bachelor's thesis

Accessible UIP client for Windows Phone 8

Vojtech Novak

January 10, 2014

Macik Miroslav Ing.

Czech Technical University in Prague
Faculty of Electrical Engineering,
Dept. of Computer Graphics and Interaction

Acknowledgement

Acknowledgements

Declaration

I declare that I worked out the presented thesis independently and I quoted all used sources of information in accord with Methodical instructions about ethical principles for writing an academic thesis.

Abstract

This work describes the development of an accessible UIP client application for Windows Phone 8.

Klíčová slova

Navigace, Generování UI, Přístupnost, Windows Phone 8, UIP

Abstract

This work describes the development of an accessible UIP client application for Windows Phone 8.

Keywords

Navigation, UI Generation, Accessibility, Windows Phone 8, UIP

Contents

Abbreviations

The list of abbreviations used in this document

FEE CTU	Faculty of Electrical Engineering of the Czech Technical University in Prague
UIP	UI Protocol developed for research purposes at the FEE CTU
TTS	Text-to-Speech
OS	Operating System
API	Application Programming Interface

1 UIProtocol

This chapter introduces the reader to the UIProtocol, its architecture and discusses the UIProtocol client design.

1.1 About UIProtocol

Universal Interface Protocol (UIProtocol) is a user interface specification language [1] being developed at FEE CTU (currently 8th version of the specification draft) for research purposes. At the time of writing this thesis, the specification is not publicly available. UIProtocol provides means for describing user interfaces and transferring data related to interaction between user and an UIProtocol based application. It is designed to be cross-platform, programming language independent and easily localized. UIProtocol is an XML based application protocol that allows for describing the hierarchical structure of the GUI along with the placement and visual appearance of the containers and components. It is designed for a client-server system and for facilitating client-server applications it defines the communication rules between client and server. The communication is based on exchange of XML documents which contain all components and values needed for rendering the UI. The client first initiates the communication and receives respective XML description from the server. The description can be of four different types: interfaces, i.e. the UI components and containers, models which contain the data displayed and actions. The communication from client to server only consists of event descriptions. For example, whena user presses a button, the event information is sent to the server which responds by model and/or interface update.

The documents of UIProtocol can be sent in either direction usually through a single channel without waiting for a request, e.g. server can send updates to the client as soon as the displayed information needs to be updated, without having to wait for an update request. Should an application not communicate with a remote server, there is the possibility of both client and server running on the same machine although this is not a typical usage.

1.1.1 UIProtocol Client

UIProtocol client is thin, i.e. no application code is executed on the client side. The device running the client is thought to be the the one user directly uses, that is, it renders the content to the user and receives input from them. From the UIProtocol point of view, the client device is also considered insecure, i.e. the device may be misused to send invalid data to the server and may be used to attack it.

1.1.2 UIProtocol Server

UIProtocol is the part of the architecture which is responsible for evaluating the client events and sending a correct response - this is where the application logic is executed.

It must be able to service multiple client simultaneously and is intended to run on a machine which is considered safe.

1.1.3 Elements of UIProtocol Communication

As mentioned previously, the information exchange between client and server concerns Interfaces, Models, Actions (which are sent from server to client) and events (sent from client to server). In the following subsections we will describe these in a greater detail.

Interfaces

Interface describes the structure and components of the user interface. Every interface can nest containers and elements that form a part of user interface.

Listing 1.1 Some Code

1.1.4 UIProtocol Client Design

The UIP client may not implement the whole feature set defined by UIProtocol. What has to be implemented is the minimal functionality that is able to render a user interface, sending event information to the server and update the application it by data coming from it. The UIP client will be developed for the Windows Phone 8 operating system and written in `C#`. The application will be developed using Visual Studio 2013.

2 Accessibility of Current Mobile Platforms

In this chapter we will analyze the accessibility features of today's most common mobile platforms. Since this thesis is about development of a UIP client for Windows Phone 8, we will put emphasis on this OS.

2.1 Windows Phone 8 Accessibility

This section covers all of the features for ease of access that are included in the Windows Phone 8 operating system. For the purposes of this project, we are particularly interested in features that may help the visually impaired users. From this point of view, perhaps the most important are the voice commands and speech recognition features which Windows phone 8 has built-in and which support a wide range of languages.

2.1.1 Windows Phone 8 Speech Features

Users can interact with the phone using speech. There are three speech components that a developer can integrate in her app and the user can take advantage of them: voice commands, speech recognition, and text-to-speech (TTS). We will explore these features in the following paragraphs. At the time of writing, the speech features support 15 major languages ranging from English to Russian or even English with the Indian accent. Czech, however, is not supported. To use the speech features the user has to download a language pack.

Speech Recognition

Users can give input to an app or accomplish tasks with it using speech recognition. An example usage can be dictating content of an SMS. This is very similar to the Voice Command feature, but the key difference is that speech recognition occurs when user is in the app, and Voice Commands occur from outside of the app [2]. The second key difference is that the Voice Commands are defined on a finite and usually small set of words (commands), whereas the Speech Recognition should recognize words from a much larger dictionary – in ideal case a whole human language.

Voice Commands

When a user installs an app, they can automatically use voice to access it by speaking "open" or "start", followed by the app name. The range of actions that can be triggered by Voice Commands is much wider, the full list of available speech commands that are provided by the operating system is listed in table **. A developer can also define her own set of voice commands and allow users not only to open the app using voice but also to carry out more advanced tasks within the app. This is very important for our work since it allows for exposing a wider range of commands to the visually impaired user. Note that technically, this still happens from the outside of the app, as described in the previous section.

Text to Speech (TTS)

TTS can be used to speak text to the user via the phone's speaker or headset.The spoken text can be simple strings or strings formatted according to the industry-standard Speech Synthesis Markup Language (SSML) Version 1.0. TTS is also used in some of the other features for ease of access which are covered in the next section.

Other Speech Features

A feature named Speech for phone accessibility enables the following: 1) Talking caller ID When getting a call or receiving a text, the phone can announce the name of the caller or the number. 2) Speech-controlled speed dial User can assign a number to a person from the contact list and then say Say "Call speed dial number" (where number is the assigned number) to call the person. Assigning the speed dial number is also speech-enabled. 3) Read aloud incoming text messages Similarly to 1, the phone can read the content of a text.

2.1.2 Other Tools for Ease of Access

Windows phone 8 comes with more features for ease of access which can help lightly visually impaired users. User can change font size in apps (not in the tiles of the home screen), switch the display theme to high-contrast colors and use the screen magnifier. Mobile Accessibility is a set of accessible apps with a screen reader, which helps use the phone by reading the application content aloud. The applications include phone, text, email, and web browsing. When Mobile Accessibility is turned on, notifications like alarms, calendar events, and low battery warnings will be read aloud.

2.2 Android Accessibility

Similarly to the previous section, here we will analyze the accessibility options for devices running the Android operating system, with the emphasis on visually impaired users. We will analyze the features of the latest Android OS released, which is version 4.4, code name KitKat. It should be noted that there were no major updates to the accessibility options since Android 4.2.2 Jelly Bean.

2.2.1 Speech Features

Android also offers the option to interact with the device using speech and has some interesting accessibility features and compared to Windows Phone 8 offers a wider language support. Similarly to Windows Phone, an Android developer can take advantage of speech recognition and text-to-speech (TTS). Android comes with a number of built-in voice commands but unlike the Windows Phone, Android does not allow developers to expose their own voice commands. The last important feature on Android is TalkBack. At the time of writing, the speech recognition supports more than 40 languages including several accents of English, and even minor languages such as Czech. Other functions do not have such a wide support.

Speech Recognition

Users can give input to an app or accomplish tasks with it using speech recognition. An example usage can be dictating content of an SMS. As mentioned before, this feature supports many languages but on the other hand required internet connection. There is not an option to use the recognition offline. We do not consider this a major drawback, as nowadays a mobile internet connection is more available than ever.

Voice Action Commands

In Android, Voice Action Commands are closely related to the Google Now feature. Google Now has a wide range of uses not specifically designed for visually impaired. It can, however, serve them well by allowing them to get information using voice. In general, google now should provide the user with relevant information when they need it. Google describes it by the phrase “The right information at just the right time”. This includes telling the user the weather forecast, showing the best route to work, calling someone, creating a reminder and much more.

Note that for some commands, the system gives you a spoken answer. The current drawback of the system is that it only supports English, French, German, Spanish, and Italian. With other languages, user can only make a voice-induced Google search with no voice response. Text to Speech (TTS) TTS can be used to speak text to the user via the phone’s speaker or headset. The spoken text can be simple strings. The industry-standard Speech Synthesis Markup Language (SSML) is supported only in a limited scope. TTS is also used in TalkBack which is described in the next section.

Other Speech Features

TalkBack is an important functionality that strives for more accessible phone control for visually impaired. Basically, it is a touch-controlled screen reader. When enabled, user can drag finger across the screen selecting the components and getting their acoustic description. By double tapping anywhere in the screen, user can open/use the last selected item. TalkBack also supports gestures. This way, a user can get a complete description of the user interface. The blog post of a blind accessibility engineer from Mozilla Foundation [3] claims that visually impaired users of this system still have to overcome some obstacles.

2.2.2 Other Tools for Ease of Access

Android too comes with more features for ease of access which can help lightly visually impaired users which include change font size and the screen magnifier.

2.3 iOS Accessibility

In this chapter, we will cover the accessibility of Apple's iOS. Again we consider the latest iOS released, which is version 7.0.4. Overall, the accessibility features of iOS are very similar to those of Android and therefore I will describe the features more briefly.

2.3.1 Speech Features

As with the previous two platforms, iOS also offers users o interact with a device using speech. Apple was the first one to introduce the features for people with disabilities, such as VoiceOver. iOS supports speech recognition and text-to-speech in 15 major languages (the same number as Windows Phone 8). iOs also comes with a number of built-in voice commands but does not allow developers to expose their own voice commands.

Speech Recognition - Dictation

Users can give input to an app or accomplish tasks with it using speech recognition. An example usage can be dictating content of a text. As mentioned before, this feature supports 15 languages and requires an internet connection.

Voice Control - Siri

Siri in iOS can be thought of as an equivalent to Android's Google Now. Siri can send emails, set reminders and more. If asked a question, it can read aloud the answer.

Text to Speech (TTS)

TTS can be used to speak text to the user via the phone's speaker or headset and this feature was added only recently, in iOS 7.0. The spoken text can be simple strings. The industry-standard Speech Synthesis Markup Language (SSML) is not mentioned in the API documentation.

Other Speech Features

It could be said that Google's TalkBack is Apple's VoiceOver. Both offer very similar functions and their key reason for existence is reading the content of the screen based on touch input and control of the device by gestures. The mentioned blog post of the blind accessibility engineer from Mozilla Foundation favors VoiceOver over TalkBack [3].

2.3.2 Other Tools for Ease of Access

iOS too comes with more features for ease of access which can help lightly visually impaired users. The user can change font size, invert Colors and use the screen magnifier (Zoom). iOS devices also support more than 40 Bluetooth wireless braille displays out of the box. User can pair their braille display with the device and start using it to navigate it with VoiceOver. In addition, iPad, iPhone, and iPod touch include braille tables for more than 25 languages.

3 Navigation Systems Analysis

There is a number of interesting papers in the field of navigation systems for visually impaired. Generally speaking, there are ongoing efforts to create maps for indoor environments, with the Google Indoor Maps being the head of this movement. Currently, the Google Indoor Maps are in beta and are not a priori intended for navigation but merely to provide the user with an approximate idea of where they are. In this chapter we will analyze some of the existing works which specifically address the problem of indoor navigation and focus on projects that specialize in navigation of visually impaired. There are two main approaches to the problem. In the first, the navigation system consists of active parts which, using triangulation or other methods, are able to determine the user's position at all times and then give her directions based on knowing where she is. In the second approach, the system does not possess the information about user's position at all times. Instead it synchronizes the position at the beginning of the navigation task and then gives the user directions broken into small chunks. When the user believes she reached the destination described by the first chunk, she asks for the next one and etc. The disadvantage of this approach is that the user can get lost and not end up at the expected location. This problem can be solved by adding more "synchronization points" to strategic locations of the building.

3.1 Current Systems for Visually Impaired

NaviTerier [4] is a research project at FEE CTU which aims directly at the problem of navigating visually impaired inside buildings. This system does not require any specialized technical equipment. It relies only on mobile phones with voice output, which visually impaired people already use. The navigation system works on a principle of sequential presentation of carefully prepared description of the building to visually impaired user by the current mobile phone voice output. This system does not keep track of the user location. Instead, it breaks the directions into smaller pieces and then sequentially gives the pieces to the user who follows them and asks for next portion when ready. This system was tested with 13 visually impaired users.
Recently this system was combined with UI Protocol platform, which is another research project of FEE CTU developed for the purpose of creating user interfaces customized to abilities and preferences of individual users. The result is navigational system called NaviTerier UIP (NUIP) [5] which combines the navigational part of NaviTerier and the ability of UIP to generate and deliver customized user interfaces that can fit better people with disabilities.

Luis et al.[6] propose a system which uses a infrared transmitter attached to the white cane combined with Wiimote units (the device of the Wii game console) placed so that they can determine the user's cane position using triangulation. The information from Wiimote units is communicated via Bluetooth to a computer which computes the position and then sends the directions to the user's smartphone via wifi. TTS engine running on the phone converts the directions to speech. The system has undergone preliminary testing with five blindfolded users.

An indoor navigation system to support the visually impaired is presented in [7]. The paper describes creation of a system that utilizes a commercial Ultra-Wideband (UWB) asset tracking system to support real-time location and navigation information. The paper claims that the advantage of using UWB is its resistance to narrowband interference and its robustness in complex indoor multipath environments. The system finds user position using triangulation and consists of four parts: tracking tag to be worn by the user, sensors that sense the position of the tracking tag, handheld navigator and a server which calculates the location of the tracking tag and communicates it to the navigator. The handheld device runs software which can produce audio directions to the user. In tests, the system proved useful; It was, however, tested only on blindfolded people.

Treuillet and Royer [8] proposed a vision-based localization for blind pedestrian navigation assistance in both outdoors and indoors. The solution uses a real-time algorithm to match particular references extracted from pictures taken by a body-mounted camera which periodically takes pictures of the surroundings. The extraction uses 3D landmarks which the system first has to learn by going through a path along which the user later want to navigate. It follows that the system is not suitable in environments that are visited for the first time. For the case when it has learned the way, the system performs well.

A promising approach is shown in the PERCEPT [9] project. Its architecture consists of the three system components: the Environment, the PERCEPT glove and Android client, and the PERCEPT server. In the environment there are passive (i.e. no power supply needed) RFID tags (R-tags) deployed at strategic locations in a defined height and accompanied with signage of high contrast letters and embossed Braille. The next part of the environment are the Kiosks. Kiosks are where the user tells the system her destination. They are located at key locations of the building, such as elevators, entrances and exits and more. The R-tags are present here and the user has to find the one she needs and scan it using the glove. The glove is used to scan the R-tags and also has buttons on it that the user can press to get the instructions for the part of the route, repeat previous instructions and get instructions back to the kiosk. Also, after scanning the R-tag the gloves sends its information to the app running on user's Android phone. The app connects to the internet and downloads the directions from the PERCEPT server. These are then presented to the user through a text-to-speech engine and the user follows them. The system was tested with 24 visually impaired users. Another example of RFID use is presented in Lopez et al. [10] where user is navigated by following paths marked by RFID labels on the floor. The white cane acts as an RFID reader and communicates with a smartphone which, as in other projects, uses TTS to give directions.

There are also research works in the fields of robotics and artificial intelligence that study the problem of navigation. More specifically, they tackle the problem of real time indoor scenes recognition [11], [12], [13]. Some of these solutions allow for creating the reference map dynamically. Even though they proved to be useful in the domain of robotics and automotive industry, their applications to navigating people are limited, as they require expensive sensors and powerful computing resources. Wearing these devices would make the traveling of the users more difficult and limited. For these reasons, the solution proposed by Hesch and Roumeliotis [14] is interesting because they integrated these devices (apart from the computing) into a white cane. However, the solution has the limitations in being too heavy and large, and the laser scanner being directional.

4 Appendix

Table 1 Windows Phone 8 Speech Commands

To	Say this
Call someone from your contact list	"Call contact name" (where contact name is the name of someone in your contact list) If the person has only one phone number in your contact card, the ca
Call any phone number	"Call phone number" (where phone number is any phone number, whether it belongs to a c
Redial the last number	"Redial"
Send a text message	"Text contact name" (where contact name is the name of someone in your contact list) This will start a text message to that person. Then you can dictate a
Call your voicemail	"Call voicemail"
Open an application	"Open application" or "Start application" (where application is the name of any application on your phone, such
Search the web	"Find search term" or "Search for search term" (where search term is what you're looking for) If you say "Find local pizza," for example, Bing will bring up a map o

Table 2 Android Speech Commands

Say	Followed by
"Open"	App name
"What does my day look like tomorrow?"or "Show me my schedule for the weekend."	
"Create a calendar event"	"Event descrip
"Listen to TV"	Displays TV
"Map of"	Address, nam
"Directions to" or "Navigate to"	Address, nam
"Post to Google+"	What you wa
"What's this song?"	
"Remind me to"	What you wa
"Go to"	Search string
"Send email"	"To" & contac "Subject" & s "Message" &
"Note to self"	Message text
"Set alarm"	"Time" or "fo
"Listen to"	Play music in
"Call"	The name of

Bibliography

[1] V. et al. Slovacek. “UIProtocol specification, draft 8”. In: *CTU in Prague* (2010).

[2] *Speech for Windows Phone 8. Speech for Windows Phone 8 - development center.* 2014. URL: http://msdn.microsoft.com/en-us/library/windowsphone/develop/jj206958(v=vs.105).aspx (visited on 01/01/2014).

[3] *Android and iPhone accessibility comparison. Switching to Android full-time – an experiment.* 2014. URL: http://www.marcozehe.de/2013/04/05/switching-to-android-full-time-an-experiment/ (visited on 01/01/2014).

[4] *NaviTerier. navigation system in buildings for the visually impaired.* 2013. URL: http://usability.felk.cvut.cz/naviterier (visited on 01/01/2014).

[5] Zdenek Mikovec Jan Balata Miroslav Macik. “Context Sensitive Navigation in Hospitals”. In: *30th Annual International IEEE EMBS Conference* (2013).

[6] Luis A. Guerrero, Francisco Vasquez, and Sergio F. Ochoa. “An Indoor Navigation System for the Visually Impaired”. In: *Proceedings of Sensors 2012.* 2012, pp. 8237–8258. URL: http://www.ncbi.nlm.nih.gov/pubmed/22969398.

[7] N. A. Giudice T. H. Riehle P. Lichter. “An Indoor Navigation System to Support the Visually Impaired”. In: *30th Annual International IEEE EMBS Conference* 30 (2009), pp. 4435–4438.

[8] SYLVIE TREUILLET and ERIC ROYER. “OUTDOOR/INDOOR VISION-BASED LOCALIZATION FOR BLIND PEDESTRIAN NAVIGATION ASSISTANCE”. In: *International Journal of Image and Graphics* 10.04 (2010), pp. 481–496. DOI: 10.1142/S0219467810003937. eprint: http://www.worldscientific.com/doi/pdf/10.1142/S0219467810003937. URL: http://www.worldscientific.com/doi/abs/10.1142/S0219467810003937.

[9] Aura Ganz et al. “PERCEPT Indoor Navigation System for the Blind and Visually Impaired: Architecture and Experimentation”. In: *Int. J. Telemedicine Appl.* 2012 (Jan. 2012), 19:19–19:19. ISSN: 1687-6415. DOI: 10.1155/2012/894869. URL: http://dx.doi.org/10.1155/2012/894869.

[10] Diego López-de-Ipiña, Tania Lorido, and Unai López. “Blindshopping: Enabling Accessible Shopping for Visually Impaired People Through Mobile Technologies”. In: *Proceedings of the 9th International Conference on Toward Useful Services for Elderly and People with Disabilities: Smart Homes and Health Telematics.* ICOST’11. Montreal, Canada: Springer-Verlag, 2011, pp. 266–270. ISBN: 978-3-642-21534-6. URL: http://dl.acm.org/citation.cfm?id=2026187.2026232.

[11] P. Espinace et al. “Indoor Scene Recognition through Object Detection.” In: *Proceedings of the 2010 IEEE International Conference on Robotics and Automation.* 2010, pp. 1406–1413.

[12] A. Quattoni A.; Torralba. “Recognizing Indoor Scenes.” In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2009, pp. 413–420.

[13] Anna Bosch, Xavier Muñoz, and Robert Martí. "Review: Which is the Best Way to Organize/Classify Images by Content?" In: *Image Vision Comput.* 25.6 (June 2007), pp. 778–791. ISSN: 0262-8856. DOI: `10.1016/j.imavis.2006.07.015`. URL: `http://dx.doi.org/10.1016/j.imavis.2006.07.015`.

[14] J.A. Hesch and S.I. Roumeliotis. "Design and analysis of a portable indoor localization aid for the visually impaired." In: *Int. J. Robot. Res.* (2010), pp. 1400–1415.