Evaluating Mobile Accessible Applications is a Challenge: Can the Virtual Observer be a Proper Solution?

Benjamin Poppinga[1], Martin Pielot[1], Wilko Heuten[1], Susanne Boll[2]

[1]OFFIS - Institute for Information Technology, Germany
firstname.lastname@offis.de

[2]University of Oldenburg, Germany
susanne.boll@uni-oldenburg.de

Abstract. For developers of a mobile accessible application, understanding the context of use and evaluating the application are the biggest challenges. In this paper we focus on the evaluation phase and report some problems we came across during our studies of the PocketNavigator, an accessible pedestrian navigation aid. Based on the identified issues we argue that it would be better if the observation method is becoming part of the actual mobile application. Consequently we propose and discuss the Virtual Observer, a logging-based observation tool which is tightly integrated into the main application. We conclude our paper with some ideas for future research and identify aspects, which need further discussion in the research community.

Key words: Mobile Accessibility, Multi-Modal Interface, User Observation

1 Introduction

In the year 2011 the presence and density of mobile devices (i.e. smartphones, tablets) have reached an all-time high and the number of sold devices is increasing continuously. With the new era of mobile application distribution platforms (e.g. Apple App Store, Android Market), millions of Apps are available to be downloaded to the mobile device. Disabled people prefer off-the-shelf devices over custom-made hardware [8], even if nowadays touchscreen phones have their limitations [1] and people need to adapt to these [4]. Thus, smartphones and tablets are of high interest for them as well. While the operating system manufacturers aim at providing support for accessible applications (e.g. Apple iPhone[1], Android[2], many of these Apps are still not accessible. Why?

[1] http://www.apple.com/accessibility/iphone/vision.html, last visited September 7, 2011.

[2] http://developer.android.com/intl/fr/guide/practices/design/accessibility.html, last visited September 7, 2011.

For some of these Apps it is impossible to map the intended functionality (e.g. a 3D city guide) to the platform's default user interface elements, which would natively support the development of an accessible application. Instead a custom made user interface is designed, where the developer has to deal with accessibility support himself. However, at this point it's mostly the case that rarely information about the needs of impaired people for this particular application domain exists. At this stage the developer has two options: to neglect the support of accessibility or invent some arbitrary accessibility support without any idea on what's really required. If the developer decides for the latter option it would be a challenge to identify if the invented accessibility features work out.

A typical and often followed design approach is the User-Centred Design (UCD) process. This process is separated into four phases: identify context of use, requirements, design & realisation, and evaluation. We argue that the biggest challenge in creating a mobile accessible application are not the requirements or the design & realisation. More, it is the missing information about the context of use and the missing methodologies to properly evaluate an accessible application.

In this paper we focus on the latter part, i.e. the evaluation of an existing application. The contribution of this paper is two-fold. Firstly, we outline and classify what problems can occur during the observation of such applications. Secondly, we introduce our logging-based observation concept, the Virtual Observer, which aims to overcome these problems. We conclude the paper with some ideas on what the future research in this area should focus on.

2 Problems with the Observation of Mobile Accessibility

In this section we want to outline the problems, an experimenter has to face when he wants to observe how a typical mobile accessible application is used. We are involved in the European project HaptiMap[3], which deals with the design and development of mobile, accessible, location-based applications. Thus, we are able to illustrate these issues on a concrete case.

2.1 The PocketNavigator: An In-Pocket Navigation Aid

The PocketNavigator[4] is one of the HaptiMap demonstrators [5]. As part of the project, several multi-modal interaction techniques have been invented, which aim at making maps and location-based services more accessible. The demonstrator is meant to present these functions to a wider audience. The PocketNavigator is written in Java and runs on Android smart phones, which are capable of Android 2.1 or later. It is available for free in the Android Market and everyone is invited to test and provide us with feedback.

At a first glance, the PocketNavigator looks like a Google Maps clone (see Figure 1). One reason is that we followed a *Design for All* approach. Thus, our

[3] `http://www.haptimap.org/`, last visited September 7, 2011.
[4] `http://www.pocketnavigator.org/`, last visited September 7, 2011.

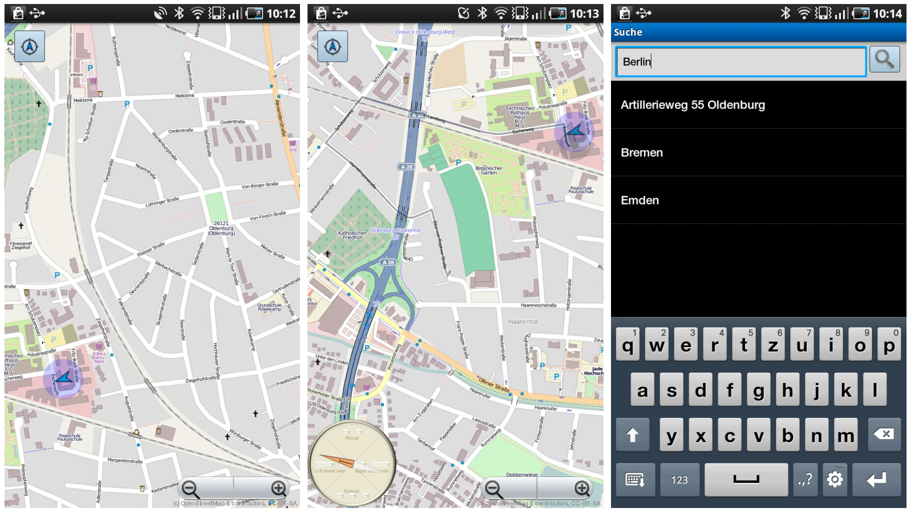

Fig. 1. Almost like Google Maps, the PocketNavigator provides the option to show a map and route to arbitrary destinations. In addition, the integrated Tactile Compass assists impaired people to orient themselves and follow a route.

application can also be used by sighted users, which mostly welcome traditional visual feedback. Another reason is that the PocketNavigator uses tactile and audio feedback to assist visually impaired or blind users. As a consequence, these features are rarely visualized. However, what we call *Tactile Compass* [6] is one of the non-visual core advantages over Google Maps. Once the user entered the desired destination of his/her trip, the *Tactile Compass* provides tactile feedback and "points" to the next waypoint of the route. Two short pulses give evidence, that a user is facing the next waypoint. If the first tactile pulse is longer, the next waypoint is on the left. If the second tactile pulse is longer, the waypoint is on the right. Three pulses indicate that the user is departing the next waypoint and should turn around. The principle of the Tactile Compass is illustrated in Figure 2.

For the development of the PocketNavigator we could rely on initial user studies we conducted within the project. However, the actual testing of our concepts remains difficult. In the following we will briefly state what most important problems we experienced during our studies.

2.2 The Context Problem

One problem in observing the PocketNavigator also happens to other mobile applications. Mobile applications are often subject to different contexts (e.g. place, time, weather, loudness). As a consequence, a user might behave different in various contexts. The context where an application is used in can hardly be captured entirely. However, at the same time in particular for disabled people

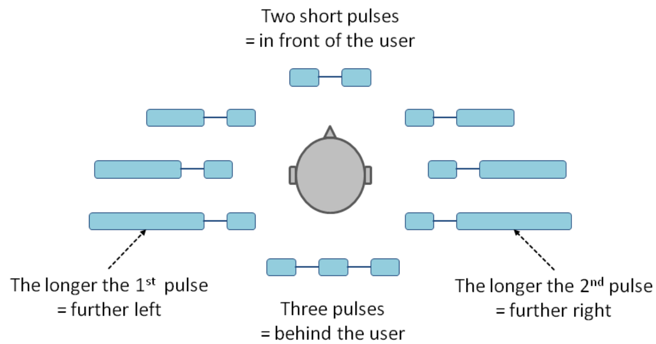

Fig. 2. The Tactile Compass mainly uses two pulses to indicate the direction, where the next waypoint is located at. If the two pulses have an equal length, the waypoint is straight ahead. If the first pulse is longer than the second one, the waypoint is on the left. The waypoint is on the right, if the second pulse is longer than the first one. Three pulses are used to display that a user has to turn around, as the waypoint is located behind.

the context is very important. For experiments, where a researcher is interested in separating cause and effect, this uncertain or incomplete context is a threat for the validity of the results. In practise the argumentation is that all experimental conditions are subject to the same changing context and thus the comparison in an experiment is still fair. However, in particular for non-experimental set-ups the exact impact of the context remains unclear, but can for sure contribute to clarify why an application is used in a certain way.

2.3 The Invisibility of Multi-Modal Feedback

Visual feedback of a mobile application and how a user interacts with it can be captured by e.g. video recording, an eye tracker, or video glasses. However, once an application provides multi-modal feedback (e.g. audio or tactile information) the observation is becoming difficult as these kinds of feedback are invisible. Common observation techniques fail to e.g. capture audio feedback in particular in noisy outdoor environments. However, for the PocketNavigator the Tactile Compass is one of the big advantages, which we want to investigate in depth to improve it in later revisions. However, the tactile feedback in the pocket remains completely undiscovered by existing observation techniques.

2.4 Context-Sensitive Applications

The PocketNavigator provides some of it's features depending on the current context. In particular, the Tactile Compass switches to what we call *scanning mode* if the device is held almost parallel to the ground. This allows a user to use the scan metaphor by pointing the device to a certain direction. Technically this is done with the help of the compass, integrated in nowadays smartphones. Once the device is put into the pocket, the *pocket mode* is enabled. This mode uses the GPS heading instead of the compass to calculate the bearing to the next waypoint, which is then be displayed as a tactile pattern. Both modes are illustrated in Figure 3.

Fig. 3. The PocketNavigator offers two modes: the *pocket mode* (left) and the *scanning mode* (right). If the *scanning mode* is activated, the device uses the compass to detect the pointing direction and the bearing to the next waypoint. If the device is in the pocket, the GPS heading is used instead.

The problem with this context-dependent interaction techniques is again, that common user observation techniques are unable to detect exactly which mode is active at the moment. The problem here is the missing precision of these observation techniques. In example, video recording can be used to determine whether the pocket mode or the scanning mode is enabled. However, the observation result would be quite imprecise as it need to be guessed, if the device is held sufficiently parallel to the ground or not.

3 Solution: Integrate the Observation Method into the Accessible Application

Nevertheless, in the sense of the human centred design the PocketNavigator needs to be evaluated, despite all the mentioned problems. As part of the project we are supposed to help developers of accessible, location-based applications in any matter (designing, realizing and evaluating). Thus, we decided to develop our own observation framework: the Virtual Observer. This framework is based on the well-known logging observation method. Logging describes the process of recording arbitrary information, which are subsequently saved into a file. There rather exist ultimate logging frameworks. Moreover, the to be logged events are identified and implemented by the developers themselves.

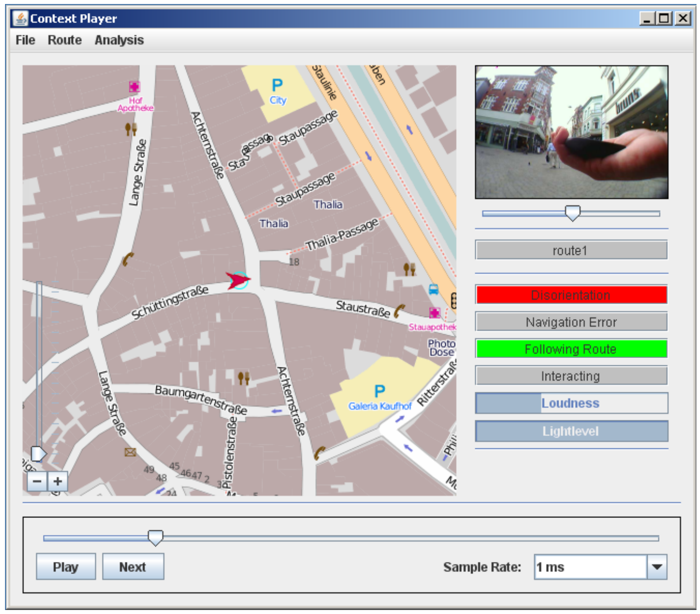

Fig. 4. The ContextPlayer is a desktop application, which helps an experimenter or developer to analyse the data recorded by the VirtualObserver.

The Virtual Observer is designed for Android and in particular location-based mobile applications. Thus, it already comes with a logging function for the most essential features of these applications (e.g. location, speed, GPS inaccuracy, compass heading). In contrast to traditional logging we also investigated the autonomous, intelligent detection and logging of more complex events. Thus, the Virtual Observer is able to detect, whether a study participant is still following the route or if she/he is disoriented at the moment. The Virtual Observer is designed to be integrated in arbitrary Android projects within a short amount of time. For a developers/experimenters convenience we are also developing a tool, named ContextPlayer, which is capable to display and interact with the recorded context information in a convenient way. Beside our logged values the ContextPlayer offers support for displaying images from the Microsoft SenseCam, a tiny camera worn around the neck [3]. A screenshot of the latest ContextPlayer version is shown in Figure 4. We plan to release the Virtual Observer and the ContextPlayer to the research community soon.

So far we applied the Virtual Observer in some of our field studies within the HaptiMap project. Until now our experiences are mixed. The general context values (e.g. location, heading, device posture, etc.) worked very well and accurate. In conjunction with SenseCam images it is easy to remember the context of a certain situation. The Virtual Observer has undergone several design iterations software-wise and thus, can be easily integrated into new applications. On the downside, the SenseCam helped to identify a problem with one of the high-level values, namely the automatic detection of disorientation. Study participants rotated their device physically, which we did not consider properly [7]. As a consequence the accuracy in detecting disorientation events was quite low.

With respect to the three identified problems (see Section 2) for the observation of mobile applications, the Virtual Observer and ContextPlayer help as follows. To address the *context problem* the Virtual Observer provides lots of additional context information. Thus, the subjective context perception through the experimenter can be replaced by the objective sensor measures. That helps to make the separation of cause and effect more accurate and reliable. Further, the Virtual Observer also addresses the *invisibility of multi-modal feedback* and *context-sensitive application* problems. The Virtual Observer records whether tactile feedback is enabled and if the device is held parallel to the ground or not. Thus, it can be accurately determined if either the *scanning mode* or *pocket mode* is active. In addition, unfiltered compass values are logged. Together with the exact user location and the next waypoint, the exact tactile patterns displayed to the user in this situation can be reconstructed.

We think that the Virtual Observer is a helpful tool to assist traditional studies of "in the wild" mobile phone usage. We also think that the Virtual Observer has a huge potential to support studies of in particular accessible applications because of it's precision and closeness to the application. However, at the moment we would not argue that the Virtual Observer can completely replace any gold standard observation techniques, like video recording.

4 Conclusions

Developers of mobile accessible applications need support in two ways: understanding the context of use and the evaluation of their accessible concepts. In this paper we focused on the latter phase of the design process and identified a set of problems. Used primarily outdoors, mobile applications are affected by a changing and dynamic context, which can not be observed sufficiently by traditional observation methods. In addition, many accessible applications use multi-modal (e.g. tactile) feedback, which is almost impossible to observe by traditional observation techniques as well. Finally, the observation of context-sensitive applications lacks precision.

To overcome these limitations we proposed the Virtual Observer, which is a logging-based on-device observation technique. The Virtual Observer is designed to support the evaluation of location-based applications. We argue that this is the conceptual way to go for comprehensive and precise user studies "in the wild".

At the same time, there remain a lot of open questions we need to face. First, it needs to be proved that the Virtual Observer is actually able to provide really valuable insights compared to other observation techniques. Then we are interested in what are the most relevant aspects to observe for developers of accessible mobile applications (e.g. touch behaviour, device posture). How can the Virtual Observer support these most relevant aspects properly? Finally we are interested in the advantages and disadvantages of this approach. One obvious thing we already noticed is that qualitative feedback is impossible to be captured at the moment. However, we are aware of techniques (e.g. Experience Sampling Method [2]), and plan to integrate these soon.

5 Acknowledgements

The authors are grateful to the European Commission which co-funds the IP HaptiMap (FP7-ICT-224675). We like to thank our colleagues for sharing their ideas with us.

References

1. J. a. Benedito, T. Guerreiro, H. Nicolau, and D. Gonçalves. The key role of touch in non-visual mobile interaction. In *Proc. of MobileHCI*, 2010.
2. J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay. Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proc. of MobiSys*, 2007.
3. S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood. Sensecam: A retrospective memory aid. In *Proc. of Ubicomp*, 2006.
4. S. K. Kane, C. Jayant, J. O. Wobbrock, and R. E. Ladner. Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities. In *Proc. of ASSETS*, 2009.
5. M. Pielot, B. Poppinga, and S. Boll. Pocketnavigator: Vibro-tactile waypoint navigation for everyday mobile devices. In *Proc. of MobileHCI*, 2010.
6. M. Pielot, B. Poppinga, W. Heuten, J. Schang, and S. Boll. A tactile compass for eyes-free pedestrian navigation. In *Proc. of INTERACT*, 2011.
7. B. Poppinga and S. Boll. An experimenters third eye: Using the sensecam as ground truth for unsupervised evaluations. In *Proc. of SenseCam Symposium*, 2010.
8. K. Shinohara and J. O. Wobbrock. In the shadow of misperception: assistive technology use and social interactions. In *Proc. of CHI*, 2011.