# Accessibility for the Blind on an Open-Source Mobile Platform

## MObile Slate Talker (MOST) for Android

Norbert Markus[1], Szabolcs Malik[1], Zoltan Juhasz[2], and András Arató[1]

[1] Laboratory of Speech Technology for Rehabilitation,
Institute for Particle and Nuclear Physics, Wigner Research Centre for Physics,
Hungarian Academy of Sciences, Budapest, Hungary
[2] Dept of Electrical Engineering and Information Systems,
University of Pannonia, Veszprem, Hungary

**Abstract.** As Android handsets keep flooding the shops in a wide range of prices and capabilities, many of the blind community turn their attention to this emerging alternative, especially because of a plethora of cheaper models offered. Earlier, accessibility experts only recommended Android phones sporting an inbuilt QWERTY keyboard, as the touch-screen support had then been in an embryotic state. Since late 2011, with Android 4.X (ICS), this has changed. However, most handsets on the market today -especially the cheaper ones- ship with a pre-ICS Android version. This means that their visually impaired users won't be able to enjoy the latest accessibility innovations. Porting MObile SlateTalker to Android has been aimed at filling this accessibility gap with a low-cost solution, regarding the special needs of our target audience: the elderly, persons with minimal tech skills and active Braille users.

**Keywords:** (e)Accessibility, Blind People, Assistive Technology, Braille, Usability and Ergonomics, (e)Aging and Gerontechnology, Mobility, Android.

## 1 Introduction

MOST for Android is a follow-up of the earlier MOST projects [1][4] aimed at porting MObile SlateTalker (MOST) to Android. It was launched in early 2011 and at the time of writing, it is just being made publicly available.

The MOST development team has spent the last year exploring the potential of the Android operating system. During this time, the team has migrated the MOST framework to the Android platform, fixed known bugs in the open-source eSpeak TTS, ported our own BraiLab speech engine to Android and updated the design of the special Braille mask for the touch-screen to include the D-pad navigation controls and holes for extra functions in addition to the usual Braille holes. This mask is an essential part of any MOST powered device and has been evolving along with the subsequent MOST versions over the last decade.

K. Miesenberger et al. (Eds.): ICCHP 2012, Part II, LNCS 7383, pp. 599–606, 2012.

The existing users of the pre-Android MOST devices are a poorly represented segment of the community, visually impaired smartphone users. These users may have little or no experience with computers or be elderly but they can communicate using Braille. Such persons remain the target audience for the new version of MObile SlateTalker.

## 2 State of the Art

**MObile SlateTalker (MOST)** [1] is a specially designed mobile software suite for the blind with dozens of useful applications making use of the phone's touch screen. It offers an unified menu system and promotes the daily usage of Braille by a proprietary input method that provides fast text entry, especially when entering non-English texts.

The MOST Consortium maintains and develops the MOST software and related hardware since 2004 with the active contribution of about 100 blind users in Hungary. Most of them are elderly or persons with minimal tech skills.

**BrailleTouch** [5] is an eyes-free text entry application for multi-point touchscreen mobile devices developed by Mario Romero and his team at Georgia Institute of Technology. It uses the chorded system of Braille and swipe gestures to input text. It does not involve any TTS. On Android, it can perform any text entry task. On iOS, it is a stand alone application that does not input text to the phone. It is and will always be free for any platform. Practically, it's not going to work on MOST Android handsets having a touch-screen hardware capable of handling only the most basic one- and two-finger gestures. As of 17th April, BrailleTouch is not yet publicly available.

**Mobile Accessibility for Android** [6] by Code Factory is an Android home screen application that incorporates a program suite of 10 applications simplified for the blind, a screen reader and an embedded Nuance Vocalizer TTS engine. Its concept is somewhat similar to that of MOST, in that it offers an unified environment for blind users, but it does not provide on-screen Braille input. Instead, the user is encouraged to use simplified gestures anywhere on the touch screen and a dynamic virtual keyboard for text entry. A full version can be downloaded from the Android Market for 70 Euros, or its 30-day trial version for free. The purchased app ships with a TTS whose language is included in the name of the app version.

## 3 How It Works

MOST is not a screen reader and does not require the presence of a screen reader installed on the same device. While using it, no graphical information is presented to the user. Instead, any pieces of data are channeled via pre-recorded and synthetic speech.

The MObile Slate Talker framework [1] provides a user interface that allows blind users to operate their device by simple touch actions, while getting an immediate

audio feedback. In the MOST system, all applications, settings and features are organized into a unified menu tree that can be explored by the sole use of the four arrows. Character entry is realized by an on-screen Braille input method. A special mask is placed over the touch screen, turning it into a digital Braille slate. During Braille text input, the arrows are used for cursoring in the text.

Our 2010 survey[4] showed that both this simplicity of operation and the availability of the Braille entry mode were praised by an overwhelming majority of our users.

## 4 Background and Objectives

A growing number of smartphones today have a highly graphical user interface based on interacting with the touch-screen dominating the front side of the device. At the same time, many blind users (mostly the women and the elderly) worry that smartphones having physical numeric keys and buttons are on the verge of extinction

On a "traditional" smartphone, the physical keys and buttons are somewhat elevated, thus they remain tactile. In addition, their location is fixed all the time regardless of the application or in which mode the phone is currently being used. These factors give the blind user a feeling of a higher degree of security when interacting with the device. Consecutive MOST projects [3][4] have addressed this issue from the earliest stages of the design and development. MObile SlateTalker has always been used on touch-screen based hand-held devices (first on Pocket PC and Windows Mobile powered PDAs, and now also on Android handsets).

The special user interface for visually impaired users is provided by a transparent mask template placed over the touch-screen. Holes on this mask guide the users' fingers to dedicated areas of the screen. Thereby MObile SlateTalker turns a touch-screen operated smartphone into a device with quasi push-buttons and keys that become tactile and preserve their location on the screen throughout the operation. As opposed to this, designers of the most recent screen-readers seem to be forced to assign more complicated gestures for their visually impaired audience than the ordinary sighted smartphone users use.

Complications culminate around the problem of text input. Entering text (especially when abound with accented letters or special characters), on a flat glass surface, remains a partially resolved issue even with the best effort and concerted haptic, auditory and speech support. Perhaps this is why even the most advanced users cannot resist the temptation to obtain an external keyboard of some kind to couple it with their "accessible" smartphone.

MObile SlateTalker's Braille-mask-based input method is effective enough to eliminate this temptation on all target devices. Our 2006 paper [2] details the characteristics of the learning process of our test users and lists the typical entering speeds for sequential Braille dot entry on the MOST Braille mask. Recently the average entering speed for Hungarian uncontracted Braille is around 13 words per minute with a maximum at around 19 wpm.

# 5 Market Assessment of Accessible Smartphones in Hungary

Apple appears to be the sole manufacturer so far to have come up with a mobile product family offering a fully integrated, fully functional gesture based screen reader (VoiceOver) free of charge in all the languages supported by the iOs operating system. On the other hand, iOs devices occupy the high-end of the cell phone market and this fact is reflected in their prices. This may contribute to the fact that according to our recent survey, there are only about 30 blind iPhone users in Hungary and they are all men.

The elderly and female smartphone users are known to prefer the more conventional Symbian phones having physical keys and pushbuttons, and MObile SlateTalker powered devices.

## 5.1 Methodology Used for Conducting the Survey

Data for the survey comes from two technically separate sources. a) basic pieces of data (e.g., age, gender and device model used) from 90 of our registered users have been selected, updated and verified. b) Over 100 members of a smartphone oriented mailing list operated by a foundation under the umbrella of the Hungarian Association of the Blind and Partially Sighted have been selected for inclusion in the study, based on their public statements within a period of approx. 13 months (from Feb. 10 2011 through Apr. 15 2012). The information collected has recently been updated and verified. Eventually 82 list members have provided sufficient data qualifying them for inclusion in this study.

## 5.2 Findings of the Survey

In the table below, numbers represent actual persons or device ownership rather than percents. The "Persons" column on the left contains numbers refering to individuals (phone users), while in the subsequent four columns to the right, you can find the number of certain types of handsets owned by these individuals. In the column labeled "MOST", numbers refer to pre-Android MObile SlateTalker devices. The "Android" column shows the number of pre-ICS Android handsets used only with the standard accessibility features enabled. Among the iOs devices listed, only 32 are actual iPhones, the remaining two are iPod Touch models.

In each column, the number of males (m) and females (f) is listed along with the total (t) number of users in all age groups. The row labeled "Only" at the bottom of the table shows the number of users that -according to the data available- only have a single smartphone of the kind specified by the column.

The survey provided data about 154 visually impaired individuals using accessible mobile devices in Hungary. 109 of them are male and 45 are female. Assessing the totals, MObile SlateTalker appears to be the most favored by women, followed by Symbian. It is striking to find no females among the iOs and Android users (probably the former being less affordable, while the latter considered to be too experimental).

The age distribution in the "Persons" column yields a peak at the 30-39 year age group, with the number of females dropping sharply over the age of 60.

In the "Symbian" column, the ages are more smoothly distributed with a modest peak at the age group of 30-39 years, but the number of women already drops over the age of 50.

In our aspect, the most interesting is the "MOST" column. It differs from the other columns almost in all possible ways. First of all, the peak appears in an older age range, namely between 50 and 59 years. Actually the majority of this user group are above the age of 50. Women constitute one third of the MOST users which is the highest representation of females among the user groups. The 20-29 age range is unique because this is the sole age group of the entire survey of which females constitute the majority.

**Table 1.** Results of the Survey on Visually Impaired Smartphone Users in Hungary

| Age | Persons | | | Handsets | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MOST | | | Symbian | | | iOs | | | Android | | |
| | F | M | T | F | M | T | F | M | T | F | M | T | F | M | T |
| 10 - 19 | 1 | 2 | 3 | 0 | 1 | 1 | 1 | 1 | 2 | 0 | 1 | 1 | 0 | 0 | 0 |
| 20 - 29 | 11 | 17 | 28 | 5 | 4 | 9 | 8 | 12 | 20 | 0 | 4 | 4 | 0 | 2 | 2 |
| 30 - 39 | 12 | 28 | 40 | 8 | 10 | 18 | 6 | 19 | 25 | 0 | 11 | 11 | 0 | 4 | 4 |
| 40 - 49 | 8 | 20 | 28 | 4 | 10 | 14 | 6 | 16 | 22 | 0 | 7 | 7 | 0 | 2 | 2 |
| 50 - 59 | 9 | 20 | 29 | 9 | 15 | 24 | 1 | 6 | 7 | 0 | 0 | 0 | 0 | 1 | 1 |
| 60 - 69 | 3 | 18 | 21 | 3 | 16 | 19 | 0 | 4 | 4 | 0 | 3 | 3 | 0 | 0 | 0 |
| 70+ | 1 | 4 | 5 | 1 | 4 | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Total | 45 | 109 | 154 | 30 | 60 | 90 | 22 | 58 | 80 | 0 | 26 | 26 | 0 | 9 | 9 |
| Only | | | | 55 | | | 42 | | | 8 | | | 1 | | |

The results of this survey show that addressing the special needs of the target users (in our case the elderly and women) in the design and ergonomics, the simplicity and affordability are reflected in the users' preferences and in their choice of product.

## 6 Porting MOST to Android

As the first technical details of the Android operating system were unveiled several years ago, it occurred to be an appropriate choice to migrate the next generation of the MOST software to. Android seemed to be a promising target especially because it had been declared to be an open-source cell phone platform.

Before the first actual Android phones hit the shops, technical reviews kept emphasizing Android's inherent advantages, its flexible nature and novel system architecture, and brand new ways the system and the applications could interact with each

other. Any system components (home screens, widgets, services, input methods, etc.) might be replaced with third-party alternatives or custom-built ones.

At that stage, one could have the impression that a third-party developer would have full control over the operating system. We certainly hoped that any of its behaviors might be modified for the benefit of the visually impaired users. In the subsequent sections, we are going to discuss these issues in detail with regard to their accessibility implications, how such obstacles can be overcome, and deal with the question whether Android can finally be considered an open system or not.

As real Android handsets became available, soon it was obvious that the promises and the reality weren't in perfect unison. The practices introduced then persist still today. Handsets from the major manufacturers ship with a locked system ROM, which means that the pre-installed factory image cannot be modified at all, and the owner of the phone is degraded to a simple user with restricted rights in the system.

MOST relies on a fixed screen layout determined by the Braille mask placed over the touch screen. This fixed layout must be kept persistently on the screen within MOST. The Braille mask will not and cannot be moved or removed for the sake of individual user operations. Therefore MOST does what it can to prevent the screen layout from changing unexpectedly.

However, certain system dialogs and activities cannot be prevented from showing on the screen or be replaced by custom ones or it just makes no sense to do so without the required permissions.

For instance, a long press of the POWER button brings up the shutdown dialog that cannot be suppressed or replaced with a custom one. Anyway, invoking a call to shut down the device is bound with a permission not granted to a third-party app.

A long press of the HOME button brings up the recent apps list dialog, that cannot be dismissed from within a third-party app although what it provides could be reproduced with the standard SDK. It is the same with the "Battery is low", "An application is not responding" and "An application has kept the device awake for a long time" dialogs.

Defining a fixed screen layout that persists all over the system feels to be a bumpy road. For entering characters in Braille whenever it is needed, the best choice is to develop and install a custom Braille input method. However, it is a multistep procedure to allow a custom input method to be available all across the system, and even then it will only show up when the user moves the focus onto an input field. Further on, an input method can get covered or hidden unpredictably by a number of things.

As an alternative, a custom built application widget with a fixed size and screen location may hold an unchanging layout within the boundaries of the widgets own context while shifting the rest of the screen content out of the way. Blind-friendly input methods and a fixed location virtual D-Pad have been introduced by the Eyes-Free Project with Android 4.X and through a compatibility package they are meant to work on earlier Android platforms as well. However, on several pre-ICS devices, both these input methods and the virtual D-Pad get easily swept away by opening the default home screen's application menu, or when a system dialog mentioned earlier or a dialing screen or in-call activity is shown.

All this means that whenever the user moves (or is moved) outside the MOST environment, keeping the fixed Braille layout on the screen becomes a challenge. At this time, the only workaround seems to be to enable targeted accessibility services or the system's default accessibility feature as a last resort and to use the inbuilt physical keyboard of the phone.

Another complicating factor is that the handset manufacturers and even some carriers tend to "temper" with the top layer of the Android system. Most often it affects the look and feel of the user interface but these manufacturer-designed layers are usually more than simple custom skins, and they always constitute a non-open-source portion of the system. In certain cases, the custom layer even affects the performance of the accessibility functions and services.

Recording a phone call using the standard android.media.AudioRecord functions is not possible. However, this is an undocumented "feature" that comes with unexpected consequences. In the best case, the recording will simply be unusable. If a phone call is answered or initiated while an audio recording is in progress, the sound file being created may be left incomplete and therefore unusable. In the worst case, the system may crash so severely that it can't even be restarted.

The workaround involves pausing the recording when a call is answered or placed and resuming the recording after the call has ended.

On Motorola handsets, a double tap on the HOME button performs the so-called "default operation" that may be chosen from a list found among the application settings in the system. There is no way to suppress or replace this feature.

When one enables the system's accessibility service, it implicitly sets the "Caller ID readout" setting affecting the phone ringer to "Caller ID repeat" and this is applied every time the phone is rebooted. The workaround involves fumbling around with the system-wide stream volume settings and playing phone and alarm ringtones from within the MOST program.

Certain operations that an Android app may want to perform rely on a related system permission. Most such permissions may be granted by the application to itself, but a number of these self-issued permissions are rejected by the system. This is a rather poorly documented area of the system APIs and the SDK and hindered our development work already at the design stage.

Some of these rejected permissions are related to quite trivial functions such as setting the system clock's date/time, calling privileged numbers (i.e., placing emergency calls directly), enabling the speakerphone, muting the microphone or feeding DTMF codes into the phone line.

As an alternative to the regular consumer phones mentioned above, Google provides the Nexus family as a so-called Dev Phone product line for advanced developers. These phones are SIM-unlocked and system unlocked, but they are located in the higher price segment of the smartphone market and are not available in all regions. Such phones' ownership is restricted to registered Android developers and copy protected apps are not available for Dev Phones for download. These circumstances make them largely inadequate for most non-developer users.

With the restrictions on a regular consumer phone, the environment in which an Android app must run tends to resemble the conditions on an iOs device, but it must be stated that even so there is a lot more freedom for a third-party developer on the Android platform.

## 7 Conclusion

As PDAs went out of fashion, the MOST Consortium had to find a new range of target devices for MObile SlateTalker. By Q3 2011 Gartner estimated more than half (52.5%) of the smartphone market belonged to Android. The same research company also estimated that Android was going to dominate the entire mobile market soon. Therefore Android handsets appeared to be the best choice for us.

Since affordability was known to be among the main priorities of our users, we had to extend our software support to handsets sold at more modest prices. These tend to run on pre-ICS android versions on which the in-built accessibility functions are still in a kind of half-baked state.

The recent Android port of MObile SlateTalker with its own input method via a special mask template provides an affordable solution for our target audience who are otherwise poorly represented on the accessibility market and whose special needs are still to be addressed by the major players, at least in Hungary. This paper summarizes our efforts and results in overcoming the often unforeseen obstacles and unleashing the potentials of Android.

## References

1. Arato, A., Juhasz, Z., Blenkhorn, P., Evans, G., Evreinov, G.: Java-Powered Braille Slate Talker. In: Miesenberger, K., Klaus, J., Zagler, W.L., Burger, D. (eds.) ICCHP 2004. LNCS, vol. 3118, pp. 506–513. Springer, Heidelberg (2004)
2. Juhasz, Z., Arato, A., Bognar, G., Buday, L., Eberhardt, G., Markus, N., Mogor, E., Nagy, Z., Vaspori, T.: Usability Evaluation of the MOST Mobile Assistant (SlatTalker). In: Miesenberger, K., Klaus, J., Zagler, W.L., Karshmer, A.I. (eds.) ICCHP 2006. LNCS, vol. 4061, pp. 1055–1062. Springer, Heidelberg (2006)
3. Markus, N., Juhasz, Z., Bognar, G., Arato, A.: How Can Java Be Made Blind-Friendly. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) ICCHP 2008. LNCS, vol. 5105, pp. 526–533. Springer, Heidelberg (2008)
4. Márkus, N., Arató, A., Juhász, Z., Bognár, G., Késmárki, L.: MOST-NNG: An Accessible GPS Navigation Application Integrated into the MObile Slate Talker (MOST) for the Blind. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) ICCHP 2010. LNCS, vol. 6180, pp. 247–254. Springer, Heidelberg (2010)
5. Romero, M.: Brailletouch Mobile Texting for the Visually Impaired. Georgia Institute of Technology (USA), http://calebsouthern.com/papers/frey_southern_romero_2011.pdf
6. Mobile Accessibility for Android by Code Factory, http://www.codefactory.es/en/products.asp?id=415