Supporting Universal Usability of Mobile Software: Touchscreen Usability Meta-test

Vlado Glavinic[1], Sandi Ljubic[2], and Mihael Kukec[3]

[1] Faculty of Electrical Engineering and Computing, University of Zagreb,
Unska 3, HR-10000 Zagreb, Croatia
[2] Faculty of Engineering, University of Rijeka,
Vukovarska 58, HR-51000 Rijeka, Croatia
[3] Polytechnic of Varazdin,
Jurja Krizanica 33, HR-42000 Varazdin, Croatia
vlado.glavinic@fer.hr, sandi.ljubic@riteh.hr,
mihael.kukec@velv.hr

Abstract. Present day mobile applications are becoming increasingly pervasive and complex, involving sophisticated user interfaces and touchscreen-based interaction designs. Their overall acceptance is highly dependent on usability, hence there exists a strong need to make related usability issues an integral part of the mobile software development. In this paper we propose a touchscreen meta-testing model, a set of individual test cases which represents what we believe to be the basic aspects of usability, common to all touchscreen applications. The main goal of the meta-test is to provide relevant feedback on elementary mobile touchscreen interaction, and to use obtained results as important parameters and usability guidelines within the mobile software development process. Along with universal usability support for touchscreen mobile applications in general, this experimental framework can provide some additional benefits, related to different possible ways of both applying meta-test model and using its final outcomes.

Keywords: universal usability, mobile software, touchscreen interaction, usability testing.

1 Introduction

The advance in technical development experienced in the past few years has transformed artifacts like mobile phones from basically simple communication devices to powerful application platforms. This has been made possible by a massive research effort itself being motivated by a vast market demand. In such a framework, both designers and developers have been under continuous pressure by the imperative of rapid application development for such a hungry market, and have worked out many different operating systems and development platforms for mobile applications. On the other hand, it has been also noticed that during the development of mobile software the process of creating an adequate functionality leaves less space to interaction design and usability issues often resulting in their neglect. As problems related to

C. Stephanidis (Ed.): Universal Access in HCI, Part III, HCII 2011, LNCS 6767, pp. 26–35, 2011.

usability are becoming all the more prominent, otherwise forcing users and engineers to take over the design process [1], it became apparent that this situation should be changed by both raising the awareness of the importance of usability and properly addressing usability issues within the early development stages. The specificity of mobile domain confronts developers with particular problems derived from both technology variety and user diversity [2]. These differences represent a barrier to the concept of universal access [3], which can be lowered by sustaining universal usability throughout the process of mobile software development. Our previous research already addressed the importance of usable interaction design in the mobile environment (e.g. [4, 5, 6, 7]), leading to our present focusing on improving both the quality of touchscreen mobile applications and related development process, by providing a usability testing framework as a support for getting both instant answers and appropriate guidelines regarding relevant usability factors.

The paper is structured as follows. In Section 2 we try to determine the most widespread HCI issues and the associated usability problems in nowadays mobile applications based on touchscreen interaction. Following the main idea of reducing these issues, thus making the ground for supporting universal usability of mobile software, in Section 3 we propose the touchscreen usability meta-test model, along with the respective test scenarios which are explained in detail. Also, some layout snapshots of the initial implementation are displayed, illustrating the current state of our efforts. The last section offers a brief recapitulation, including the list of additional expected benefits of the proposed meta-test model, which represents the outline of our future research plan.

2 Common Usability Issues of Touchscreen Mobile Applications: Causes and Consequences

In general, no matter all the benefits it imposes, comprehensive design at programming level of both interface and interaction may not be sufficient for the elimination of all usability problems. Eventual residues can be the result of inherent characteristics of the application itself, what can be closely related with standard approaches to mobile software development, the first of them shown in Figure 1a, where testing and evaluation of touchscreen interaction is performed by using the (physical) mobile device. In this way the developer can obtain very precise and detailed target device feedback, but the main disadvantage of this method is obvious: using a single gadget there is a high probability of making the application device-specific, with user interface (UI), interaction design, and related usability being linked to a distinct group of device models. On the other hand, mobile software development can be based on the utilization of a touch sensitive monitor and emulator platforms (Fig. 1b). Obviously, the development cycle results more efficient, but in such a working setup it is quite hard to evaluate the actual user experience (UX) supported by emulation, especially for different contexts of use. Consequently, possible interaction design limitations can remain undetectable, and eventually overseen (cf. the problem of touchscreen boundary objects).

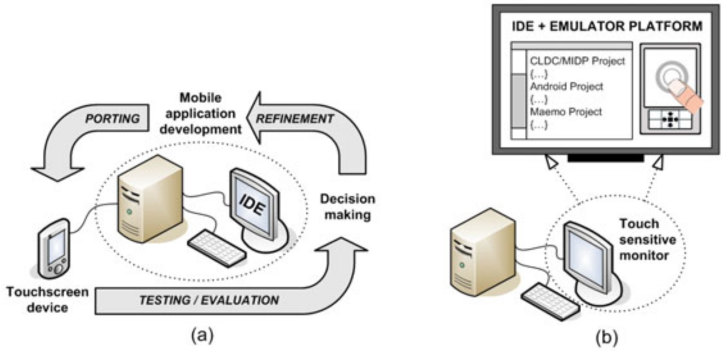

Fig. 1. Standard approaches to the mobile software development process

As a result, although each of the abovementioned approaches to development have their own pros and cons, both can impose severe usability problems. Regardless of development method, complexity, application domain, and target device, several interface elements and related use cases can be extracted as general characteristics of touchscreen mobile interaction. These identified use cases represent what we believe to be the typical usability issues, common to all touchscreen applications. Specifically, they are: optimal size of touchscreen objects, visual search-and-selection, edge-positioning, occlusion, dragging gestures, and efficiency of virtual on-screen keyboards. Descriptions of these issues, along with references to related work, are shown in Table 1.

Table 1. Typical usability issues in mobile touchscreen interaction.

Issue	Standpoint	Related work
Interactive object size	There are no formal recommendations and precise guidelines about quantitative measures for target object size. The problem translates further to portability of object sizes among different mobile devices (with different screen characteristics).	[8], [9]
Visual search and selection	There is a need to efficiently present a number of UI controls/elements (e.g. icons) on a small screen so that the user can easily find a target object among visual distractors.	[10]
Edge-positioning and occlusion	Regardless of the interaction method used (one-thumb, two-thumbs), the problem arises with the occlusion of objects underneath the finger(s) touching the screen. Moreover, interface objects that are located along the edge of the screen of the mobile device can be difficult to reach.	[11], [12], [13]
Dragging gestures	Interaction techniques similar to drag-and-drop can be applied to a touchscreen to improve tasks such as target selection, zooming, panning, and scrolling. However, efficiency and accuracy of performing the corresponding gesture should be thoroughly analyzed.	[14], [15]
Virtual keyboard usage	Text and/or numeric entry via virtual on-screen keyboards imposes the problem of activating a sequence of closely placed UI controls. The core of this usability issue (erroneous input) lies in the controls' size and their mutual distance (i.e. keyboard layout).	[16], [17]

Problems related to HCI aspects of mobile applications highlighted in this section may sometimes be regarded as only minor annoyances of no particular importance. However, we deeply believe that this is not true, as the cumulative effect of irritating the user while she/he is trying to get through the interface can surely have a negative effect on usability. Therefore, we are trying to acquire as much as possible information about usability of fundamental GUI actions that correspond to abovementioned issues. We believe that this is possible by making use of a specially designed touchscreen usability meta-test, a model of which we propose and describe in detail in the next section.

3 Touchscreen Usability Meta-test

The main goal of the touchscreen usability meta-test is to provide relevant (both quantitative and qualitative) feedback on elementary touchscreen actions, and to use the obtained results to make usability issues an integral part of the development process, thus combining the advantages of both the developer-centered and the user-centered design (UCD).

Following UCD guidelines, granting enough attention to abilities and skills of the end user (throughout a "user understanding" phase) should precede activities like defining interaction (by writing use cases) and designing the user interface proper (by prototyping and evaluation) [18]. Generally, appropriate usability testing involves measuring typical users' performance on carefully prepared tasks that are typical of those for which the system is being designed, as well as carrying out user satisfaction surveys (through questionnaires and interviews) for obtaining users' subjective opinions [19]. Usability tests are generally application-specific, where user performance is usually measured in terms of the number of errors and time to complete application oriented tasks. While performing these tasks, user interaction with the application software is somehow logged (automatically by the software itself, or observationally by the evaluators), and the collected data is afterward used to calculate performance times, identify errors, and help explain the nature of users' interaction behavior.

However, the main problem with traditional usability testing (when it is not purposely put aside and finally ignored) is its dependency on target system/application, as well as on the usual practice of conducting usability testing after the deployment phase (when it is usually too late for "usability debugging"). In order to achieve the goal of providing support for universal usability of mobile software in general, relevant feedback on touchscreen interaction elements usage is needed in the preliminary stage of development already. However, this feedback should not be associated with some particular mobile application, but should instead represent applicable data based on common usability issues as noted in the previous section. Consequently, our touchscreen usability meta-test model (Fig. 2) consists of five test scenarios which include the following: (i) object sizes (in)convenient for the direct touch technique, (ii) search for a given interface object, (iii) selection of boundary located interface objects, (iv) drag-and-drop gesture feasibility, and (v) keystroking using a soft (emulated, touch) keypad.

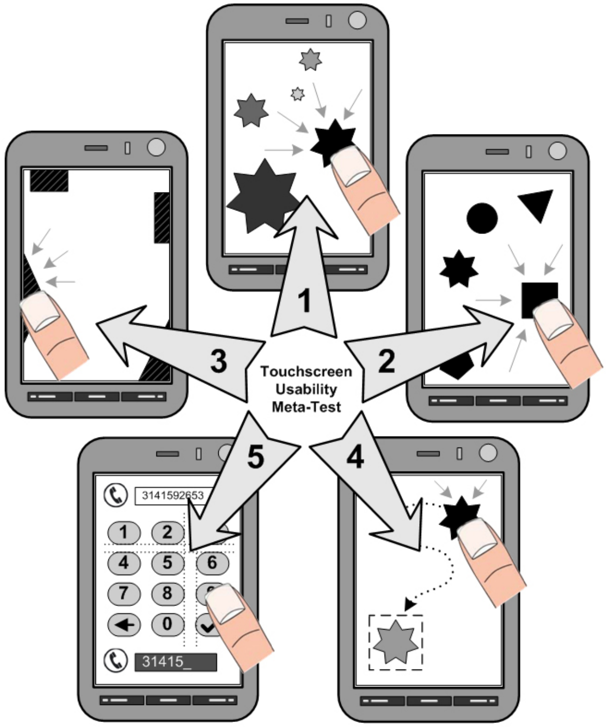

Fig. 2. The meta-test, consisting of five test scenarios, deals with fundamental aspects of mobile touchscreen interaction: appropriate size of interactive object (1 – *Target Size*), search for a distinct object among distractor items (2 – *Object Search*), selection of edge-positioned objects (3 – *Boundary Objects*), dragging gestures (4 – *Drag&Drop*), and sequential input via a virtual on-screen keyboard (5 – *Soft Keystroking*).

3.1 Meta-test Scenarios

The first test case, *Target Size* (Fig. 2: arrow 1), addresses the size of an object visible on the target device display. Herein, the object can represent a range of UI elements that are responsive to the direct touch technique (e.g. button, icon, textbox, scrollbar slider). The purpose of this test scenario is to determine the most suitable object size that is applicable to devices having different display resolutions and running different application types. Proper object size should minimize wrong selections, void touch-screen actions and the overall direct touch inaccuracy. This test is based on direct touch repetitive use in targeting (tapping) a displayed object. The application software displays one object at a time, at a randomly chosen position, while the user's goal is to be as much as possible efficient and accurate when hitting objects with her/his thumb or forefinger. With appropriate logging procedures, all of the users' interaction data can be available for the subsequent statistical analysis, and particularly task completion time and user targeting precision.

The second test scenario, denoted *Object Search* (Fig. 2: arrow 2), is very similar to the *Target Size* test, but includes on-display *distractor* objects. Once again, the user is confronted with a task sequence of targeting a given graphical target object, but in the same time additional items are also drawn on the application canvas. Accordingly, the user goal within the given task consists in searching the application visual space for a particular object and in its actual selection using direct touch. This test addresses the search problem found in real mobile applications with touchscreen interaction; e.g. activating the preferred option from a menu toolbar, selecting the particular picture from a thumbnail list, clicking the proper hyperlink in a mobile Web browser, or typing the right letter on a touch keypad. With the test outcomes thus obtained, we expect to get a better insight into the correlation between the type and number of distraction objects and users' performance while working with rich mobile interfaces. Additionally, elements of the *Target Size* test can be included in this scenario, hence both target and distraction objects can be randomly resized throughout the respective task sequences. Furthermore, the number of distractors can be altered during the test, making the final results dataset even more valuable.

Finding out how, and to what extent, the edge-positioning of GUI elements can implicate touchscreen interaction constraints is the main goal of the *Boundary Objects* test (Fig. 2: arrow 3). In this test setting the user is supposed to repetitively hit targets that are randomly positioned near the application canvas border, with the interaction speed and accuracy being constantly measured and logged. Two different test outcome datasets can be obtained in this scenario, since the user can achieve the given task goals using either the thumb (when the hand holding the device is the same one used to target boundary objects), or her/his forefinger (when one hand is holding the device and the other one is performing the interface actions). Boundary objects can additionally be rendered with different sizes and shapes, hence particular interaction tasks could be quite demanding, e.g. hitting relatively small objects in the far corner of the device display. Although interface design procedures should generally avoid the introduction of interactive boundary objects in mobile applications, it would be very useful to find out their optimal size and the best possible distance from the display border.

The next test case, *Drag&Drop* (Fig. 2: arrow 4), covers selecting and moving the target object by dragging it to a different canvas location. The objective of this test is to get a better insight into the usability of the touchscreen based drag-and-drop interaction, by calculating both error ratio and average time for a typical task completion. Such user task consists of (i) selecting a moveable object (the token) by direct touch, (ii) dragging it to a new given location (target area) following the given route, and (iii) eventually releasing the token by lifting the finger from the device display. By performing these steps, various dragging gestures can be tested and analyzed. Throughout the user task sequence, the size of both token and target area could be (randomly) changed, while the dragging task will be much more challenging with these sizes coming to a similar value. Since the exact positions of token and target areas are also randomly generated, thus creating the possibility for their placing near the application border, we can assert that the *Drag&Drop* test has certain properties inherited from both *Target Size* and *Boundary Objects* test scenarios. The results obtained in this part of the meta-test should provide information on the cost-benefit ratio and on the feasibility of introducing dragging gestures into a mobile application interaction design.

Finally, the last test scenario within the meta-test model, *Soft Keystroking* (Fig. 2: arrow 5), studies the specific touchscreen problem of activating a sequence of closely placed UI control commands. Two most common instances of this problem are: (i) using a soft (on-screen emulated) keypad for text input (e.g. composing an e-mail, or SMS text inputting), and (ii) using a soft numeric keypad for entering a particular series of numbers (e.g. dialing a phone number). Herein, the core of the usability issue lies in the interactive controls' size and their mutual distance. Therefore, the related test case is based on the emulation of the numeric keypad and consists of tasks requiring the input of a given phone number. When performing this test, the user's objective is to try to be as much as possible quick and precise while interacting with the soft keypad. Interaction speed can be measured through task duration; on the other hand the precision attribute encompasses several values stored in the interaction log: the number of incorrect digits in the entered sequence, the number of "backspace" keystrokes, and the number of void touchscreen actions. The purpose of the test scenario is to find out optimal dimension and spacing values for a particular set of interactive controls, which are to be used in a frequent activation mode.

3.2 Implementation

The described model of touchscreen usability meta-test is intended to be run in an environment with real test users on real mobile devices, while each test user participates in every test scenario.

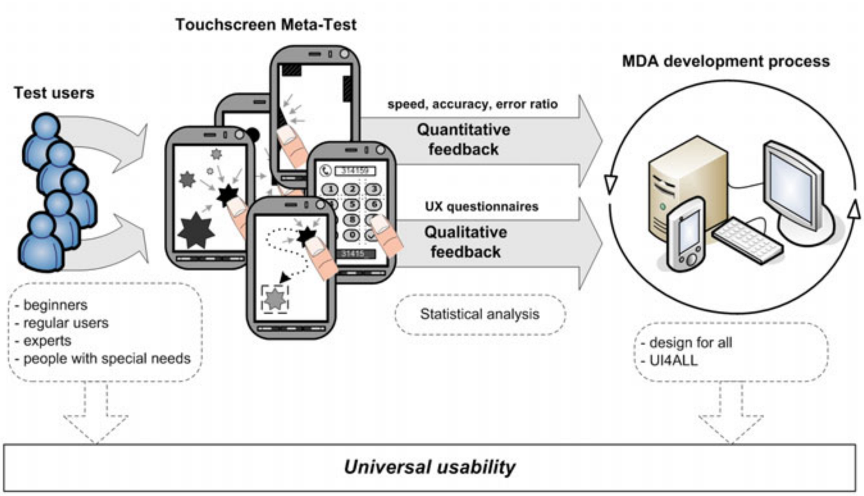

Fig. 3. Applying the proposed meta-test model. Involving different user categories in the testing process promises a possibility for optimal UI design – a step forward in pursuing universal usability.

While performing the corresponding tasks, users' interaction speed and accuracy has to be constantly measured and logged (by the application software itself), thus providing extensive datasets for statistical analysis (calculating performance times, identifying errors, explaining the nature of users' interaction behavior). We believe that the outcomes obtained from this experimental framework can represent a valuable input and a strong starting point for mobile software development, in the same time increasing the level of universal usability (Fig. 3).

In the current stage of development, we are refining our data logging system along with the module for statistical data analysis (ANOVA included). While the first version of proposed meta-test model was implemented using *Java2ME* technology for CLDC/MIDP architectures [20] (so as to be run as *Java MIDlet* on every suitable Java-enabled touchscreen mobile device), a new version is developed using *Android SDK* for increasing group of device models based on this popular mobile operating system [21]. Some screenshots of the current *Android* implementation are shown in Figure 4.

Fig. 4. From left to right: *Object Search*, *Drag&Drop*, and *Soft Keystroking* test instances, running in an *Android SDK* emulator with HTC skin.

4 Conclusion and Future Work

Through the five test scenarios pertaining to the presented meta-test model, fundamental issues of touchscreen interaction can be tested and subsequently analyzed. These typical challenges in mobile HCI can be found in almost every contemporary mobile application, hence benefits in applying the meta-test are expected regardless of the domain of the software being developed. The meta-test final outcomes could provide a synergetic effect as: (i) developers should early enough gain valuable knowledge about devices, users, and touchscreen usability, (ii) mobile software development enriched with support for universal usability should result in better applications with a "highly practical" interaction and (iii) target users should be more efficient and more satisfied when interacting with a particular application, while the overall user experience would be increased to a higher level.

Our future research directions follow from the expected additional contributions of the meta-test model, and will specifically include: (i) better insight into the correlation between real devices and emulator-based interaction effects, (ii) analysis of usability discrepancy between different display technologies, (iii) support for mobile software UI rapid prototyping, (iv) utilization of game-based usability testing concept, and (v) involvement of a wide population base in the testing process. These points of interest can be studied according to different ways of both applying the meta-test model and using its final outcomes (see Table 2).

Table 2. Additional expected benefits of the proposed touchscreen usability meta-test.

Benefit	Approach to applying the meta-test / Using test outcomes
Better insight into the correlation between real devices and emulator-based interaction effects and usability issues	Performing the usability meta-test experiment in controlled environment with two setups: (i) using real touchscreen mobile devices, and (ii) interacting with device emulators on a touch sensitive monitor.
Detailed overview of discrepancy in usability aspects arising from different display technologies	Involving a broad spectrum of different touchscreen mobile device models within meta-test, with capacitive and resistive displays equally numbered.
Support for rapid prototyping and predictive evaluation of mobile software UI	Using meta-test outcomes as formal parameters for predictive evaluation of UI prototypes (implemented with mobile software for on-device UI layout design).
Proof of concept for game-based usability testing	Adopting and presenting usability meta-test cases in the form of mobile game with scoring system. In such an experiment, we can presume better users' involvement in the testing process, as well as more trustworthy results.
Widest possible test-users base	Involving a wide population base in the testing process by making use of popular web services. This includes (i) publishing the meta-test model online as a free mobile game application, and (ii) using social networks for "game" promotion.

Acknowledgments. This paper describes the results of research being carried out within the project 036-0361994-1995 *Universal Middleware Platform for e-Learning Systems*, as well as within the program 036-1994 *Intelligent Support to Omnipresence of e-Learning Systems*, both funded by the Ministry of Science, Education and Sports of the Republic of Croatia.

References

1. Scott, K.M.: Is Usability Obsolete? ACM Interactions XVI(3), 6–11 (2009)
2. Shneiderman, B.: Universal Usability: Pushing Human-Computer Interaction Research to Empower Every Citizen. Communications ACM 43, 85–91 (2000)
3. Stephanidis, C.: Editorial. International Journal – Universal Access in the Information Society 1, 1–3 (2001)
4. Glavinic, V., Ljubic, S., Kukec, M.: A Holistic Approach to Enhance Universal Usability in m-Learning. In: Mauri, J.L., Narcis, C., Chen, K.C., Popescu, M. (eds.) Proc. 2nd Int'l Conf. Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2008), pp. 305–310. IEEE Computer Society, Los Alamitos (2008)

5. Glavinic, V., Kukec, M., Ljubic, S.: Digital design mobile virtual laboratory implementation: A pragmatic approach. In: Stephanidis, C. (ed.) UAHCI 2009. LNCS, vol. 5614, pp. 489–498. Springer, Heidelberg (2009)
6. Glavinic, V., Ljubic, S., Kukec, M.: Transformable Menu Component for Mobile Device Applications: Working with both Adaptive and Adaptable User Interfaces. International Journal of Interactive Mobile Technologies (IJIM) 2(3), 22–27 (2008)
7. Glavinic, V., Ljubic, S., Kukec, M.: On efficiency of adaptation algorithms for mobile interfaces navigation. In: Stephanidis, C. (ed.) UAHCI 2009. LNCS, vol. 5615, pp. 307–316. Springer, Heidelberg (2009)
8. Lee, S., Zhai, S.: The Performance of Touch Screen Soft Buttons. In: Proc. 27th Int'l Conf. Human Factors in Computing Systems (CHI 2009), pp. 309–318. ACM Press, New York (2009)
9. Parhi, P., Karlson, A.K., Bederson, B.B.: Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. In: Proc. 8th Conf. Human-computer interaction with mobile devices and services (MobileHCI 2006), pp. 203–210. ACM Press, New York (2006)
10. Repokari, L., Saarela, T., Kurki, I.: Visual Search on a Mobile Phone Display. In: Proc. Research Conf. South African Institute of Computer Scientists and Information Technologists on Enablement through Technology (SAICSIT 2002), pp. 253–253. SAICSIT (2002)
11. Karlson, A.K., Bederson, B.B.: One–Handed Touchscreen Input for Legacy Applications. In: Proc. 26th SIGCHI Conf. Human Factors in Computing Systems (CHI 2008), pp. 1399–1408. ACM Press, New York (2008)
12. Perry, K.B., Hourcade, J.P.: Evaluating One Handed Thumb Tapping on Mobile Touchscreen Devices. In: Proc. Graphics Interface (GI 2008). ACM Int'l Conf. Proc. Series, vol. 322, pp. 57–64. Canadian Information Processing Society, Toronto (2008)
13. Roudaut, A., Hout, S., Lecolinet, E.: TapTap and MagStick: Improving One-Handed Target Acquisition on Small Touch-screens. In: Proc. Working Conf. Advanced Visual Interfaces (AVI 2008), pp. 146–153. ACM Press, New York (2008)
14. Potter, R.L., Weldon, L.J., Shneiderman, B.: Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. In: Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 1988), pp. 27–32. ACM Press, New York (1988)
15. Albinsson, P., Zhai, S.: High Precision Touch Screen Interaction. In: Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 2003), pp. 105–112. ACM Press, New York (2003)
16. Zhai, S., Hunter, M., Barton, A.S.: Performance Optimization of Virtual Keyboards. Human-Computer Interaction 17, 89–129 (2002)
17. Zhai, S., Kristensson, P.O.: Interlaced QWERTY: Accommodating Ease of Visual Search and Input Flexibility in Shape Writing. In: Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI 2008), pp. 593–596. ACM Press, New York (2008)
18. Detweiler, M.: Managing UCD Within Agile Projects. ACM Interactions XIV (3), 40–42 (2007)
19. Sharp, H., Rogers, Y., Preece, J.: Interaction Design: Beyond Human-Computer Interaction, 2nd edn. John Wiley & Sons Ltd., Chichester (2007)
20. Java ME Technology, `http://www.oracle.com/technetwork/java/javame/tech/index.html`
21. What Is Android?, `http://developer.android.com/guide/basics/what-is-android.html`