Voice Operated Guidance Systems for Vision Impaired People: Investigating a User-Centered Open Source Model

Bruce Moulton[*1], Gauri Pradhan[2], Zenon Chaczko[3]

[*1] *Corresponding author* *School of Electrical, Mechanical and Mechatronic Systems, Faculty of Engineering and IT, University of Technology, Sydney, Australia*
[*2, 3] *School of Computing and Communication, Faculty of Engineering and IT, University of Technology, Sydney, Australia*
brucem@eng.uts.edu.au, zenon.chaczko@uts.edu.au

Abstract

People who have impaired vision regularly use white canes and/or guide dogs to assist in obstacle avoidance. Guide dogs can also be of limited assistance for finding the way to a remote location, known as "wayfinding". Several electronic devices are currently available for providing guidance to a remote location, but these tend to be expensive, or make use of a Braille interface. This project investigated the suitability of a user centered client server approach for the development of a talking GPS system intended to fill a niche for outdoor wayfinding. The work resulted in a working prototype proof-of-concept system that uses a speech-recognition speech-synthesis interface. The prototype solution includes a custom web application which accesses the Google maps API. The system is intended to be scalable and extensible with additional features such as sensors for obstacle avoidance and access to web-based information such as weather, train or bus timetable information. The client server approach was found to be suitable for the development of this type of application.

Keywords

GPS, client server, blind, impaired vision, speech recognition, wayfinding, guidance, navigation

1. Introduction

The World Health Organisation estimates there are about 314 million vision impaired people in the world, of which about 45 million are blind [1]. The leading causes of blindness are cataract, uncorrected refractive errors, glaucoma, and macular degeneration. Many people who are seriously vision impaired use a white cane and/or a guide dog to avoid obstacles. Recent years have seen the development of several technologies designed to assist vision impaired people,

both for *obstacle avoidance*, and to travel to specific locations, or *wayfinding*.

The research described here focused on the development and evaluation of a prototype wayfinding system that makes use of GPS (the Global Positioning System) and speech recognition. The decreasing cost of GPS units, coupled with the recent growth in the availability of voice recognition services through the internet, presents an opportunity to create a low cost solution. A key priority of the prototype was to meet the user's navigational needs while ensuring low cost and portability. The system architecture includes a wireless interface to enable the addition of sensors and other subsystems in the future. Speech recognition technology was used as the primary interface.

The project implemented a functional proof of concept prototype which enables users to obtain spoken GPS navigational information using an intuitive speech recognition interface. The prototype system can advise the user where he/she is currently located, and provide spoken directions to travel to a remote destination.

Freely available software components were used wherever possible. It is important to note that the final product is not intended to replace the white cane— canes and guide dogs would remain essential for detection of obstacles.

2. Background

Navigation systems for visually impaired people perform two main functions. The first is to sense obstacles or dangerous terrain in the immediate environment (obstacle avoidance), and the second is wayfinding, or navigating to remote locations [2, 3]. Older-generation electronic obstacle avoiders, such as the *Laser Cane* and ultrasonic-based *NavBelt*, inform the user of nearby barriers, and in some cases can be used to find paths that circumvent such obstacles [4, 5]. Despite technological advances of recent years, the

blind community has not favored such devices over white canes. White canes are very reliable and effective at detecting any kind of obstacle obstructing the person's immediate pathway such as "uneven surfaces, holes, steps, and puddles" [6]. White canes are also relatively inexpensive and easy to use. For these reasons, recent years have seen research focused more on wayfinding rather than obstacle detection.

GPS wayfinding systems are primarily suitable for outdoor environments because the receivers are commonly unable to perform well in an indoor environment. Methods for relative positioning indoors include sensors using sonar, digital tags and accelerometers [7, 8, 9, 10].

Currently available outdoor wayfinding GPS systems for the visually impaired include *Tormes*, *Trekker, BrailleNote GPS* and *Loadstone GPS*. The Tormes system was developed by the European Space Agency (ESA) in 2004, and is the most accurate system available. It uses a system called EGNOS (European Geostationary Navigation Overlay System) which overlays signals from geostationary satellites to verify that GPS data is correct. EGNOS essentially improves the accuracy of GPS from 10m to 2m [11]. Tormes combats the "canyon effect" (tall buildings blocking satellite) by using a system called SISNet (Signal in Space through Internet), which provides corrective data through the internet when satellites are obscured. The Tormes system is, however, relatively expensive—it retails for over US$3000, and it is currently only available in Spanish. The *Trekker* system includes talking menus, talking maps and GPS information—it costs US$895 [12]. *BrailleNote GPS* is a satellite navigation add-on for owners of the *BrailleNote* family of products—it retails for about US$1500. *Loadstone GPS* is open source software, and enables users to interact with GPS information using synthetic speech and can be used on Symbian S60. While it is true that several systems have been validated from a technical point of view, few are widely used amongst the vision impaired community [13].

Most of the current systems make use of GPS combined with detailed digital street maps to display information about the locations of nearby streets, points of interest and instructions for traveling to desired destinations.

For a person to follow a particular route, the person must ordinarily have some concept or plan of that route. A blind traveler can learn the route either by being guided by a sighted escort, or by verbal instructions [14]. Once a route has been learned, successful travel requires that the individual be able to: (1) detect and avoid obstacles and (2) follow the route (i.e., to know their position and orientation and make necessary corrections). It is proposed that the performance of both tasks can be supported by electronic travel aids [14].

Two key abilities are required to successfully follow a route [2]. These are "the ability to recognize key choice points or decision points in the environment", and the ability to "interpret, and understand the layout of features in the environment". An implication is that the user must be made aware of the intersection of roads and other Points of Interest (POIs) such as train stations, parks, and shops which are commonly used by sighted people to navigate.

Some of the current GPS systems make use of Braille keyboards for user input and/or system output. However, not all vision impaired people can read Braille. A survey conducted by the Royal National Institute for the Blind found that only 3 per cent of visually impaired people are sufficiently fluent in Braille to read a book or magazine [15]. This is because the majority of blind people lose their sight later in life, either progressively or through an accident and therefore do not learn to read Braille as a child [16]. Another reason for the decline in the number of people using Braille is because it has been superseded by assistive technologies such as audio books, screen readers and speech recognition software which are easier to use.

To ensure that a navigation system will be accessible to the greatest proportion of vision impaired people, usability is a key focus of the project, and speech recognition technology was identified as a priority feature of the system. Further, by replacing the Braille keyboard with a speech recognition user interface, the device will be more portable and less cumbersome to use while walking.

Speech recognition technology has been under development for more than three decades, and demand for this technology is expected to rise dramatically in the near future as people access the internet using mobile phones [17].

Recent advances in speech recognition technology, coupled with the advent of modern operating systems and high powered affordable personal computers, have culminated in the first speech recognition systems that can be deployed on portable devices. According to Morgan [18], speech processing technology "provides a natural and intuitive interface for the user". Market research conducted in 2005 found that those who choose to use a speech recognition system do so because they "find speech is easy to use and understand and it provides a quick transaction process" [19].

Adaptive technologies such as computer screen reading software, talking mobile phones and portable Braille notetaking devices, are critical for people who

are blind or vision impaired both in schooling and employment. However much of this equipment is expensive, with a typical screen reading software licence costing around $1,000 [20].

As a result, the need to provide adaptive equipment at a lower cost will be critical to ensuring maximum success for any newly developed aid for visual impaired people.

3. Methodology

The system was designed using object orientated methods. Requirements were determined, and software objects and their interactions were defined to fulfill those requirements. A static view of the software objects was formed using a class diagram which illustrates the methods and attributes of each component. Dynamic views of the software objects' interactions were created using sequence diagrams.

The development of the raw user requirements included the following considerations: (a) the system must be inexpensive, portable and easy to use; (b) the system must convey to a blind user wayfinding information including current location and the direction/location of a remote destination; (c) it is desirable that the system can be later expanded to convey obstacle avoidance information relating including positions of stationary/moving objects and the terrain surface.

The requirements analysis stage investigated the tools and techniques that blind people employ to navigate around urban areas. It also evaluated other devices currently available on the market, investigated the extent to which currently available hardware and software options could be used to create a workable solution.
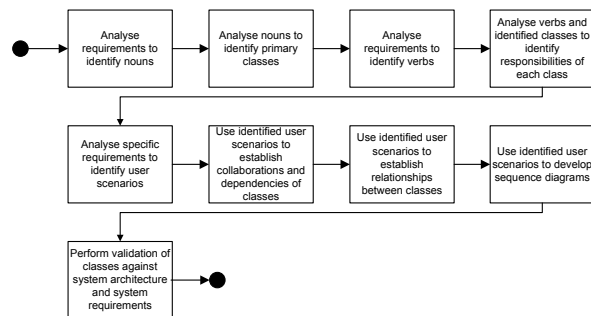

Figure 1. The method used to develop the class diagram and sequence diagram

The development process is outlined in Figure 1. Noun and verb parsing on the system requirements was conducted to determine primary classes. Upon the identification of classes and their responsibilities, user

scenarios were then identified and used to establish collaborations and dependencies of these classes. These scenarios were also used to develop sequence diagrams. Finally the identified classes were validated against the system architecture and system requirements. As a result of this process, it was determined that the project would attempt to develop a functional prototype using GPS and voice recognition technology.

A first iteration of the build plan delivered the features which had been identified to be the core functionality of the system, including a voice recognition interface which would provide the current location and directions to another location. A second iteration extended the functionality of the system and provided SMS help and a Bluetooth interface to other systems.

The methodology employed the following tools and technologies: Franson GPS Gate v2.6.0.340, Microsoft Internet Explorer Speech Add-in (part of SDK v1.0), Internet Explorer, BlueSoleil v2.6.0.8, MyNetFone's MyText SMS API, Google Maps API. BlueSoleil was used to wirelessly access the Bluetooth enabled GPS receiver and headset.

The Franson GPS Gate was used to take the raw GPS data from a connected Bluetooth GPS device and filter it using the standard NMEA protocol to obtain the latitude and longitude values. The user interface implemented using Microsoft Speech Internet Explorer add-In in conjunction with Speech Application Language (SALT) tags. SALT is based on a small set of XML elements. The main top level output is the <prompt…> tag which is used to ask the user for input. These tags are used to specify the content of the input which is required. The top level input is the <listen…> tag which is used to start speech recognition or audio recording. It contains the grammar element used to specify the different things that the user is expected to say. The ASP.NET application is accessed by a computer running Internet Explorer (IE) with the speech add-in or by Pocket IE with the speech add-in.

The SMS service provider utilised in the implementation model was the MyNetFone's MyText SMS API. It allows HTTPS requests to be called from any web browser or application [21]. The key Google Maps API classes which were used were GDirections and GReverseGeocoder. GDirections was used to obtain walking direction results from the current location to a given destination address. The directions which are obtained are read out to the user. In order for this class to operate it requires a map and/or a text panel, therefore the results can be displayed visually. GReverseGeocoder was used to get the current location as an address. To obtain a reverse geocode (address) for user's current location, it requires a GPS geocode

'point' which is made up of a latitude and longitude. It does not use any caching mechanism and is limited by Google to 10,000 requests per IP and per day [22].

Hardware used in the system includes headphones, a Bluetooth GPS receiver and a PDA. Ideally, users want headphones that do not fully cover their ears. This is because it is safer if a user is able to listen to ambient noises and focus on elements of the environment other than solely the system's output [23, 24, 26]. However for the prototype the headset used was the type that covers the ears. The Bluetooth receiver used in the final solution was the SiRF Star III 20ch Bluetooth GPS receiver. It outputs the standard NMEA protocol with GGA, GSA, GSV, RMC and VTG. The specification for the position accuracy of this device is 10 meters.

Two candidate architectures were analyzed, and one chosen based on suitability to the unique problem space, ability to meet the requirements and inherent implementation risks. The most important focus was ensuring that it meets the purpose of the system as detailed in the SRS. The feasibility of implementing the system was also taken into consideration when developing the candidate architectures. A risk analysis that considered schedule and implementation risk was undertaken to ensure the final architecture was of the least risk to the project and thus more feasible to develop.

Initial analyses of the problem considered quite a few alternate system architectures from which two final candidate architectures were chosen—these were Broadcast and Client Server. The broadcast model is an event-driven system which responds to user interaction and external sensory data when it is made available. The Client Server architecture on the other hand, divides the system into two tiers. Each of the candidate architectures was described and analyzed using the following two views: (1) a logical view, which described the domain-level responsibilities of the system, and (2) a physical view, which described the interconnecting physical components of the system.

The broadcast model provides concurrent communication of data and messages to all system components. This allows for built-in redundancy and robustness if any hardware component fails. This model also minimizes the coupling in the overall system since it leaves component to component communication to the Event and Message Handler to relay the messages. A logical view of the broadcast model is given in Figure 2.
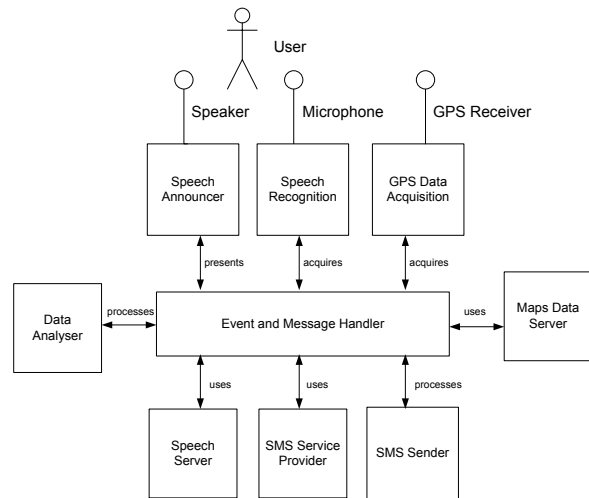

Figure 2. Broadcast model, logical view

Client server models, on the other hand, distinguish between local client systems, which typically reside on the user's hardware, and server systems, which ordinarily reside in some distant location over a network. The basic characteristics of a client is that it initiates requests, waits for replies, is usually connected to only a small number of servers at any one time, and users typically interact with a client using a graphical user interface. Characteristics of a server are that it does not initiates requests or activities, it waits for and replies to requests from connected clients [27]. Viewing the speech-GPS system in this way, client server characteristics are readily apparent. Thus a client server model was selected with the following components:

Speech Recognition (Client): allows the user to interact with the system; translates the provided user input (from the microphone) and presents the data to the system for processing

Speech Announcer (Client): presents the system response to the user; translates the provided system input into speech and outputs this to the user through the speaker.

GPS Data Acquisition (Client): provides the sensory information to the other system components by retrieving the data from the GPS sensors; primary responsibility of this component is to gather the sensory information, process the data and present it in a meaningful format to the other system components.

Data Analyzer (Client): responsible for error checking data, controlling the system and requesting data from other system components.

Maps Data (Server): provides an interface to obtain web services data from the Google Maps API.

The physical model included the following characteristics (1) the primary device (the PDA) is physically connected to the headphones and

microphone; (2) the PDA is connected to the internet via Wi-Fi; (3) the PDA connect to the Internet to obtain map services and send SMS messages (4) a GPS receiver provides information to the PDA via Bluetooth; (5) the system provides services to the presentation components of the system, for example, announcing real-time diagnostic data.

The diagram below illustrates the way in which class validation against the architecture was conducted. Each class that was identified was mapped back to the components defined in the system architecture.
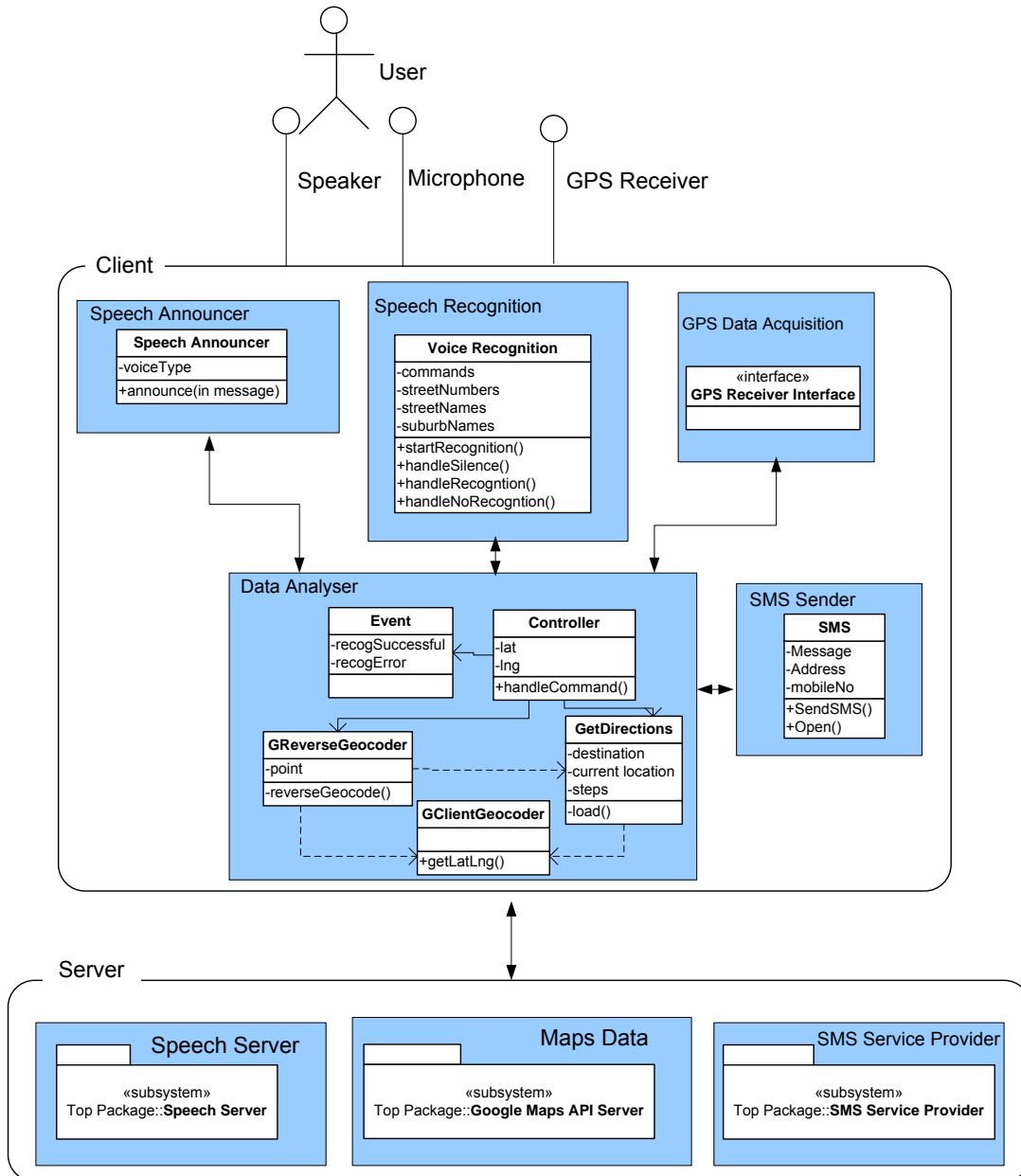

Figure 3. How the design was mapped to the Client Server model

The prototype solution was implemented as a web application. Reasons for this included (a) the Google API exclusively supports web applications only; (b) Secondly, Microsoft's Speech Internet Explorer add-in provided a way to accomplish the Speech Recognition feature conveniently for web applications, and circumvented the need to design complex grammar structures (which are required for standalone applications) as this is handled by the Speech Server instead; (c) an evaluation of the possibility of implementing the features in a stand alone application proved to be infeasible. For example, the cost of obtaining a maps database SDK was quoted at $20,000 per license.

Substantial consideration was given to the possibility of adapting currently available off-the-shelf GPS navigation systems that are commonly used be drivers of vehicles. These systems are especially attractive because some of them provide spoken directions. However, it was determined that it certain aspects of these systems prevent them from being used by vision impaired people. One of the barriers to the adaptation of these systems is that they ordinarily rely on a touch screen graphical user interfaces for many functions, such as setting a destination., and a touch screen is not suitable for a person who has impaired vision.

4. Preliminary evaluation

The system operates under an assumption that only "expected" words need to be recognized. Other words are not be recognized as valid input. It is therefore important for the users to know the exact words they can say for the system to recognize their input. The main approach which is used to achieve this is to prompt the user for input.

On start-up, the entire list of main menu commands are given to the user as the following prompt is played:

SYSTEM:
Welcome to UC's main menu. To get your Current location say 'Current Location', to get directions say 'Get Directions,' to change Obstacle Detection say 'Obstacle Detection, to send an SMS, say 'Help'. To stop UC talking say 'stop' or 'mute' at anytime. To exit say 'exit'. To hear these options again say 'Main Menu' or 'Repeat'.

This is a necessarily long list, which takes a significant amount of time for the system to announce fully, as first time users would need to hear all of the necessary information. However, experienced users who already know the commands are able to interrupt

the system at any time and skip to the desired function. An example of a user interrupting the system is given as follows:

SYSTEM:
Welcome to UC's main me-

USER:
Current Location

SYSTEM:
You selected Current Location. You are currently located at 5 Aiken road, West Pennant Hills, New South Wales, 2125 Australia

In this way, the system can flexibly cater to the needs of both first time users who require detailed instructions and experienced users who prefer fast performance. The following script provides an example of the system's output for the Get Directions function.

USER:
Get Directions

SYSTEM:
You selected Get Directions. Please say the destination street number or train station.

USER:
Beecroft Station

SYSTEM:
You selected Beecroft Station. Directions to Beecroft Station are:
- Head east on Aiken Rd toward Salina Ave
- Turn left at Cumberland Hwy
- Turn right at Hannah St
- Turn right at Wongala Cres

Table 1 provides all of the standard user commands.

Table 1. The user commands

Command	Description
Main Menu	Gives the following list of the main menu commands: To get your Current location say 'Current Location', to get directions say 'Get Directions,' to change Obstacle Detection say 'Obstacle Detection, to send an SMS, say 'Help'. To stop UC talking say

	`'stop' or 'mute' at anytime. To exit say 'exit'. To hear these options again say 'Main Menu' or 'Repeat'.`
Current Location	Gives the current address: `You are currently located at e.g. Aiken road, West Pennant Hills, New South Wales, 2125 Australia`
Get Directions	Prompts the user for a destination address or station: `Please say the destination street number or train station`
Help	Prompts the user to say "Send" to send an SMS: `To send an SMS for help with your current location say 'Send'`
Repeat	Repeats the last system output again
Send	Sends an SMS with the current address: `SMS has been Sent`
Stop	Stops the current system output and listens to the user
Exit	Exits the program

The command words are standardized, but it is expected that the input from each user will vary according to their accent, pronunciation, and to a small extent, their word preference. For this reason, the system was designed to recognize common alternative commands which differ slightly choice of words. For example, the user may say "Location" instead of "Current location".

For the "Get Directions" feature, the system is capable of recognizing either an address or a point of interest such as the name of a train station. For the purposes of the proof of concept solution, street names in Sydney CBD and West Pennant Hills were loaded into the speech recognition system.

Though the prototype list of commands is relatively short, it could be relatively easily extended to include a vast number of street names and station names without affecting the speed performance of the system. However, it is recognized that increasing the system's vocabulary increases the likelihood of the system misinterpreting a command.

The current version of the prototype requires further work before it would be potentially suitable for a blind person. To give more meaningful directions for a blind user, we must know their orientation and tell them the directions *relative* to their orientation. It would also be much more useful if the system tells the user when they are walking in the wrong direction. The bearing necessary for this could be taken from a digital compass. In addition, it would useful to give approximate distances, for example, "turn right at Hannah St after 50 meters".

Ideally the system should be able to recognize the user's voice and disregard background noise from the street. Implementing a mute function where the system stops listening for input would be an advantage. Alternatively, a "push to talk" configuration would help in limiting the impact of background noises.

It would be desirable to providing a configuration file so that the user can configure User name, Mobile number (that the SMS gets sent to) and more street names and suburbs. This would allow the user to personalize the system so that their home suburb and places of interest will be recognized by the speech recognition system. It would mean that the system could send a more meaningful SMS for help to emergency services by including the person's name.

It would be possible to add a function to enable the user to request train timetable information. Assuming the user is traveling to a train station, the user might say "timetable, Northern Line", to which the system might respond with something such as "The next trains are at 8:17am and 8:24 am".

The proof of concept system demonstrates that it is possible to obtain directions to points of interest (POIs) as it allows users to request directions to train stations. The system could be extended to include other POIs such as parks, hospitals, bus-stops, restaurants, and shops. This extension would only be limited by the Google Maps API being capable of servicing these requests.

Ideally, as the visually impaired person moves through the environment, he/she would hear the names of buildings, street intersections, etc. spoken as though coming from the appropriate locations in auditory space as if they were emanating from loudspeakers at those locations. Besides leading the visually impaired person along a desired route, the system would hopefully allow the person to develop a much better representation of the environment than has been the case so far.

5. Conclusion

Currently available electronic navigation aids have not received any great level of support among the blind community. The vast majority of visually impaired people prefer to not use electronic aids, and use only canes or guide dogs. It is proposed that the underlying reasons for this include the relatively high costs and relatively poor levels of user satisfaction associated with existing electronic systems.

The work undertaken in this project investigated the suitability of a user-centric client server approach for the development of a new electronic wayfinding system intended primarily for people who have impaired vision. Speech recognition technology was identified as being a suitable basis for the interface of the system.

The project successfully delivered a functional prototype that uses intuitive hands-free voice recognition technology. The client server model enabled the system to make use of existing tools including the MS IE Speech Add-in and the Google Maps API. In addition, the client server approach enables much of the processing to be undertaken in remote locations, rather than on the user's hardware.

The prototype solution is intended to be suitable for further extensions including an electronic compass, announcing distances (in meters), using push-to-talk to reduce the affect of background noise, and adding additional configuration capabilities. In addition, the system may be augmented by the provision of additional services such as the ability to access train and bus timetable information. Further, a digital compass may be supplemented by other sensors for the purpose of obstacle avoidance. The client server approach draws on well understood principles, and was found to be suitable for the development of this type of application.

6. References

[1] http://www.who.int/mediacentre/factsheets/fs282/en/ [accessed 19.5.2009]

[2] Loomis, J. M., Golledge, R. G. & Klatzky, R. L. (2001). GPS-based navigation systems for the visually impaired. In: Barfield, W. & Caudell, T. eds Fundamentals of wearable computers and augmented reality. Mahwah, NJ: Lawrence Erlbaum Associates. pp. 429-446.

[3] Freedom Scientific, 2000, StreetTalk GPS Solution, http://www.freedomscientific.com/fs_products/StreetTalk.asp [accessed 20.10.2008].

[4] Brabyn, J. A. (1985). A review of mobility aids and means of assessment. In D. H. Wamn & E. R. Strelow (Eds.), Electronic Spatial Sensing for the Blind (pp. 13-27). Boston: Martinus Nijhoff.

[5] Shoval, S., Borenstein, J., Koren, Y. (1996). The NavBelt − A computerized Travel Aid for the Blind based on Mobile Robotics Technology, viewed 10th July 2008, http://www-personal.umich.edu/~johannb/Papers/Paper45/Paper45.html

[6] Borenstein, J. (1997) A Computerised Travel Aide for the Active Guidance of Blind Pedestrians, Proceedings of the IEEE International Conference on Robotics and Automation, Albuquerque, NM, Apr. 21-27, 1997, pp. 1283-1288.

[7] Golding, A.R., Lesh, N. (1999). Indoor navigation using a diverse set of cheap, wearable sensors. In: third international symposium on wearable computers. San Francisco, CA, pp 29–36.

[8] Long, S., Aust, D., Abowd, G.D., Atkeson, C. (1996). Cyberguide: prototyping context-aware mobile applications. In CHI '96 conference companion, pp 293–294

[9] Ertan, S., Lee, C., Willets, A., Tan, H., Pentland, A. (1998). A wearable haptic navigation guidance system. 2nd international symposium on wearable computer, Pittsburgh, PA, pp 164–165

[10] Bourbakis, N., Kavraki, D. (2005). A 2D Vibration Array for Sensing Dynamic Changes and 3D space for Blinds' Navigation, http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01544470 [accessed 26.10.2008]

[11] European Space Agency (ESA) (2003). More autonomy for blind people thanks to satellite navigation, http://www.esa.int/esaNA/SEMVQOS1VED_index_0.html [accessed 20.11.2008]

[12] http://www.humanware.com/en-australia/products/blindness/talking_gps/trekker/_details/id_88/trekker.html [accessed 19.5.2009].

[13] Lumsden, J. (2008). User Interface Design and Evaluation for Mobile technology, pg 694-695, IGI Global.

[14] Shoval, S., Borenstein, J., Koren, Y. (1996). The NavBelt − A computerized Travel Aid for the Blind based on Mobile Robotics Technology, viewed 10th July 2008, http://www-personal.umich.edu/~johannb/Papers/Paper45/Paper45.html

[15] Bruce, I., McKennell, A. and Walker, E. (1991). Blind and partially sighted adults in Britain: the RNIB survey. London: HMSO.

[16] "The Open University" (2008). Assistive technologies for the Blind and partially sighted, http://openlearn.open.ac.uk/mod/resource/view.php?id=187372 [accessed 24.10.2008].

[17] Hamm S. (2008). IBM's Speech Recognition, Business Week, http://www.businessweek.com/innovate/content/aug2008/id20080818_745252.htm [accessed 20.11.2008]

[18] Morgan, S.(2005). Creating .Net Applications That Talk, InformIT, viewed 24th October 2008, http://www.informit.com/articles/article.aspx?p=378965

[19] ETNwhat? (2005). New Research Reveals Future Speech Technology Trends, viewed 20th November, http://www.fastthinking.com.au/site/page.cfm?u=17&c=1098

[20] Australia 2020 Summit Final Report (2008). Submission 7102: Blind Citizens Australia, http://www.australia2020.gov.au/submissions/viewTopic.cfm?id=7102&count=1 [accessed 24.10.2008].

[21] MyNetFone (2005). MyText SMS API, https://www.mynetfone.com.au/residential-voip-solutions/features/mytext-sms/mytext-sms-api [accessed 28.09.2008]

[22] Goeminne, N. (2007). Google Maps API Reference, Class GReverseGeocoder v 1.0.5, http://www.zeali.net/zpages/Google_Maps_API_Reverse_Geocoder/reference.html [accessed 29.9.2008].

[23] Sendero Group (2008). Sendero Group – Accessible Location and Navigation, http://www.senderogroup.com/shopgps.htm [accessed 29.10.2008].

[24] Moulton, B. (2009) Enabling safer design via an improved understanding of knowledge-related hazards; a role for cross disciplinarity Australasian Journal of Engineering Education (in press, accepted 22.12.2008).

[25] Moulton, B. and Y. Forrest (2005) Accidents will happen: safety-critical knowledge and automated control systems. New Technology, Work and Employment, Vol. 20, No. 2, 102-114.

[26] Loyd, L. (2008). What are the characteristics of client/server architecture, viewed 30th October 2008, http://www.faqs.org/faqs/client-server-faq/section-14.html [accessed 29.10.2008]