traffic controller and our Bluetooth modules. NavCon will take input signals from the traffic controller via R2-232 serial or Ethernet, and relay the data to the Bluetooth modules after processing. The system-level block diagram is shown in Figure 1 below. The Bluetooth modules are to be mounted at the four corners of an intersection atop an intersection's 'ped-heads'. Ped-heads are the electronic signs that signal to 'Walk' or 'Don't Walk' when a pedestrian is crossing an intersection.

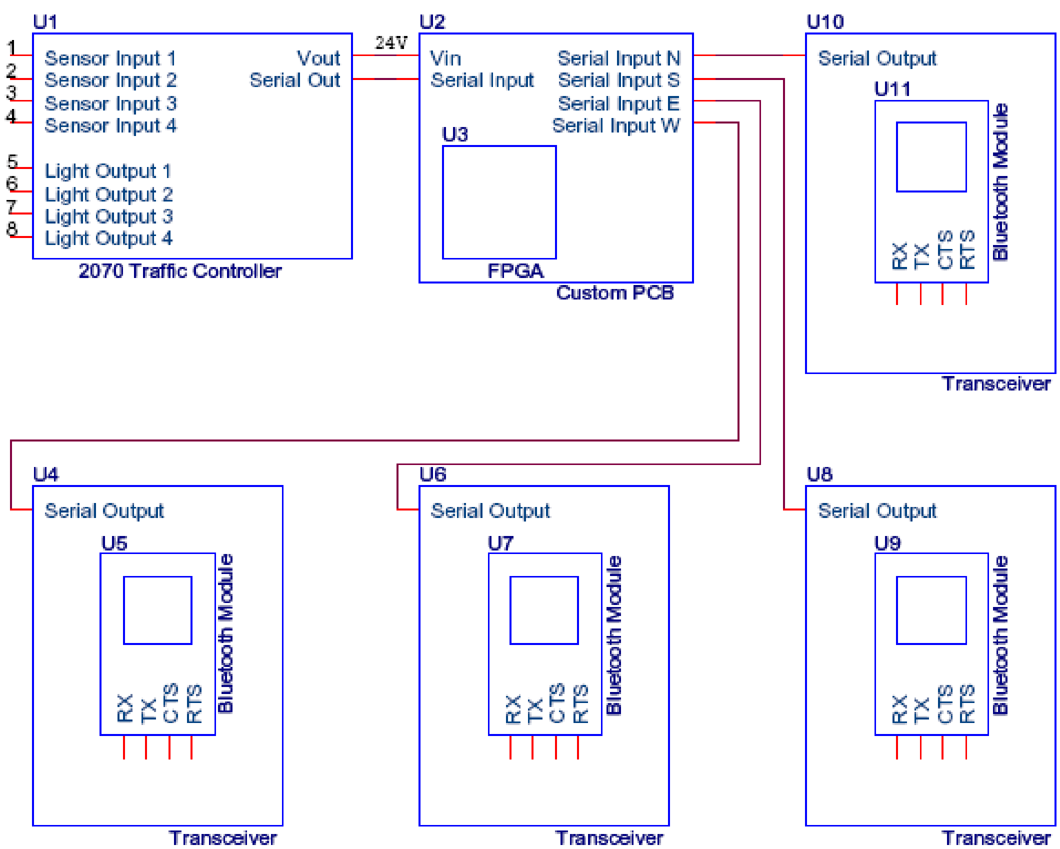


Figure 1. System-level block diagram.

Initial NavCon development has begun using Altera's Quartus II Integrated Development Environment and the Altera-Modelsim simulator software packages. Once we acquire the Nios II Development Board this week, we will transfer our Verilog code to the board which contains a Cyclone II FPGA that contains 33,216 logic elements. We plan to store audio and/or text in the non-volatile flash memory chip known as the EPCS or serial loader on the Nios II board that is normally used to store the FPGA's configuration file. We need the data to be retained even if there is a power outage, so for this reason we need the data stored in a non-volatile location. The audio and/or text files will contain information such as street names and traffic light status commands that will be sent to the Bluetooth modules and transmitted to the end user's mobile phone via Bluetooth. We will also be designing another custom PCB for the Bluetooth modules. Ideally, we will have four Bluetooth modules, so will need four of these PCB's printed. The Bluetooth PCB's will be used to mount the Bluetooth modules and accommodate for extra logic needed.

Bluetooth Modules

The Bluetooth modules are class 2 devices that have a 10 meter range and can support 7 simultaneously serial connections based on the type of firmware used. Our target data throughput for the system is 700kbps, since that is the bottleneck of the over-the-air (OTA) connection between a module and a mobile phone. We plan to implement our own external antenna design which will be omni-directional with a short-range 360 degree signal dispersion. The exact design details of the antenna are still yet to be determined, but we have met with Professor Peterson and he has volunteered to assist us with the design.

The modules that we will acquire will conform to the Bluetooth 2.0 specification. The 2.0 spec has numerous improvements over the older version 1.2. Version 2.0 has 3 times faster transmission speed, lower power consumption, a simplification of multi-link scenarios due to more available bandwidth and improved bit error rate performance. When searching for the Bluetooth modules to be used for our system, a few of the requirements in order of priority were as follows:

(a) Bluetooth 2.0

(b) DIP packaging for mounting so the chip could be re-used between development and actual implementation

(c) enough of a low-level firmware to be able to support 7 active separate serial connections

(d) availability / lead-time

(e) price

“In order to use Bluetooth wireless technology, a device must be able to interpret certain Bluetooth profiles. The profiles define the possible applications. Bluetooth profiles are general behaviors through which Bluetooth enabled devices communicate with other devices.”(1)

The profile that we will be utilizing is the Bluetooth Serial Port Profile (SPP). The Serial Port Profile allows Bluetooth devices to perform serial cable emulation. This is how we will have a connection to transfer data from the Bluetooth modules to our custom software on a mobile phone.

We have a few candidates for the Bluetooth module that we will use and will decide which to acquire the day that this report is to be submitted. Our primary candidate module is manufactured by A7 Engineering in San Diego that will possibly be donated for our project through their ‘Bluetooth Research Sponsorship Program’. They have a new version module that is set to hit the market either later this month or possibly early next year. One of our group members has been in communication with a lead contact at A7, and has consummated the possibility of having the pre-public versions of the module. Our second candidate module is a connectBlue module that meets the above listed requirements and is fairly priced. We will purchase two of these modules for the development phase if we do not hear back from A7 Engineering. Our third candidate module is the Linkmatic 2.0 out of the UK. It too, meets all of our requirements, but since the purchase would be made from overseas, shipping costs are high. Further considerations pertaining to the acquirement of the Bluetooth modules can be reviewed in the ‘Trade Studies’ section below.

Traffic Controller

The traffic controller that has been donated to us is an Econolite 2070 controller. We are not specifically concerned with the exact inter-workings of the controller, but our main requirement to allow it to be integrated into our system was that it needed to have a ‘timing plan’ preloaded into it that would act as a mock intersection for us. “The primary objective of a signal timing plan is to alternate the right-of-way among various phases at an intersection to provide for the orderly movement of traffic, minimize average delay to vehicles and pedestrians, reduce the potential for certain types of crashes and maximize the traffic capacity.”(2)

Econolite has kindly offered to provide us with a timing plan preloaded, as well as 24-7 tech support as though we were a customer.

Communications & Power

We've considered various methods of communication to interface our FPGA/PCB hardware to the Bluetooth modules over a distance of about 30 meters maximum. We discussed the ability to provide power to each individual Bluetooth module. Most manufacturers recommend RS-232, but we are transmitting over a significant distance, thus need a differential transmission scheme to help avoid EMI interference. We decided that the next logical candidate was to use RS-422. RS-422 is a differential transmission method, and can normally accommodate for higher BAUD rates than RS-232. This means that our custom PCB on the FPGA side will contain four RS-422 serial chips via 8P8C connectors as well as USB female connectors to accommodate for sending power to each Bluetooth module.

The power supply requirements for our application were simple and can be easily implemented without additional design complication. Each of the Bluetooth's transceivers will have a voltage regulator to filter undesirable voltage feeds into the device. The main controller board will have its power supply from the traffic light controller between 12 to 24 volts. It too will need some kind of voltage regulator so the amount of voltage supply will be between 5 to 10 volts depending on the FPGA voltage requirement and other chips on board (i.e. memory).

Software Development

In order for the system to be completely functional, a great deal of software needs to be written. For the Navigation Controller, this includes hardware description code in Verilog for the FPGA, and assembly and C code for the Nios II soft-processor. On the mobile phone side, an application will need to be developed that will communicate over Bluetooth with the Navigation Controller, and issue audio alerts when necessary.

NavCon

Because we are going to be using the Altera Nios II soft-processor, an operating system (OS) will be required. We are currently evaluating an embedded version of Linux, called µClinux, which we may use for the base OS. Because we will need to interface with custom Bluetooth modules, we may or may not need to write a Bluetooth driver for Linux, since the BlueZ stack already exists. Reusability depends on the modules we acquire, and how low-level they allow us to get with respect to the Bluetooth protocol.

The task of the NavCon server software is to communicate with up to four Bluetooth modules connected to the board and issue commands to each Bluetooth radio. In addition, by having the modules centrally controlled, we are able to track pedestrians as they pass from one crosswalk to the next. This will enable us to give more route-specific information if desired.

Essentially, the main NavCon server software will actively be scanning for new Bluetooth devices and attempt to connect to them. The PedNav Protocol, currently in the draft stage, will be used to communicate with mobile phones. Because Bluetooth's built-in security, such as pairing and encryption, will not be used due to implementation weaknesses on many devices, we must use our own security architecture. We are planning to implement a simple public-key infrastructure (PKI) that would enable us to have encrypted and authenticated communications in the application layer. One trusted authority would issue certificates for different NavCon installations around a city, and each installation of the phone software would know, upon connecting to a NavCon server, whether or not its public key has been signed (i.e. approved for public use) by the city.

PedNav

Although most of the logic of the system is contained in NavCon, the mobile phone software, known as PedNav is equally important. It will be the only interface with which the pedestrian needs to interact. PedNav is written in Java 2 Micro Edition (J2ME) due to its availability on all modern mobile phones and other devices. This allows us to have a single code base for millions of devices available on the market today. In addition, due to numerous Java Specification Request (JSR) standards, we are also able to access the Bluetooth hardware from Java applications.

The role of PedNav is to accept incoming connections from nearby NavCon systems. Proper authentication and security will be used to make sure that the NavCon system is trustworthy, through the use of certificates, which are installed alongside the application. For example, whenever a new NavCon-enabled traffic light comes into range, audio will be streamed to the phone and played back via a headset or the built-in speakerphone. The user will also be able to interact with the software by using the keypad on their phone. This would allow menu-based applications to be implemented, such as selecting from a list of bus routes or timetables.

Trade Studies & Challenges

One major challenge that we were presented with in the beginning of the design phase was figuring out how we were going to implement and test our system. After numerous group discussions and a key meeting with a lead traffic engineer in the City of Santa Cruz, we came to the conclusion that there would be too much liability in attempting to install our system at a real operating traffic intersection. This was mainly due to finding out that only Professional Engineers and technicians authorized and certified by the State are allowed to have access to traffic control equipment at an operational intersection. As an alternative, we decided to try to acquire a traffic light controller to use in lab to simulate an intersection.

Our initial contact with traffic engineering departments at a few cities in the Bay Area about obtaining a controller served to be very useful. Although we didn't have any luck with obtaining a controller in the beginning, we did receive responses from a few senior traffic engineers that were very helpful in providing us with information relative to our research. One such official was the senior traffic engineer in Fremont. He informed us that a new federal standard called the "2070 platform" is to be implemented across the country in the years to come. He suggested that it might be a good idea to target this platform for our project. After researching the 2070 standard and other traffic control protocols and standards, we decided that we would attempt to attain a 2070 controller.

We composed a listing of all major cities in California and in the United States according to highest population as a means by which to make support requests. We found the traffic engineering department contact information for most cities in the Public Works sections of the cities websites. We also wrote to the U.S. Department of Transportation (DOT) in Washington DC to make a request for support, and they replied with the recommendation to contact State DOT's, as well as provided the listing for the contact information for each state. Before writing to each of the cities and state departments, we researched the city and/or state and found any sort of interesting news relative to development in the area of transportation. Our idea was to not come across as sending a template to every city in the country. All together, we contacted about 150 city departments, state agencies and traffic controller manufacturers across the U.S. requesting the donation of a 2070 controller, or relevant information about controllers for our research. We found out that most state Department of Transportation agencies as well as local city traffic departments are not using the 2070 controllers yet; so to this end, we received many responses denying the ability to assist in our efforts. We did receive a response from Econolite, one of the largest manufacturers of 2070 controllers, that is willing to provide us with a controller and with technical support. Since none of our group members have a background in traffic engineering, we speculated that any technical assistance in this area will be beneficial to the advancement of the project. We collectively decided that we would accept the offer from Econolite, and they shipped it out to us this week!

Another major challenge that we were faced with was the attainment of the Bluetooth modules. We have been researching modules since the beginning of the project, and have finally compiled a list of candidates as mentioned above in the Bluetooth module section. One of the early modules that we considered was by Bluegiga called the WT12. It met all of the requirements that we needed except for the mounting. We requested and gained access to the Bluegiga Tech Forum to acquire documentation for the WT12 module that was previously our primary candidate due to features and lower price than the connectBlue OEMSPA333 (below). After reviewing the documentation, we spoke with the sole sales rep in the U.S., and found out further details about options for the mounting of the module for development vs. actual implementation. We met with Prof. Peterson to review these details and concluded that since it was a surface mount module (it looked like we might have been able to solder to them, since they protruded out more than other surface mount modules), we would need to implement a layout and have PCB's printed for the development phase. The module's cost $50 each, but we would need to spend an extra $50 for the prototyping PCB + time for design layout and turnaround, and then later have the real PCB's made for actual implementation. This led us to exclude this module from our list of possibilities.

We also contacted connectBlue in Sweden to inquire about obtaining their new OEMSPA333 Bluetooth modules, and also a distributor in Germany that advertised carrying them. The distributor was out of stock, and the manufacturer said that they did not have any available, but were shipping to the U.S. last week for the first time to a distributor called Future Electronics. We contacted Future Electronics in San Jose, but they said that they were not in the inventory computer yet and they were not very helpful beyond that. Next, we contacted the Rochester, New York office, but they said that we needed to talk to the local distributor in our area in San Jose. They told us that the head office was in Tennessee, but there was no listing for the Tennessee office, so we called the Massachusetts office and asked to be transferred to the Tennessee office. The Tennessee office does not accept phone calls from the public, so we were transferred to the corporate office in Montreal, Canada. We finally made contact with a representative there who was able to look into Future's receipt of the connectBlue modules. She informed us that there was an 8-10 week lead time, but that she had found 10 modules in Europe, and that there was the possibility of having two of them transferred to the U.S. for us to purchase.

The OEMSPA333 module has both a pin header and is surface mountable. This means that we can use it for both prototyping and development. The cost is $115 per module plus shipping from Tennessee.

After excluding the WT12 from our list, and not yet hearing back from Future Electronics in Montreal, we continued the search for a module that was not solely surface-mountable. We found the Linkmatic 2.0 from a distributor called RF Solutions in the UK, which uses Bluegiga's WT12 Bluetooth chip, but has a DIP package. The company has a listing on their website of zero in stock though. We also found another distributor for this module in Spain, but a challenge was that the website was in Spanish. They advertised that they spoke English, but upon calling them, we found out that this wasn't the case. One of our group members spoke to the rep in broken Spanish and found out that they had three of the Linkmatic modules in stock for $137 each + $86 shipping from Spain! We verified this shipping cost by checking the FedEx and UPS websites for an estimate of such a delivery. We also called RF Solutions in the UK and they informed us that they actually have two Linkmatic modules in stock at $126 each, even though this was not reflected on their website. We assume that the shipping will be about the same as from Spain.

We also found out that A7 Engineering has a Bluetooth Research Sponsorship Program for student projects using Bluetooth technology here in the U.S. The modules that they make are surface mountable, but we figure that if they are donated, we can spend the money to make PCB prototypes for development. We found out that to apply for the program, we needed to write up a proposal requesting support. They also asked that we take pictures of our lab on campus that we will be working in, that we advertise their logo, and that we make our final paper available on their website upon completion of our final report. We wrote up the proposal document, and sent it off and are waiting for a response.

With the above information, we prioritized our options as follows:

1. A7 Engineering
2. connectBlue
3. Linkmatic

The Future

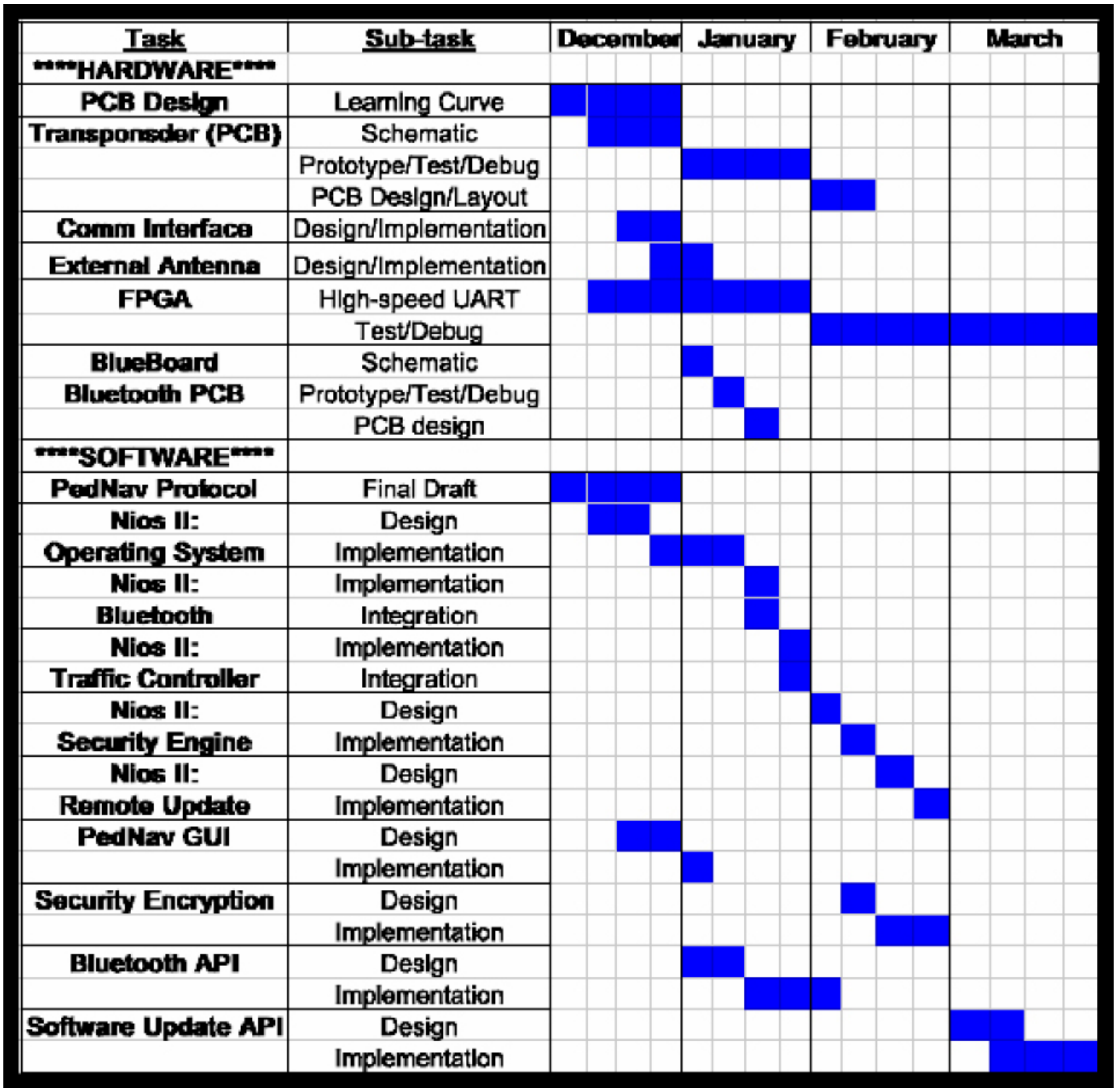


Figure 2. Project Gantt chart.

The Gantt chart in Figure 2 shows the project status and our long-term development tasks and goals. Alex and Thai will share the task of hardware design in terms of PCB layout and FPGA design. Amir will handle mobile phone software development, and Shawn will lead the FPGA programming effort.

We are considering many possible features for which this application can be used in additional to the minimum system goal proposed. One idea is to install the system at bus stops to relay information about schedules and estimated arrival times for particular buses at bus stops. Ideally, anyone with a Bluetooth-capable mobile phone would be able to take advantage of our networked system service free of charge.

Another idea is that we could install transceivers at shopping malls to provide store directories. The system could use the current position of the user to relay information such as directions to nearby stores, restaurants, parking lots, etc.

The possibilities of a generic pedestrian navigation system are only limited by one's creativity!

References

(1) http://www.bluetooth.com/Bluetooth/Learn/Works/Profiles_Overview.htm

(2) http://www.iowasudas.org/documents/13H-1-05.pdf