Digital design Mobile Virtual Laboratory implementation: a pragmatic approach

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

[1] Faculty of Electrical Engineering and Computing, University of Zagreb,
Unska 3, HR-10000 Zagreb, Croatia
vlado.glavinic@fer.hr

[2] College of Applied Sciences,
Jurja Krizanica 33, HR-42000 Varazdin, Croatia
mihael.kukec@velv.hr

[3] Faculty of Engineering, University of Rijeka,
Vukovarska 58, HR-51000 Rijeka, Croatia
sandi.ljubic@riteh.hr

Abstract. The omnipresence of m-devices, and especially those of the cellular phones type, certainly makes the basis for the introduction and use of m-learning systems, but their implementation heavily depends on the area to be covered hence showing a different degree of complexity. This especially holds for the area of digital design, where there is the need for handling logic schemes which includes both displaying and modifying them in addition to the usual imaging of text and graphics. In this paper we discuss essential HCI issues, which are related to the implementation of a Mobile Virtual Laboratory for digital logic design. We make use of a pragmatic approach by blending a number of interaction methods known from other application fields hopefully providing a holistic effect in the process of learning and teaching in the mobile.

Keywords: digital logic design, m-devices, m-learning, touch sensitive screen, virtual laboratories

1 Introduction

Present technological developments prompted by an ever increasing demand of mobile services on the telecom market have brought to an equally intensive development of mobile devices (m-devices) [1], such that the generational gap between m-devices and desktop computers is been constantly reduced. As the technological progress of both computers and the Internet has favored the expansion of e-learning systems, the development and proliferation of m-devices enables, although with a slower pace, the advancement of m-learning systems. The fundamental hindrance for a faster and smoother introduction of m-learning systems lies in limitations imposed by m-devices themselves [2], which mostly arise from their reduced dimensions.

The advantage of the m-learning concept has led us to the inception of the Mobile Virtual Laboratory (MVL) that would support the process of teaching and learning on the move for the area of digital design [3]. According to the domain specificity, the major requirements to be put in front of the respective MDA (Mobile Device Application) implementation should include the following: (i) display of digital schemes on reduced area screens typical of m-devices, (ii) successful navigation within digital schemes which are not necessarily displayed on an m-device screen as a whole, (iii) easy creation/modification of digital schemes possibly using new technological enhancements (as e.g. touch sensitive screens), and (iv) simultaneously running simulations coupled to the digital circuit being developed on-screen. As the latter subject is beyond the scope of this paper, we focus on usability [4] and HCI problems concerning display, managing, and navigation within digital circuitry on the mobile device screen (Fig. 1). In solving the abovementioned issues we make a pragmatic use of already known techniques, which we consider to have a holistic impact to the solution of the digital design MVL problem.

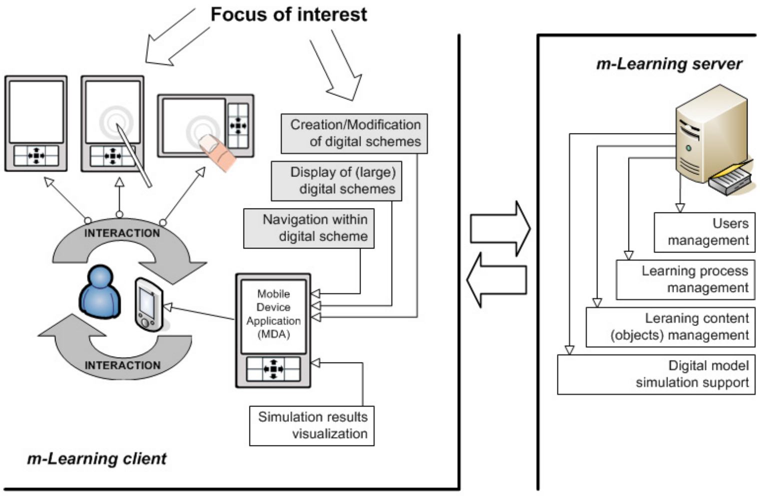
Focus of interest
Creation/Modification of digital schemes
Display of (large) digital schemes
Navigation within digital scheme
INTERACTION
INTERACTION
Mobile Device Application (MDA)
Simulation results visualization
m-Learning client
m-Learning server
Users management
Learning process management
Leraning content (objects) management
Digital model simulation support

Fig. 1. Focus of interest within a digital design MVL system

2 Related Work on m-Learning and Our Motivation

The potentiality of m-learning overall success lies in the omnipresence of mobile devices in classrooms and university halls. In fact, it has been noticed that “there is no doubt that one of the ways of motivating people such as the typical 20 or 30 year old is by using Information and Communication Technology” [5]. A research conducted at University of Birmingham on a sample of seventeen MSc students [6] with the goal of “investigating whether students would find a handheld computer useful for supporting their learning” has shown that although technology has been useful to students, “mobile learning organizers used in this study did not ‘revolutionize’ student’s styles or patterns of learning”. However, the authors acknowledged this study as just the first attempt at designing integrated learning organizers.

Mobile Learning Engine (MLE) [7] represents a multimedia mobile device application which supports many features, some of them being display of formatted text with integrated images and hyperlinks, audio and video playback, single/multiple choice quizzes and graphical marking questions. Strong multimedia capabilities have been developed to partially overcome limitations of screen size, as authors believe that replacing continuous text with different kinds of multimedia brings better results in the learning process. MLE has been tested for the medical knowledge domain but it can easily be utilized for other domain environments, too. Deficiencies of mobile devices have been reduced with the usage of Mobile Interaction Learning Objects (MILOs) and strong emphasis on multimedia, resulting with new benefits in transforming mobile devices into mobile *learning* devices.

Cellular phones and/or PDA based MDAs developed for performing virtual experiments by remote controlling real hardware is an interesting concept presented in [8]. Concerning HCI issues, while commenting virtual laboratories in general, the authors found that “action of wiring and rewiring offers the true challenge” while working with digital components.

Apparently, both usability challenges and HCI issues represent a common problem in the abovementioned work. Some of them can be reduced through technological advancement of mobile devices (more computing power and storage capability, better screen resolution, communication channels of larger bandwidth), but others require much more effort to be even partially solved. Firstly, it is obvious that the learning content should be presented in the most appropriate way for mobile users. One of the possible solutions is to create small units of domain knowledge containing as much as possible multimedia content adapted to target mobile device. If an m-Learning system is to be used in a special knowledge domain requiring practice and training (such as a laboratory environment), it should then include special MDAs tailored to support interactive examples and simulations from this particular area. Such interactive techniques can be very helpful in the learning process, because “a number of studies indicated that interactive, dynamic computer simulations have been successfully applied in promoting exploratory learning of new concepts as well as in changing existing mental representations” [9].

3 Design and Implementation of an MVL

This section discusses in more detail our implementation of a mobile virtual laboratory, the component of our m-Learning system which is to be used as a “support MDA” for teaching digital design, and emphasizing both user interface design and HCI issues.

Mobile applications are becoming more versatile, with many options and functions being lately offered to the user. To be able to efficiently use all of the offered options, the MDA user should be able to find the required option quickly and without much effort. Some of the mobile applications and related functions are self explanatory and intuitive (e.g. the picture browser), and as such they do not have much options. On the other hand, a digital design MVL application represents a complex software unit with many options and modes of behavior. Such an application must provide the user with a method of swiftly browsing through its options and eventually of finding the desired one. The described functionality is provided by our transformable menu component (TMC) for mobile device applications [10]. TMC development is motivated by our goal to improve mobile user interaction with menu navigation and provide our MDA with a rich GUI component capable of managing a large number of functions and options which the user can apply during her/his work without having to invest much effort in the interaction. The TMC has the same form as PC desktop application menus, as well as suitable size and shape with respect to limitations of mobile device displays. The abovementioned motivation goals (i.e. HCI issues) are encompassed by using both adaptable and adaptive procedures, thus making the menu component both personalized and easy to use. Recent improvements to the TMC, which are currently under development, include support to touchscreen events and actions.

3.1 Creation and Modification of Digital Schemes

Creation and modification of digital schemes in any type of simulator or even CAD system are a painstaking task. Applications for simulating digital logic running on desktop computers, having at their disposal all of the respective luxuries, can be very complex to use, and encompass numerous menus and toolbars, such that creating a logic scheme of an even moderate size can be a time consuming task. The user has to place a number of components, move them to different locations, align them manually or by means of some tools, connect (or virtually wire) them, and then iterate the cycle. The procedure for creating digital logic schemes in our MDA is the same as in any other simulation or CAD tool but with a reduced number of actions the user has to take. Interconnection routes between components are automatically routed using the A* search algorithm, eliminating the need of user intervention here [11, 12].

Having at disposal minimal input capabilities, we had to reduce the number of actions the user has to take to activate certain functions. Different input methods have been implemented and combined so far, some of them specially tuned for specific task [13]. From the standpoint of graphical imaging, TMC and other interface components are designed to be partially, and in some cases even fully transparent, in order to limit occlusion of background objects as less as possible. A feature that additionally limits the occlusion of background object is the ability to relocate TMC to four different locations [10]. As TMC uses icons, we expect it to be more usable through implementation of icons usage guidelines [14]: textual information is replaced wherever possible, with icons, using icons familiar in the domain of digital schematics (mostly logic symbols), see Fig. 2.

On the other hand, we note that many contemporary mobile devices are already equipped with touchscreens, a technology which undoubtedly is going to improve the input capabilities of mobile device but yet remains to be thoroughly investigated [15]. Motivated by the existing research done in the area of touchscreen usage we have implemented the MDA interface so as to be able to concurrently use interaction events from both keypads and touchscreens of mobile devices. In any moment, without need to explicitly change mode of operation from keypad to touchscreen and vice versa, both input methods can be used. Using a touchscreen fewer actions are needed to obtain the same result than using the keypad, which requires navigating to an option before activating it. Within the touchscreen environment all of the on-screen options are displayed just one tap distant from their activation, in order to avoid navigation to the item. Conversely, when using the keypad, both adaptive and adaptable algorithms can be used to decrease the number of actions to be carried out [10].

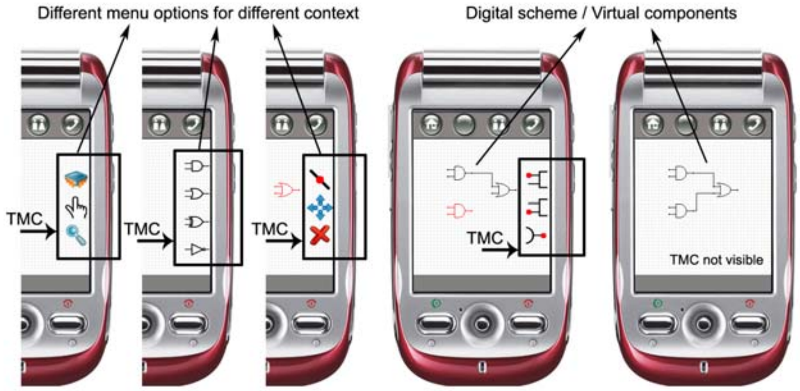
Different menu options for different context
Digital scheme / Virtual components
TMC
TMC
TMC
TMC
TMC not visible

Fig. 2. TMC is a semitransparent GUI component imaged only when required, thus minimizing occlusion of background objects

When using a touchscreen we assume that the user is tapping the screen using her/his fingers and not the stylus (if present), hence for TMC we are using the Direct Touch technique. Because of the relatively large size of fingers in comparison to the screen size, screen targets must have adequate sizes for the user to be able to tap on them with the required degree of precision. According to [16], mobile device touchscreen targets should have a size of at least 9.2 mm, which in pixels varies considerably from one device to another. Pixel size is important in the design phase, when user interface components are modeled. Because of big variations between different devices, accommodation must be provided by implementing adaptability and adaptivity in order to create interface components with adequate sizes for a particular touchscreen [17]. TMC uses raster icons which also need to be adapted to different screen sizes to conform to the 9.2 mm rule. Rather than resizing raster icons, we have taken a much simpler approach: the margin belonging to the icon is resized, thus improving tapping accuracy on small screens with higher resolution. The change in size of the icon margin can be calculated by the MDA (adaptive behavior), while subsequently the margin size can be changed by a suitable user action (adaptable behavior).

Iconic interaction [18] is not used only by the TMC but also by the MDA's virtual breadboard. *Breadboards* are tools used to assemble electronic circuitry during the experimental phase, resulting in prototypes being built without soldering the components. This enables their subsequent disassembly and reassembly as required in the design process. A *virtual breadboard* uses icons (in fact logic symbols) to represent real digital components, as shown in Fig. 2 and Fig. 3. The respective icon (logic symbol) here stands for a representation of the underneath data structures carrying semantics and related component data. Icons/logic symbols pertinent to a touchscreen environment should also adhere to the 9.2 mm guideline for target size [16], what is achieved by the virtual breadboard zooming feature. The zooming factor can be automatically calculated by the MDA (adaptive behavior) and subsequently possibly changed by the user (adaptable behavior).

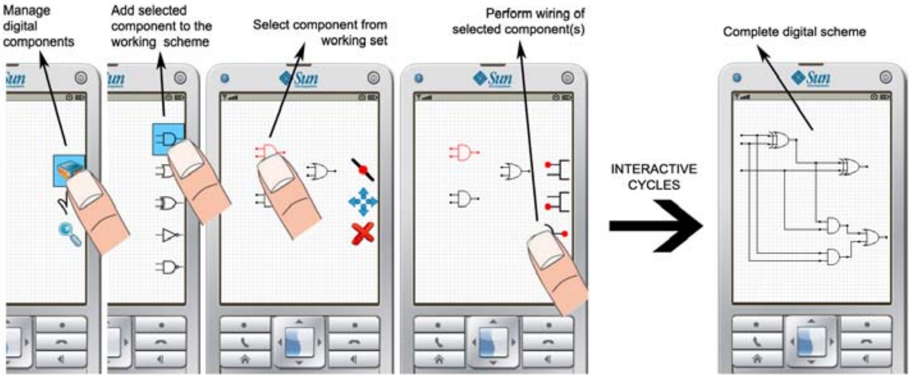
Manage digital components
Add selected component to the working scheme
Select component from working set
Perform wiring of selected component(s)
INTERACTIVE CYCLES
Complete digital scheme

Fig. 3. From left to right: creation of a simple digital logic scheme. Using the touchscreen the user points to the first icon of a vertical TMC hence opening a submenu, then points again to the logic gate to be placed, and eventually places it by again tapping the screen. The A* algorithm is subsequently used to calculate the routes and connect the symbols.

The latter two paragraphs illustrate how to accommodate the user interface to touchscreen actions by resizing the interface widgets, however some of them due to their nature cannot be magnified enough like e.g. interconnection lines in the MVL setting. Users planning to create interconnections between two virtual wires must select a point in a pixel-wide wire where the connection is to be made. Using a finger a user would probably tap correctly on the line, but the accurate tap position will be somewhere in the range of the abovementioned 9.2 mm. Zooming the line (in both dimensions) as any other graphical element will make it wider on the screen, resulting with a smaller length imaged on-screen with the surrounding interconnected symbols possibly out of screen, thus loosing the schematic context. The precise selection of a target having dimensions of just a few pixels is very difficult when using Direct Touch on touchscreens, hence warranting a completely new approach. For instance, techniques like TapTap and MagStick are experimentally proven to successfully tackle with the "selection of targets with the thumb on small tactile screens: screen accessibility, visual occlusion and accuracy" [19]. TapTap uses one tap to generally determine the region of interest and a second one to precisely select the target in the previously selected region of interest which is zoomed in the middle of the screen. However, selection of points on a line by zooming it is not applicable in our case, and consequently we cannot use the TapTap. Conversely, because of its "magneticity" feature, MagStick can be very successfully applied for selection of connection points on lines, since it supports sticking to a line but allow also movements of the selection point, as opposite to sticking and holding to the closest selected point on the line.

3.2 Display and Navigation Issues Associated with Small Screens

The display of digital schemes on reduced area screens presents problems related to the impossibility of having imaged the whole circuit, since digital schemes can very easily be large and complex, exceeding the imaging capabilities of m-devices (and especially of cellular phones). In this context we rely on some known techniques, which are used quite successfully for displaying both (regular) Web pages [20] and geographic maps [21]. Both from a conceptual and an implementation view, regarding digital schemes as vector graphics enables an easier display scaling (i.e. zooming in/out), see Fig. 4, a characteristic that is considered important for such contents. Additionally, simple vector graphics manipulations also support an easier adaptation to various m-devices with differing screen resolutions; this also involves the possibility of a subsequent implementation of fish-eye techniques for highlighting the focus of manipulation.

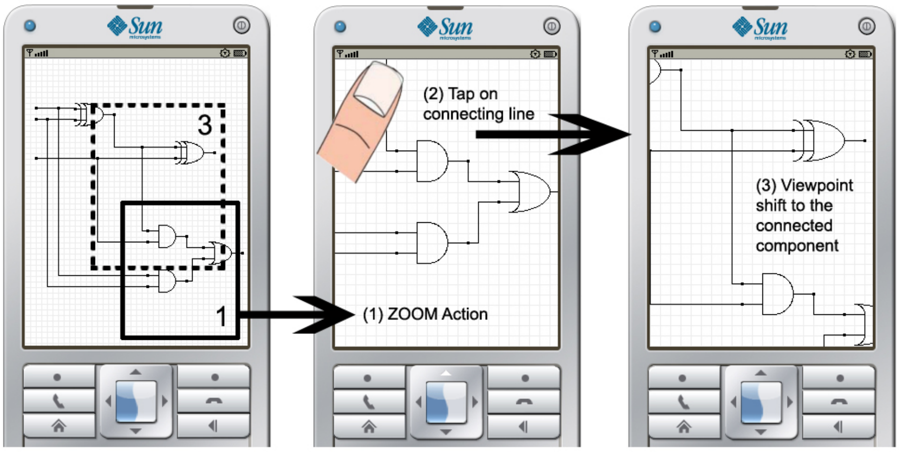
Sun
3
1
(1) ZOOM Action
(2) Tap on connecting line
(3) Viewpoint shift to the connected component

Fig. 4. Navigation to off-screen elements using interconnection lines

Implementing an effective navigation is a general issue in m-device screens [22, 23]; in digital design (and in our MVL implementation) it should get a more considerate attention because of the specific content to be displayed. Here the analogy with navigating (digital) geographic maps cannot be assumed to hold fully. Namely, spatial relationships are the core of the information to be relayed to the user of maps, while this is not the case in digital schemes where the information lies in interconnecting of digital components. Therefore, navigation to digital components (represented by logic symbols) relies either on classic window panning or on using information about components' internal relationships; this latter is especially helpful when trying to reach digital components not imaged on-screen.

Performing manipulations on digital schemes (creation and/or modification) is certainly not an easy task, and is quite demanding for an MDA because of limited input capabilities provided by an m-device, which are presently constrained on arrow keys navigation. Hence the rationale for using a pointing device based interaction style, using a suitable touch sensitive screen, what comes especially useful when handling unconnected components. Such an approach could also be interpreted as syntactic sugar for a digital circuitry visual language.

The virtual breadboard implements only basic methods for navigating to off-screen objects, which includes scrolling, panning and zooming (the latter is supported by the usage of vector graphics). Some more advanced methods, on the other hand, include techniques like Halo and HOP. Halo is a technique interesting “for pointing users to the presence of off-screen objects” [24] by drawing circles around them while HOP is an “interaction technique that enables quick access to off-screen objects” [21]. A comparative evaluation of Halo and simple direction arrows pointing hasn't found significant differences in performing simple spatial tasks [25]. This fact together with the prospect of occluding much of the screen by Halo arcs (visible parts of circles drawn around off-screen object) in the case of larger number of components prompted us to implement a simpler technique much closer to the direction arrows, which uses information on component internal relationships to make the user aware of possibly existent off-screen objects. Interconnections leaving the screen are a clear indicator of existing off-screen objects; hence the MDA and its virtual breadboard do not require any special graphical element for showing off-screen objects. Interconnections are also used to navigate to the next connected element: tapping on the outgoing line near the screen boundary will shift the viewpoint to the next connected element, see Fig. 4.

4 Conclusion

While developing a particular mobile device application for MVL, a strong ambition to take advantage of improved capabilities of present day mobile devices occurs. Consequently, we endeavor to make this application as possible powerful, graphically rich and highly interactive. In the same time, in the implementation process we are confronted with the usability challenges and HCI issues, hence our efforts are focused on the quality of mobile user interaction. Having in mind special requirements in the digital design domain, we have considered and evaluated a number of well-known interaction methods, trying to pragmatically adapt them in our MDA, while at the same time keeping in mind both features and type of contents our application handles.

Our work targets the implementation of a suitable MDA with (i) iconic interaction which enables easy creation and modification of digital logic schemes on mobile device screens, and (ii) the possibility to use both device keypad and touch sensitive screen (if available), without special need to explicitly change interaction mode. We deem that such an approach, which consists of using a visual language (digital schemes) and bimodal interaction (arrow keys and fingers), accomplishes a didactically innovative and effective m-learning concept for teaching digital design.

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. The GSM Association (GSMA), Online at: http://www.gsmworld.com/europe_observatory/
2. Shudong, W., Higgins, M.: Limitations of Mobile Phone Learning. In: Proc. IEEE Int'l Workshop on Wireless and Mobile Technologies in Education (WMTE'05), pp.179--181. IEEE Computer Society, Washington DC – USA (2005)
3. Glavinic, V., Ljubic, S., Kukec, M.: Mobile Virtual Laboratory: Learning Digital Design. In: Luzar, S.V., Hljuz, D.V. (eds) Proc. 29th Int'l Conf. Information Technology Interfaces (ITI'07), pp.325--310. SRCE University of Zagreb, Zagreb – Croatia (2007)
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'08), pp. 305--310. IEEE Computer Society, Los Alamitos – USA (2008)
5. Attewell, J., Gustafsson, M.: Mobile Communications Technology for Young Adult Learning and Skills Development (m-Learning). In: Milrad, M., Hoppe, H.U., Kinshuk (eds) Proc. IEEE Int'l Workshop on Wireless and Mobile Technologies in Education (WMTE'02), pp. 158--160. IEEE Computer Society, Washington DC – USA (2002)
6. Corlett, D., Sharples, M., Chan T., Bull, S.: A Mobile Learning Organiser for University Students. In: Proc. 2nd IEEE Int'l Workshop on Wireless and Mobile Technologies in Education (WMTE'04), pp. 35. IEEE Computer Society, Washington DC – USA (2004)
7. Holzinger, A., Nischelwitzer, A., Meisenberger, M.: Mobile Phones as a Challenge for m-Learning: Examples for Mobile Interaction Learning Objects (MILOs). In: Proc. 3rd IEEE Int'l Conf. Pervasive Computing and Communications Workshops (PerCom'05), pp. 307--311. IEEE Computer Society, Washington DC – USA (2005)
8. Alkouz, A., Al-Zoubi, A.Y., Otair, M.: J2ME-Based Mobile Virtual Laboratory for Engineering Education. International Journal of Interactive Mobile Technologies (iJIM) 2(2), 5--10 (2008)
9. Holzinger, A., Kickmeier-Rust, M.D., Wassertheurer, S., Hessinger, M.: Learning Performance with Interactive Simulations in Medical Education: Lessons Learned from Results of Learning Complex Physiological Models with the HAEMOdynamics SIMulator. Computers & Education 52(2), 292--301 (2008)
10. 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)
11. Xing, Z., Kao, R.: A Minimum Cost Path Search Algorithm through Tile Obstacles. In: Proc. 2001 Int'l Symposium on Physical Design (ISPD'01), pp. 192--197. ACM, New York NY – USA (2001)
12. Johann, M., Reis, R.: Net by Net Routing with a New Path Search Algorithm. In: Proc. 13th Symposium on Integrated Circuits and Systems Design (SBCCI'00), pp. 144. IEEE Computer Society, Washington DC – USA (2000)
13. Karlson, A.K., Bederson, B.B.: One–handed Touchscreen Input for Legacy Applications. In: Proc. 26th SIGCHI Conf. Human Factors in Computing Systems (CHI'08), pp. 1399--1408. ACM, New York NY – USA (2008)

14. Holzinger, A., Errath, M.: Mobile Computer Web-Application Design in Medicine: Research Based Guidelines. Int'l Journal Universal Access in Information Society 6(1), 31--41 (2007)
15. Perry, K.B., Hourcade, J.P.: Evaluating one handed thumb tapping on mobile touchscreen devices. In: Proc. Graphics Interface (GI'08), ACM Int'l Conf. Proc. Series vol. 322, pp. 57--64. Canadian Information Processing Society, Toronto – Canada (2008)
16. 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'06), pp. 203--210. ACM, New York NY – USA (2006)
17. Glavinic, V., Granic, A.: HCI Research for e-Learning: Adaptability and Adaptivity to Support Better User Interaction. In: USAB'08. LNCS, vol. 5298, pp. 359--376. Springer, Heidelberg (2008)
18. Beaudouin-Lafon, M.: User interface support for the integration of software tools: an iconic model of interaction. ACM SIGSOFT Software Engineering Notes 13(5), 143--152 (1988)
19. 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'08), pp. 146--153. ACM, New York NY – USA (2008)
20. Baluja, S.: Browsing on small screens: recasting web-page segmentation into an efficient machine learning framework. In: Proc. 15th Int'l Conf. World Wide Web (WWW'06), pp. 33--42. ACM, New York NY – USA (2006)
21. Irani, P., Gutwin, C., Yang, X.D.: Improving selection of off-screen targets with hopping. In: Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI'06), pp. 299--308. ACM, New York NY – USA (2006)
22. Mehra, S., Werkhoven, P., Worring, M.: Navigating on handheld displays: Dynamic versus static peephole navigation. ACM Transactions on Computer-Human Interaction (TOCHI) 13(4), 448--457 (2006)
23. Roto, V., Popescu, A., Koivisto, A., Vartiainen, E.: Minimap: a web page visualization method for mobile phones. In: Proc. SIGCHI Conf. Human Factors in Computing Systems (CHI'06), pp. 35--44. ACM, New York NY – USA (2006)
24. Baudisch, P., Rosenholtz, R.: Halo: a technique for visualizing off-screen objects. In: Proc. SIGCHI Conf. on Human Factors in Computing Systems (CHI'03), pp. 481--488. ACM, New York NY – USA (2003)
25. Burigat, S., Chittaro, L., Gabrielli, S.: Visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In: Proc. 8th Conf. Human-Computer Interaction with Mobile Devices and Services (MobileHCI'06), pp. 239--246. ACM, New York NY – USA (2006)