TECHNIQUES FOR IMPROVING ACCESSIBILITY OF MOBILE APPLICATIONS FOR BLIND AND VISUALLY IMPAIRED USERS

Casey Burkhardt
Villanova University
Department of Computing Sciences
Villanova PA 19085 USA

ABSTRACT

This paper provides a synopsis of several methodologies for improving the accessibility of mobile devices and applications for persons who are blind or visually impaired. Areas of focus include an analysis of different input methods, including tactile key presses and gesture-based touch screen input. Also, the paper looks closely into the non-visual device responses in terms of verbal and non-verbal auditory and haptic feedback. In addition, the concepts of environmental and context-aware accessibility are introduced in terms of their applicability in the mobile device space.

Finally, a research project, dealing with the implementation of a mobile application with context-aware accessibility features is outlined. It is suggested that this project be developed in parallel across various mobile platforms to allow for comparison of available accessibility software development kits and developer resources.

KEY WORDS
accessibility, mobile devices, ease of access, blind, visually impaired

1. INTRODUCTION

As mobile devices become increasingly prominent in the consumer market, it is important to address the accessibility of these devices to both mainstream users and persons with disabilities. Mobile devices and smart phones, portable media players, and personal digital assistants are becoming more and more popular in the daily lives of many, and are finding ways in which these devices can be made more accessible is an interesting and important challenge. In addition, the trend of development and use of specialized mobile applications opens an opportunity to relate this growing area of computing to software engineering. By identifying useful and feasible techniques for implementing accessibility features at the application level, application publishers can ensure their products reach the same customer base as naively accessible devices. This paper will provide background into the current mobile device environment, and provide techniques and use cases for the implementation of accessibility features relevant to blind and visually impaired end-users.

2. ACCESSIBILITY AND MAINSTREAM USAGE CONCERNS

Although improving mobile device and application accessibility is an excellent practice for inclusion of sight impaired users, it has an impact on the overall customer base. Users are increasingly looking for solutions which allow them to multitask, the definition of the usage scenario for many of today's common mobile devices. Allowing a device or application to be used "eyes free" adds an extra incentive for this use case. Permitting mainstream users to access a feature or perform a task on a modern mobile device without having to visually interact with the user interface is an intuitive and often underestimated capability. In addition to aiding to the accessibility spectrum, all users benefit from the change, especially if it can be implemented in such a way that it does not interfere with other visually-intractable elements located on the interface [1]. This can best be accomplished by using an alternate input method or interaction technique that performs the same action as manipulating a visual object on the device's display. One can only imagine the countless cases that such a feature would be useful to the average user.

3. NATIVE CAPABILITIES VERSUS MOBILE APPLICATION SCOPE

Throughout the discussion of accessibility in the mobile device space, it is important to note the differences between native device capabilities and application specific capabilities. The earlier is often limited by the implementation of the device's core platform, operating system, or physical sensors. The latter is often bounded by the development efforts of the third party developing the application, and the limitations put on them by the native capabilities. This paper will investigate some of these bounds with respect to specific accessibility techniques, but in general, the overall factor keeping device-wide accessibility from occurring is in fact the actual limits placed on developers by the authors of the various software development kits used to create applications on the individual devices. If these libraries offered the ability to easily implement accessibility-related features, the application developers and publishers would certainly take the time to reach the full potential of their customer base.
4. ACCESSIBILITY TECHNIQUES

The discussion of specific techniques for improving accessibility at both the device and application level in the mobile space can be best derived by looking at the areas which limit the present accessibility of many devices. Specifically, blind and visually impaired users encounter problems especially in terms of dealing with indicating intent to and receiving feedback from the device [2]. In addition, the ability for the mobile device to adapt to the present environment by contextually enabling certain accessibility features is a relevant concern.

4.1. USER INPUT

The problem of allowing blind or visually impaired users to effectively, accurately, and quickly enter data into a mobile application is not a trivial task. There are several different methodologies which allow for some level of reasonable interaction, but it is proposed that the optimal level of interaction has not yet been reached. Many of the concerns in this specific area relate to the actual size of these devices. They are, by their very nature, quite small, and therefore require the mechanisms for data entry to be very small as well. A number of devices also rely solely on the use of a capacitance-based touch screen for allowing the user to communicate intent. Unfortunately, this is a less than reasonable solution for many people with visual impairments, as the use of a "soft keyboard", one which does not have any tactile responsiveness, is not possible.

4.1.1. TACTILE KEY PRESSES

The use of a physical keyboard is perhaps the most effective and accurate, although not the fastest way for a person who is blind or visually impaired to perform data input tasks on a mobile device. It has been seen that users who do not have visual access to the device are able to maintain a reasonable level of accuracy performing tasks on a tactile keypad or keyboard [3]. However, there is still the limitation of the size of the physical keyboard. Often times, using punctuation or other special characters is difficult due to the necessity of key-combination presses being required to achieve the desired result. As a general rule, if the device has a physical keyboard, or the capability of a physical keyboard, applications should support input from that source, if appropriate.

Speed is also a serious concern for the use of physical keyboards. Although allowing for more accuracy, traversing lists of options or data is not a trivial task, and can take some time to locate the desired item. One strategy for combating this frustration is allowing the list to be sorted lexicographically and shortcuts placed at the beginning of each lettered section, thereby allowing the user to skip to a certain segment of the list. Although helpful, in practice, this can still be a time consuming exercise for an end-user.

4.1.2. TOUCH SCREENS AND GESTURE BASED INPUT

As more and more devices trend towards the use of capacitance or pressure sensitive touchscreens, there has been an increasingly strong movement to make this type of interaction reasonable for blind and visually impaired users. The idea of "gestures" appears to be at the forefront of possibilities to bring accessibility to this type of device. As seen in "Slide Rule", the developers of the user study created an application on the iPhone OS platform which allowed for hierarchical list traversal based on the vertical and horizontal scrolling with a finger through a series of menus. Secondary gestures, such as "tapping" or "double tapping" indicated intent and committed an action. Tertiary gestures, such as "swiping" to the left or right changed to the next view or application [4]. This model’s popularity is certainly with cause, as it allows for structured and reliable navigation within an application, but lacks in that it does not provide for a way to enter actual data into a field or express intent insofar as selecting a letter or number.

A solution to this problem was proposed and implemented on the Google Android platform by researcher T.V. Raman and developer Charles Chen. The idea of the "wheel" as a way for someone with limited sight to enter data is an intuitive and flexible approach to the problem [5]. In this implementation, a user places a single finger on a touch screen device and a circular grid of letter groupings (A-H, I-P, etc.) appears on the screen. The user then moves their finger to the desired grouping based on the section of the alphabet they are targeting, and the grid of groupings is replaced with the actual letters, numbers, or punctuation from that particular group. Moving their finger in the desired direction again yields in the entry of the desired character. Shaking the devices triggers the built-in accelerometer to initiate a backspace. Although this form of data entry takes some training and practice, it yields a reliable and easy to use methodology for entering data.

Figure 1: Screenshots of Raman's and Chen's implementation of the “wheel” data entry approach.
4.2. DEVICE FEEDBACK

Another concern for the accessibility of mobile devices for blind and visually impaired users is receiving feedback from the device when an action occurs on screen. This problem is particular notable because it is the key to interaction with a given application. This problem also serves as the most important in terms of bringing eyes-free access to mobile devices to mainstream users as well. Both auditory feedback and haptic feedback will be analyzed both in terms of their feasibility and effectiveness.

4.2.1. AUDITORY FEEDBACK

Auditory feedback refers to sound that is emitted by a mobile device, whether by an internal speaker or external accessory (such as a Bluetooth headset), that indicates an on-screen action has occurred. These actions may include a new alert, an update to an item in an open application, or even an indication that the user as expressed intent by pressing on a button within the interface. There have been several implementations and user studies where the effectiveness of auditory feedback has been analyzed. Often times, text-to-speech technology has been used to communicate an event, but there have been implementations of accessibility related features where non-speech audio has been used with a similar level of effectiveness [3][5]. There have been some challenges in the text-to-speech realm specific to mobile devices due to the limited computational power of these devices. As they were not originally intended to support text-to-speech features, many do not have the ability to generate speech output on the fly, so they often rely on pre-generated or predictive text-to-speech.

4.2.2. HAPTIC FEEDBACK

Haptic feedback refers to the use of vibration or other physical stimulus to indicate that an action has occurred. Most mobile devices contain some type of structure that allows for haptic feedback to occur, but its uses are extremely limited. Primarily this type of feedback has been used only to alert the user to some type of action, but not necessarily communicate what that action was. However, by alerting a pattern of haptic feedback based on an event type, a device could potentially communicate more information about an event than just its occurrence. Presently, most accessible solutions utilize this type of feedback to confirm that the user has selected an intractable interface item such as a letter on a "soft keyboard" or a button that performs some significant action.

4.3. CONTEXTUAL ACCESSIBILITY

One of the most interesting conceptions of accessibility within mobile devices and applications relates to the contextual application of accessibility features. By using this implementation, user-customizable accessibility features are enabled and disabled automatically based on what the mobile device senses regarding the surrounding environment. The use of artificial intelligence and machine learning allows for a quite intuitive experience for users who are blind or have low-vision [6]. To provide a more accurate illustration of this type of accessibility feature, some examples should be defined.

- Using the device's accelerometer as a mechanism for measurement, target areas for user interface buttons, soft keyboards, or other intractable elements are enlarged if the device is in motion.
- If the mobile device is in a low-light environment, screen contrast is increased for greater visibility.
- If the device is in a loud environment, automatically increase the volume of auditory output.

5. FUTURE WORK

- Contextual Accessibility Implementation and User Study – Work to develop a mobile application which provides support for context-aware accessibility features by using the guidelines set forth by Kane [6]. After development, conduct a user study with a number of blind or visually impaired users and compare the implementation to a similar application without context-aware capabilities.

- Evaluation of Current Accessibility Software Development Kits – Conduct a study that evaluates the accessibility features and advantages provided by the currently available software development kits for various mobile platforms. It would be necessary to take into account not only the effectiveness of features provided, but also the ease of implementation for mobile application developers.

- Develop an open standard for Accessibility Frameworks - After identifying the most important and useful features amongst the available accessibility related software development kits, it would be best to unify their implementations into a single standard. Such a standard would define the ways in which these developer resources should work and how their implementations can be simplified and unified.

6. CONCLUSION

Having seen some of the methods which can be utilized to better enhance the accessibility of mobile devices and applications, the question of implementation can now be addressed. It is necessary to investigate how these important and user-experience-enhancing features can actually make their way into mainstream technology products. In order to make this happen, the resources must be in place for developers to quickly and easily add such features into their application. The easiest way of facilitating this is be creating, improving, and maintaining accessibility related software development kits for the
major mobile platforms. In an ideal situation, these resources would be made as similar as possible to allow for straightforward code portability.

7. PROPOSED RESEARCH

An example of a future research project would be the implementation of a mobile application that introduces the context-aware features introduced by Kane [6]. Multiple teams will create the same application in parallel across several mobile platforms, noting throughout the process the ease of use of any software development kits utilized. The end results will then be presented to a panel of blind and visually impaired users as part of a user study, and the overall results will be compared against a control application, which does not provide any contextual accessibility.

Once this process has been completed, the results from the study will be compared against the development teams’ notes regarding the ease of implementation. Based on this information, an evaluation of the various developer resources on each platform can be compiled and recommendations can be made to the platforms’ manufacturers.

REFERENCES


