A Touch Sensitive User Interface Approach on Smartphones for Visually Impaired and Blind Persons

Elmar Krajnc, Mathias Knoll, Johannes Feiner, and Mario Traar

FH JOANNEUM, Internet Technology
Werk-VI-Straße 46, 8605 Kapfenberg, Austria
{elmar.krajnc,mathias.knoll,johannes.feiner,
mario.traar.itm08}@fh-joanneum.at
http://www.fh-joanneum.at/itm

Abstract. This paper presents a user interface concept for touch screens which enables visually impaired or blind people to control applications. More and more people tend to switch to advanced smart phones with touch screen technology. So do blind people in order to have a powerful computer in their hands to support them. Unfortunately, a touch interface is not as easy to control as classical hardware buttons with a fixed location and haptic feedback. With advanced frameworks it should be possible to modify applications in a way to support the usage of touch screens for the visually impaired. The suggested new solution for Android mobile phones is to provide specialised "talking touch" views, such as a "talking touch list", which allow fast input with audio feedback. An early prototype version showed an already promising positive response on first usability studies with the target group.

Keywords: Touch, Accessibility, Mobile Computing, Visually Impaired, Android, iPhone, GUI.

1 Introduction

Nowadays the most common personal information devices are mobile phones. Today about 20% of the phone owners use smart phones and the number is rising [2]. Modern smart phones provide much CPU power and users have thousands of applications at their fingertips. Fingers are also important for the user interface. Normal mobile phones with a small display and a keypad are getting rare. Smart phones with only a few keys or buttons and a large touch screen are now state of the art technology. Mobile operation systems which support mostly touch screens represent about 50% of the market and it is predicted that in 2015 about 80% of all smart phones will run a touch screen supported OS [6]. For blind people the transition to up-to-date handhelds means to switch from classical phones with hardware buttons to new phones with touch technology. This shift is a huge step further from configurable "soft keys", where users do not know the meaning of a key in advance. Soft keys require to read the text (the title of the

A. Holzinger and K.-M. Simonic (Eds.): USAB 2011, LNCS 7058, pp. 585–594, 2011.

button) which describes the current action. Therefore blind people have to guess from the current context what a button is for. That kind of navigation through applications is more difficult, but with touch screens phone users have no idea at all, whether there are any buttons to click on, or where the buttons are located on the current user interface.

Android phones using the Google frameworks and Apple iPhones have some assisting technology built in (see Section 2). For others, as for example Microsoft Win Phone 7, not even screen readers are available at the moment.

1.1 Hypothesis and Expected Results

The idea to provide visually impaired and blind people with new ways to interact with the applications on their phones. This requires the improvement of the software which controls the input and output for each navigation step within an application. Selecting options and entering information should be as easy and as accurate as possible. A framework should provide developers with the required functionality. Adapting existing applications for blind people should not be too difficult. Implementing only a few modifications, by using the framework described in Section 3.2, should be enough.

The following chapters describe the context of our studies including the usage and the development constraints (Section 3.1). The current state of the art of research and also the existing support on different smart phone platforms is explored in Section 2. The overall design, the concept and the prototype development are laid out in Section 3.2. Finally, the consequences of using the new "talking touch" framework can be found in Section 3 (which describes the feedback of the usability tests also), and the outlook of ongoing development is in the last Section 5.

2 Related Work

Different aspects of usability for visually impaired people have been research topics for a long time. Audio based navigation support is discussed by Sánchez et al. [13]. Several smartphone based assistive technologies are compared in [11]. See Krajnc et al. [9] for a developer's view on mobile software development for blind people, especially the user centred interaction design for the target group. General usability standards have been set by Nielsen [12] back in 1993 and most of them still apply to interfaces on mobile phones. Special design guidelines for mobile computers are also given by Holzinger and Errath [4].

Web accessibility[1] (e.g. set text size, contrast, flash built for screen readers, image alternatives) is important, but not enough to handle modern touch screen

[1] Accessibility standards for people with disabilities can for instance be found e.g. in U.S. Section 508 where Standard 1194.26 focuses especially on desktop and portable computers (keyboards and other mechanically operated controls, touch screens, biometric identification, and ports and connectors) http://www.uspto.gov/web/offices/cio/s508/08desktops.htm.

mobile phones. The American Foundation for the Blind (AFB) still does not list any touch screen phone on their "cell phones and related software" web pages[2].

For people with impaired vision the modern smart phones provide different accessibility technologies such as screen readers and physical feedback. The *iPhone*, for example, provides a screen reader called "Voice Over"[3] which can also be controlled by gestures. If Voice Over is switched on the clicking differs from normal mode, i.e. when touching a button a corresponding description can be heard and on double-tap the button gets activated. Often, additional feedback such as a sound is generated when moving from one button to another. Gestures like swiping help to move back and forth between items or screens. Text input of long text is done by moving one finger over the soft keys and by confirming with a tap of another finger.

Simple other features like switching to "white on black" or zoom functionality are available. On powerful devices even (limited) voice control is available. The iPhone provides only music and phone book support.

The biggest problem is still to provide enough orientation and context on the screen. Fitting out smart phones with internationalisation features – providing automated spoken text in different languages – is even more difficult and expensive. Getting used to and learning the ins and outs of screen readers is not too big a problem for disabled people. Initial training pays off for blind people very soon. So hardware buttons with tactile feedback can indeed be replaced by advanced touch screen functionality. Support by means of vibration, spoken text and sounds is possible. On request bluetooth hardware can be attached. On the Apple pages a list of more than 30 supported braille devices[4] can be found.

Several assistive technologies are available for the *Android* phones, but they are neither built into the system nor are they activated by default. Users choose from different libraries for supporting their applications. On the Android, unlike on the iPhone, the developers are requested to put much effort into adapting each and every application they implement to the special needs of the visually impaired. End users must even check if any text-to-speech library is installed.

An advanced library is the "eyes free"[5] software which replaces the normal graphical interface with an eyes-free shell. The main idea is to provide radial stroke-based gestures to trigger a certain functionality. For example, the "Talking Dialer" lets users enter numbers via those gestures. On the one hand, vibrating

[2] The American Foundation for the Blind lists cell phones and related software at `http://www.afb.org/ProdBrowseCatResults.asp?CatID=74`.

[3] Voice Over: `http://www.apple.com/accessibility/iphone/vision.html`

[4] iPhone supported braille devices: `http://www.apple.com/accessibility/iphone/braille-display.html`.

[5] Find eyes-free for android at `http://code.google.com/p/eyes-free` or download TalkBack/Kickback/Soundback from `https://market.android.com/details?id=com.google.android.marvin.talkback`.

feedback assures proper selection and mistakes can be corrected (unintended input undone) by shaking the phone. Developers can look up good practices[6] for coding against the accessibility layer. This software layer provides APIs to text-to-speech, haptic feedback and trackball or directional controller (d-pad) navigation.

Windows Mobile 6.5 phones can use some accessibility features with screen readers such as Talks[7] or Mobile Speak[8]. Win Phone 7 does not offer accessibility features (except for TTY support and speech recognition). Hopefully more features will be provided in the upcoming versions[9].

Several aspects of routing (visually) impaired people are addressed by Bradley and Dunlop [3]. They focus on the human-computer-interaction (HCI) point of view, especially on the mental and physical demands for people with special needs.

Mobile or smart phones are cost-effective assistive technologies for blind people [11], therefore suitable software is requested. For visually impaired and blind people a lot of research has been done in various areas like e-learning or medicine. Only few special surveys in the field of touch screens for blind people have been performed. See for example Bigham and Cavender [1] for audio captchas which allow blind people to use this way of secure login. Holzinger et al [5] describe aspects of improving mobile applications for elderly and Sánchez et al. [13] for an approach of navigation through audio-based virtual environments for blind people. Findings about developments employing user centred design (UCD) for visually impaired people can be found in Krajnc et al. [9].

Approaches for alternative input methods include special ways to enter letters. Even special writing concepts have been invented as such as EdgeWrite [14] which introduces a mobile device friendly unistroke font. Blind Sight [10] offers auditory feedback to this users and allows them to interact with the mobile phone without looking at the screen.

Some of these related projects offer the opportunity to interact with keys and buttons. To provide an accessible user interface on touch screens we need new forms of interaction. In Slide Rule [8] the touch screen is operated with multi touch gestures. With the help of a touch screen and vibration V-Braille [7] represents Braille characters on a standard mobile phone.

[6] It is good practice on Android to make input elements focusable and provide every input widget with a content description. See at `http://developer.android.com/guide/practices/design/accessibility.html`.

[7] Nuance Talks `http://www.afb.org/prodProfile.asp?ProdID=752&SourceID=74`

[8] `http://www.optelec.nl/hulpmiddelen-voor-blinden_slechtzienden-en-dyslectici/producten/toegankelijke-mobiele-telefoon/spraaksoftware/mobile-speak.html`

[9] For accessibility on Win Phone 7 see `http://www.microsoft.com/windowsphone/en-us/howto/wp7/basics/ease-of-access-on-my-phone.aspx`. Mango Update is scheduled for fall 2011 `http://www.microsoft.com/windowsphone/en-us/cmpn/mango-overview.aspx`.

3 Conception and Implementation Results

3.1 Conception

For this paper we assume the following context: the target group comprises blind or visually impaired people who use an application for way finding (indoor and outdoor navigation). For the development of this application open source software development was used to make the results available to the community. The Android[10] framework is the first choice for developers who seek a maximum of flexibility. The development kits allow software to be run on the broadest range of hardware. In future work software for the iOS platform might be coded also, but the "experimental possibilities" on iPhones and iPads much more limited.

Once the design of the GUI was finished, its implementation started. The milestones were on the one hand providing an easy to use framework of menu and list interfaces and on the other hand the completion of one menu feature as a proof of concept. Figure 1 shows a diagram of the task at hand. A demo of the talking touch list view is shown in Figure 3.

3.2 Implemented Prototype with "Talking Touch List View" and "Talking Touch Menu View"

Views are the visible part of an activity in an Android application. They can either be defined in XML files or in source code. Due to the fact that the "Talking Touch View" is created to be used by blind and visually impaired people, distinct classes had to be designed. They provide the functionality needed. The main task for the view is inherent in the "text-to-speech" functionality, which is initialised in one abstract class called "AbstractWays4AllView". Derived from it are the classes "AbstractTalkingTouchView" and "AbstractTalkingTouchListView". The first is responsible for the menu interface and the second for the list interface. Whenever an activity is in need of one of these views, it will implement a view derived from the abstract classes (see Figure 2).

The touchable menus are highly dynamic. A menu implementation contains four areas where the user may be directed to another activity by double tapping them. Aside from menus, lists constitute a major and vital ingredient for graphical user interfaces. In order to create a list component, especially for blind people, certain conditions had to be met.

Firstly, the default list representation in Android comes along with a neat scrolling feature. Visually impaired users depend on element's with fixed positions. So this kind of scroll list is not usable for them, because they cannot remember an element's location. For this reason it was decided to build a new type of list which is presented as a set of pages.

The advantage of the list implementation is its independency from resolutions of different devices. The touchable areas are highly dynamic, too. All the features of the menu representation (e.g. text-to-speech capabilities) are also part of the

[10] Find the documentation — especially about accessibility — in the SDK online `http://developer.android.com/sdk`.

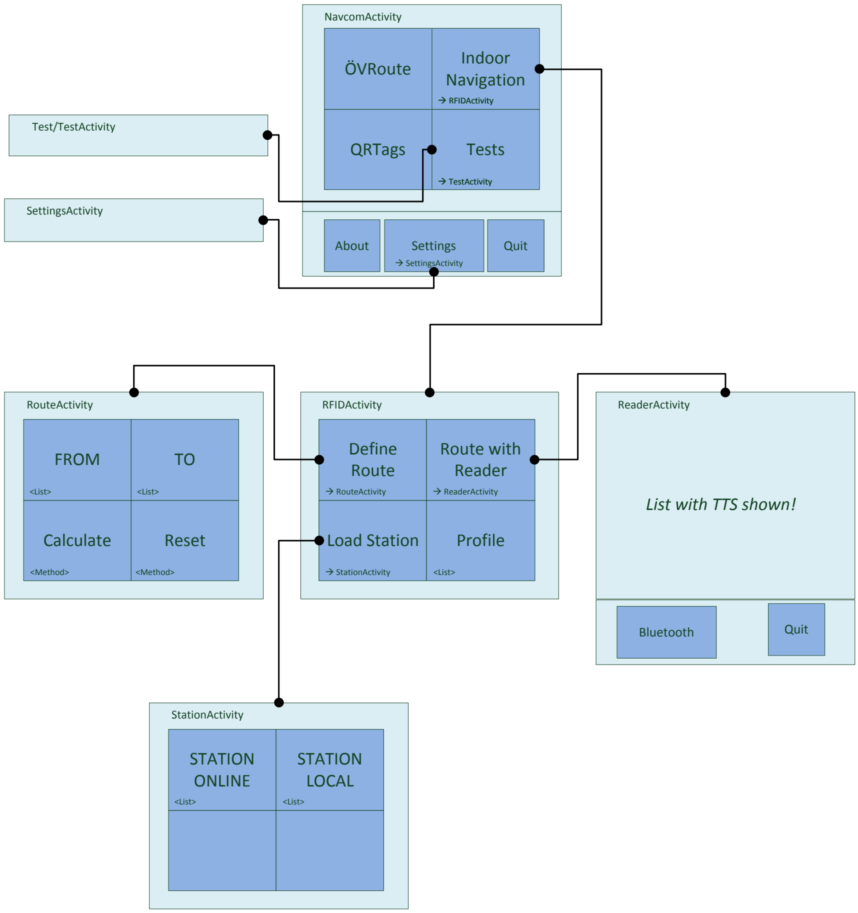

Fig. 1. The diagram showing the GUI sections

list view. A vital functionality is to provide the information where the list starts, ends, to which page someone may proceed and how many pages are left.

When starting the application the first time, a surface with several logos appears. As this is an application for visually impaired people, no further graphics are needed. For development and documentation reasons, it is however, possible to switch to a visual representation of the touchable areas of the application.

3.3 Implementation of Talking Dialer Gesture Input

Inputs from the user have so far not been covered by the concept of the Talking Touch View. There are some approaches that facilitate the textual input by blind people. For instance the Talking Dialer of the Eyes-Free project offers a special

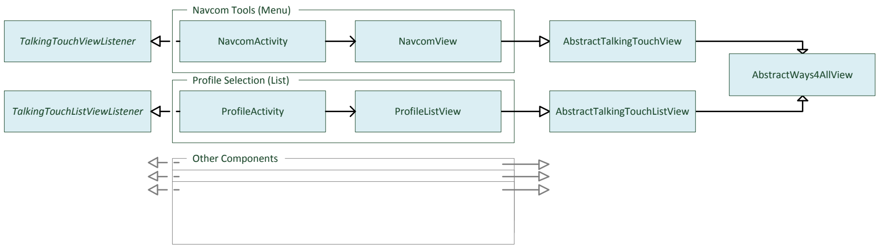

Fig. 2. Classes building one GUI element

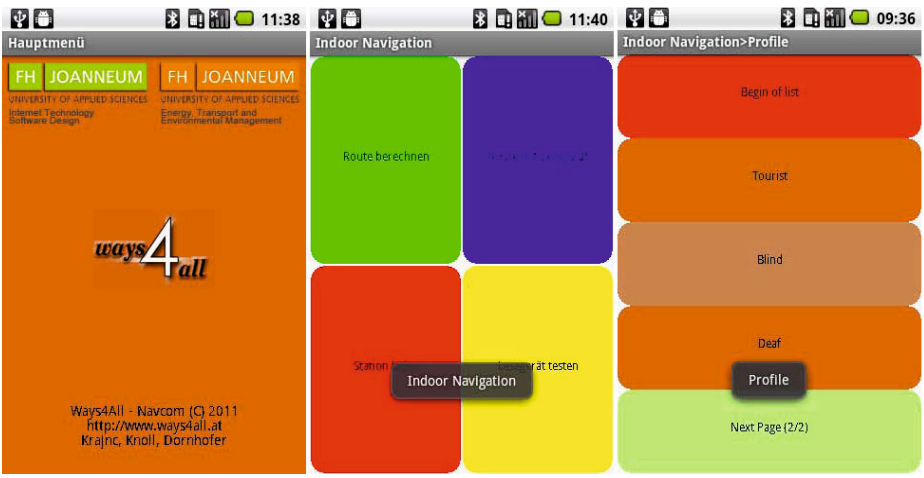

Fig. 3. The 3 screenshots of the actual application are showing the main screen, the menu areas of the "Indoor Navigation" menu and the list of profiles

input scheme or user input by means of the android gestures. In this application we made the text input method exchangeable. For example, one might choose to input via the Talking Dialer Gesture Input (see Figure 4). If the user prefer one special kind of textual input, she can change it in any way.

4 Evaluation of the User Interface

To evaluate our user interface we conducted a usability study. We recruited seven users to perform a usability test [12]. There were two blind participants, two participants with a visual disorder and three with no visual restraints. Five out of the seven persons were male and had experiences with smart phones. Two were female and had no experience with smart phones (detail see table 1).

To test the user interface we prepared a Thinking Aloud (TA) test [12]. After a short introduction to the touch screen device the participants got assignments.

Fig. 4. The screenshots of the textual input with gestures

Table 1. Test participants

	Age	Gender	Visual Faculty	Experience with Usability Testing	Experience with Smart Phones
Testperson 1	31	m	blind	yes	yes
Testperson 2	29	f	Blind	no	no
Testperson 3	26	m	normal	no	yes
Testperson 4	30	f	visual disorder	no	no
Testperson 5	23	m	normal	no	yes
Testperson 6	22	m	normal	yes	yes
Testperson 7	35	m	visual disorder	no	yes

They had to perform six different tasks on the talking touch user interface and four tasks on the gesture interface. They were, for instance, asked to "change the profile to *Blind*" or to "enter the telephone number *543289*". Four testers could successfully finish all tasks of the test. The Talking Touch Interface had a success rate of 88% and the gesture application had a success rate of 89% (find the details in Table2).

According to the feedback given by the participants after the usability test, the navigation and the position of the items were rather simple and easy to access. The selection with a double click is very intuitive and was recognised without any explanation. Some minor problems occurred when the users switched from the four-items menus to the list menu. Also the speed of the voice output was sometimes too fast, but the speech rate could be changed in the general settings. The feedback given after selecting a menu item was sometimes not precise enough.

Table 2. Testresults

	TP1	TP2	TP3	TP4	TP5	TP6	TP7
Ways4All Task 1	ok	ok	ok	ok	ok	ok	ok
Ways4All Task 2	ok	ok	ok	ok	ok	ok	ok
Ways4All Task 3	ok	-	ok	ok	-	ok	ok
Ways4All Task 4	ok	-	ok	ok	-	ok	ok
Ways4All Task 5	ok	ok	ok	ok	ok	ok	ok
Ways4All Task 6	ok	ok	ok	ok	-	ok	ok
Gesture Dailer Task 1	ok	ok	ok	ok	ok	ok	ok
Gesture Dailer Task 2	ok	ok	ok	ok	-	-	ok
Gesture Dailer Task 3	ok	ok	ok	-	ok	ok	ok
Gesture Dailer Task 4	ok	ok	ok	ok	ok	ok	ok

The feedback of the gesture input was good. It was very intuitive and most participants had no problems (One problem was reported, which depends on various writing styles of numbers and letters). All the reported problems did not stem from the basic concept. Therefore, any necessary adaptations can be implemented in future versions of the software without much effort.

5 Future and Outlook

Further research with additional output and input facilities is necessary and already planned as the next steps. For example, the force feedback (vibration) and audible output (fait or loud sound in low or high frequency) will be used as feedback to users. With different shake gestures users can provide input (acknowledge or reject the current selection). The visually impaired - as a marginal group in our society - are awaiting the results which can really improve their mobility and independency in their daily lives.

References

1. Bigham, J.P., Cavender, A.C.: Evaluating existing audio CAPTCHAs and an interface optimized for non-visual use. In: Proc. 27th International Conference on Human Factors in Computing Systems (CHI 2009), pp. 1829–1838. ACM (2009)
2. BITKOM: Jeder fünfte handynutzer besitzt ein smartphone (telekommunikation und neue medien e.v. bundesverband informationswirtschaft. presseinformation) (October 2010), `http://www.bitkom.org/de/themen/54894_65506.aspx`
3. Bradley, N.A., Dunlop, M.D.: An experimental investigation into wayfinding directions for visually impaired people. Personal Ubiquitous Computing 9(6), 395–403 (2005)
4. Holzinger, A., Errath, M.: Mobile computer web-application design in medicine: some research based guidelines. Univers. Access Inf. Soc. 6, 31–41 (2007), `http://dl.acm.org/citation.cfm?id=1283708.1283718`

5. Holzinger, A., Searle, G., Nischelwitzer, A.: On some aspects of improving mobile applications for the elderly. In: Proc. 4th International Conference on Universal Access in Human Computer Interaction (UAHCI 2007), pp. 923–932. Springer, Heidelberg (2007)
6. IDC: Idc forecasts worldwide smartphone market to grow by nearly 50 (March 2011), http://www.idc.com/getdoc.jsp?containerId=prUS22762811
7. Jayant, C., Acuario, C., Johnson, W., Hollier, J., Ladner, R.: V-braille: Haptic braille perception using a touch-screen and vibration on mobile phones. In: Proc. 12th International ACM SIGACCESS Conference on Computers and Accessibility (ACCETS 2010), pp. 295–296. ACM (2010), http://doi.acm.org/10.1145/1878803.1878878
8. Kane, S.K., Bigham, J.P., Wobbrock, J.O.: Slide rule: Making mobile touch screens accessible to blind people using multi-touch interaction techniques. In: Proc. 10th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS 2008). pp. 73–80. ACM (2008)
9. Krajnc, E., Feiner, J., Schmidt, S.: User Centred Design Interaction Design for Mobile Application Focused on Visually Impaired and Blind People. In: Leitner, G., Hitz, M., Holzinger, A. (eds.) USAB 2010. LNCS, vol. 6389, pp. 195–202. Springer, Heidelberg (2010), http://www.springerlink.com/content/c3813w7857315241/
10. Li, K.A., Baudisch, P., Hinckley, K.: Blindsight: Eyes-free access to mobile phones. In: Proc. 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI 2008), pp. 1389–1398. ACM (April 2008)
11. Narasimhan, P., Gandhi, R., Rossi, D.: Smartphone-based assistive technologies for the blind. In: Proc. 2009 International Conference on Compilers, Architecture, and Synthesis for Embedded Systems (CASES 2009), pp. 223–232. ACM (2009)
12. Nielsen, J.: Usability Engineering. Morgan Kaufmann (1993)
13. Sánchez, J., Sáenz, M., Pascual-Leone, A., Merabet, L.: Navigation for the blind through audio-based virtual environments. In: Proc. 28th International Conference Extended Abstracts on Human Factors in Computing Systems (HIEA 2010), pp. 3409–3414. ACM (2010)
14. Wobbrock, J.O.: EdgeWrite: A Versatile Design for Text Entry and Control. Ph.D. thesis, Human-Computer Interaction Institute School of Computer Science Carnegie Mellon University (July 2006), http://reports-archive.adm.cs.cmu.edu/anon/hcii/CMU-HCII-06-104.pdf