

Mobile computer Web-application design in medicine: some research based guidelines

Andreas Holzinger · Maximilian Errath

Published online: 20 April 2007
© Springer-Verlag 2007

Abstract Designing Web-applications is considerably different for mobile computers (handhelds, Personal Digital Assistants) than for desktop computers. The screen size and system resources are more limited and end-users interact differently. Consequently, detecting handheld-browsers on the server side and delivering pages optimized for a small client form factor is inevitable. The authors discuss their experiences during the design and development of an application for medical research, which was designed for both mobile and personal desktop computers. The investigations presented in this paper highlight some ways in which Web content can be adapted to make it more accessible to mobile computing users. As a result, the authors summarize their experiences in design guidelines and provide an overview of those factors which have to be taken into consideration when designing software for mobile computers. “*The old computing is about what computers can do, the new computing is about what **people** can do*” (Leonardo’s laptop: human needs and the new computing technologies, MIT Press, 2002).

Keywords Information interfaces and representation · Interface design · Mobile computing · Life and medical sciences · Internet applications

Introduction

Mobile computing as a challenge for universal access in the information society

Mobile Computing is an umbrella term and describes any technology that enables people to access information and which supports them in daily workflows independent of location. Actually, it is remarkable that many of the mobile computing research results and breakthroughs came from the Human–Computer Interaction (HCI) field. However, there is still room for a great deal of progress in this extremely successful and worthwhile area of endeavor. The phenomenal growth of mobile computing, which has not been accompanied by an equivalent increase in the understanding of end-users for the complexities of this subject, has caused an increase in the need for research in mobile computing and a parallel campaign to increase the awareness and proficiency of the end-users.

Consequently, the area of mobile computing is a good example of the *New Computing* in the sense indicated by Ben Shneiderman: the new computing technologies must enable end-users to accomplish their tasks and to relax, enjoy, and explore [54].

Why mobile computing in health care and medicine?

A representative example of mobile computers is the handheld computer, which is often referred to as a Personal Digital Assistant (PDA). This device presents a number of challenges in HCI and Usability Engineering (UE), including the tension between the appropriate user interface design, the device and the social context of the device’s use [43].

A. Holzinger (✉) · M. Errath
Institute for Medical Informatics, Statistics and Documentation
(IMI), Research Unit HCI4MED, Medical University of Graz,
Auenbruggerplatz 2/V, 8036 Graz, Austria
e-mail: andreas.holzinger@meduni-graz.at

Handheld computers are often used together with desktop computers [42] and support nomadic and ubiquitous computing [30]. This is of tremendous interest for physicians and health care professionals.

Medical doctors and nurses work in an environment which requires high mobility. Within their daily routine, their sphere of activity alters frequently between wards, outpatient clinics, diagnostic and therapeutic departments and operating theatres. Although access to stationary clinical workstations is provided in the hospital, their locations do not always coincide with the user's current workplace. In order to fulfill a high health service standard, the medical staff has an extensive demand for information at a number of locations—which actually only mobile computers can supply [52]. For example, up-to-the-minute electronic patient record information is not always available at the bedside [6, 68]. New orders or diagnostic results noted during rounds must be transcribed to the electronic patient records via a clinical workstation at a later time—whereas a mobile computer enables direct access [41, 31].

However, although mobile computers have been available for a relatively long time [9] in hospitals, different studies [4] show that health care professionals are reluctant to use poorly designed mobile systems, as the work at the point of patient care is very time-bound and hectic.

To design and develop mobile systems with high acceptance, it is essential to obtain empirical insight into the work practice and context in which the mobile system will be used. Consequently, mobile devices are only useful when design and software validation aspects have been taken into account [24].

Usability and usefulness

While it is difficult to make devices with small displays usable, there is also the fundamental challenge of making them useful. In this context, usefulness means that the application is beneficial and can help the end-users to achieve a particular goal, whereas good usability means that the application is easy and pleasant to use.

To achieve both goals, particularly for mobile devices, it is strongly recommended applying a User-Centered Design (UCD) approach [36]. UCD evolved in the field of HCI and was first articulated as such in User-Centered System Design [47] and focuses strongly on the intended end-users [62].

Objectives and structure of this paper

Applications for mobile computers are often the *mobile part* of a larger non-mobile Web application. In this case, the mobile solution must be particularly designed for a

quick and short interaction *on route*. This is of particular importance for the highly mobile group of end-users addressed in this paper, i.e., medical doctors and health care professionals within a hospital. The developed application was intended to raise the acceptance of the main system by the availability of a mobile solution—in the sense of ubiquitous usability.

This paper aims to highlight some aspects of how Web content can be adapted to make it more accessible to mobile handheld computer users. The overall aim of the project was to provide automatic adaptations of content, thus enabling users to gain access to as wide a range of the material as possible.

Sect. 2 provides some details about the problem to be solved and the resulting application. Sect. 3 describes design and development issues and Sect. 4 presents and discusses some of the issues we derived in the form of condensed guidelines.

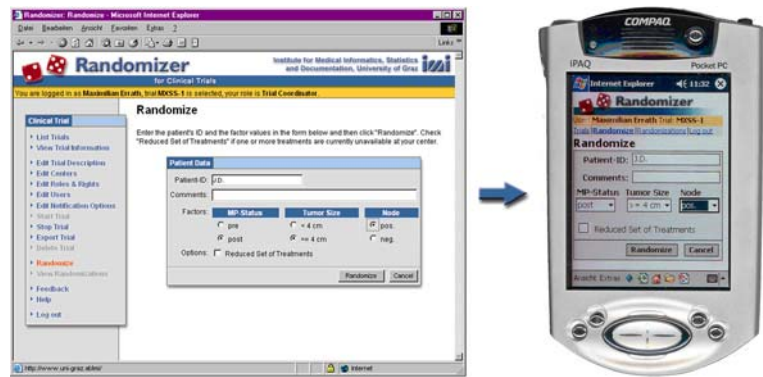
Example application: the randomizer for clinical trials

Randomization in clinical trials

In medical research, the randomized clinical trial is the gold standard for assessing treatment effects [8]. For example, in a trial of a standard treatment versus a new treatment, random allocation of patients to treatment groups ensures that each patient has the same chance of receiving either the new or the standard treatment. There are two main reasons why randomization is used. First, randomization leads to treatment groups that are random samples of the population, and thus statistical methods based on probability theory can be used. Second, it minimizes bias, thus leading to groups that are generally comparable, thereby ensuring that observed differences between the treatment groups are due to differences in the treatments alone. Random allocation does not guarantee that the groups will be identical apart from the treatment given, but it ensures that differences between them are due to chance alone. For more information on randomization and good practices in clinical trials see, for example, the ICH Guidelines (www.ich.org) or the CONSORT statement (<http://www.consort-statement.org>).

In most clinical trials, randomization is performed using pre-printed randomization lists, sealed envelopes and telephone-based services (for example using automated response technology). In large and multi-center clinical trials (i.e., trials where multiple centers are contributing patients), a commercial third party—the *trial coordinating center*—is responsible for patient registration, data collection and randomization.

Fig. 1 The randomization software offers patient randomization and trial management features using a standard Web-interface designed for desktops. Core functionality (i.e., randomization) is also available via Web pages optimized for PDAs



Web-based solution

Performing randomization and trial data collection via the Internet using Web-based applications has several advantages over the traditional telephone-based services, particularly in multi-center trials.

Using the Web reduces communication delays and expenses, provides a worldwide 24-h service, reduces transcription errors, supports better auditing due to comprehensive logging of each transaction and saves the researchers time. Finally, yet most importantly, a Web-based solution also facilitates communication between the trial users (investigators, statisticians, trial coordinators, etc.).

For example, the latest version of the trial handbook or a directory of trial users' email addresses can be stored on the central website used for randomization and downloaded by trial users on demand.

From Web-based to Web-based mobile solution

A Web-based solution for randomization in multi-center clinical trials was implemented, which has been called the *Randomizer*. The main Web-application (desktop version) was so successful that it is now commercially available (see www.randomizer.at). This software provides a self-serve, easy to use and secure 24 h-a-day randomization service that runs exclusively on the Internet. Investigators randomize patients with only a few clicks, by completing an on-screen form with patient details, and are immediately notified of the treatment allocation. Using a flexible role-based access control, trial coordinators can nominate investigators, biometricians, monitors and pharmacists. For multi-center trials, user management can be delegated to participating centers. Trial coordinators can open or stop patient recruitment at any time. They can also choose to be notified via e-mail on events such as randomization, emergency un-blinding and others. Many methods of dynamic randomization, such as permuted blocks, biased

coin, urn randomization, and minimization are available. As an additional service for biometricians, the *Randomizer* provides a powerful simulation tool, which can be used to generate static randomization lists, to test a trial design or to study the implemented randomization methods. All transactions are logged. The trial's audit trail and the list of randomizations can be downloaded and analyzed at any time by the trial monitor. Of course, the network traffic between the Web-browser and the *Randomizer* is encrypted using SSL (Secure Sockets Layer) with strong (128 bit) encryption.

To make the available core functionality of this software—i.e., randomization of new patients—in places without an infrastructure to support computers and networks, using handhelds equipped with cell phone (GSM) or WLAN modules was considered. This offers a rapid solution for physicians who work with existing health care practice habits and flows with the health care professionals' daily routine (see Fig. 1).

Design, development, experiments and evaluation

User-centered development

During the development of the *Randomizer*, the *User-Centered Development* method was applied. Although User Interface Design for mobile computers and usability evaluation of software for mobile computers has been an emerging area of research for some time, most available work has reported about techniques for evaluating the usability of mobile computers within laboratory settings [29], whereas interaction of end-users must be studied in a *real life setting* [19, 20].

It is also generally known that software developers rarely use Usability Engineering Methods (UEM) on software development projects [23]. This includes basic UE techniques such as early focus on the end-user, empirical measurement and iterative development of prototypes [45]. Few

projects to date have adopted a fully integrated User-Centered Development approach in one strategic shift [19, 20].

What is important is an expansion from UCD to User Centered Development [35, 21]. Every user-centered approach involves knowing who the end-users will be [32, 1] including their capabilities and limits, needs and expectations, their goals and the tasks required to achieve those goals, as well as the physical and social environments and the context in which end-users must achieve those goals [55]. One of the most critical aspects is to hold the motivation of a user at a satisfactory level [16]. This involves processes of participatory design, end-user testing and iterative design [13, 53]. Also with consideration of the International Standard 13407 [65], prototyping is a vital element during development [21, 22].

Prototyping

When designing an interface, designers are confronted with a dilemma: on the one hand, it is not possible to evaluate an interface until it is built; on the other hand, after it has been built, changes are difficult and expensive. One solution to surmounting this dilemma is prototyping. This means producing cost effective interactive versions of an interface design. Most of all, together with UE Methods, this enables to bring the end-user into the development process *at a very early stage*. Some general advantages of prototyping include reduction of development time and consequently a reduction of costs; facilitation of end-user involvement from the beginning, thus supporting user-centered development; facilitation of system implementation, since the end-users know what to expect, consequently resulting in higher end-user satisfaction. However, prototyping also presents some disadvantages: it might possibly lead to insufficient analysis, resulting in incomplete documentation; the end-users might expect the performance of the final system to be “identical” to the prototype; developers can become reluctant to modify their prototypes.

Working prototypes vary according to either the breadth or depth of features implemented. Basically, they cut down on either the number of features, or the depth of functionality of features. Consequently, Vertical, Horizontal and Scenario prototypes can be distinguished. *Vertical prototypes* concentrate on “in-depth” functionality for a few selected features, *Horizontal prototypes* concentrate on full interface features, but no underlying functionality, and *Scenario prototypes* include only the functionality of specific scenarios or paths through the interface.

Generally, prototyping is a rapid technique to include *direct feedback* from the end-users into design. Paper based prototypes avoid both time and effort, which is definitely required to develop coded user interfaces.

A further advantage is that only simple tools are necessary (paper sheets, stickers, scissors etc.) [51, 21].

Paper mockups

Paper mockups do not need to cover all aspects of functionality and technological possibility. However, they must provide a general overview about particular functionalities and convey the right information [15].

The term *paper mock-up* commonly means “to prototype screen designs and dialog elements on paper”. This is accepted as unquestionably the cheapest method. Wisely used, it can be also a very efficient method and in combination with other methods (e.g., thinking aloud), it can be very effective. Generally, with ordinary office materials, each interface element (e.g., dialog boxes, messages, buttons, menus etc.) can be sketched and hand-printed on pieces of paper and index cards (e.g., post-it notes).

One of the main advantages is that paper mock-ups encourage the end-users to come up with more suggestions, since the mock-ups are obviously easy to change. This leads to a quick and easy creation of alternatives; however, this assumes that the developers have autonomy in the creation of the user interface, which is not always the case. Our experience in the development of Randomizer confirmed that end-users were more likely to change the paper mock-up than—later during the project—the high-fidelity prototype. This is also evident from previous experience: end-users generally take paper prototypes very seriously, and they are more likely to express usability problems [21, 44, 33].

Some advantages of paper mock-ups include:

- First sketches allow early and immediate usability feedback;
- Because the prototype is not too detailed, designers and end-users can concentrate on abstract dialog concepts and not on technological details; and
- Prototypes are cheap to produce (sketches, easy to use, fast turn-around, can encourage team design, early usability feedback with throwaway designs results in a maximum feedback for minimum effort).

Some disadvantages of paper mock-ups include:

- It is relatively difficult to capture interface behavior (see, e.g., the Wizard of Oz Technique, [37]);
- In combination with usability methods (see chapter 3.4) it needs more time than theoretically predicted (see, e.g. [21]); and
- There is still low confidence in this method amongst software developers, due to limited acquaintance with these methods during engineering education [20] (Fig. 2).

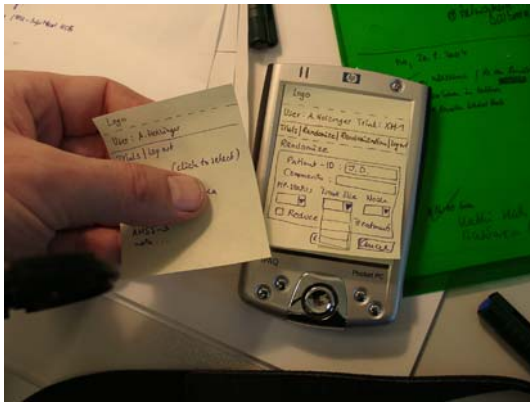


Fig. 2 Paper mock-ups proved to be efficient within the development of the Randomizer

Experimental design and methods

During the design phase of the desktop version of the Randomizer, especially during the analysis of the workflows of the medical personnel, the need for a mobile application soon emerged. However, since the requirements for a mobile solution are definitely different, a plan articulated in three (iterative) phases was developed in order to both follow an end-user-centered development and to generally gain insight into the challenges of developing mobile applications:

- Phase 1: Investigations and observations of real-life workflows on how the end-users would and could be best supported by a mobile application;
- Phase 2: User-Centered Development on the basis of paper-mock ups, careful studying certain parts of workflows of the target end-user group; and
- Phase 3: Experiments and evaluations, these are necessary to gain insight into the differences between a desktop and a mobile application.

During phase 1, it was very important to carefully understand and specify the context of use, the characteristics of the end-users, as well as the organizational and physical environment where the end-users *usually* and *actually* will use their application. Due to the fact that the end-users' tasks characterize the context in which the mobile solution is used, a task was defined as the way in which a specific goal is attained. Consequently, the following issues were carefully analyzed:

- Characteristics of the end-users (this includes previous knowledge, skill, experience, expertise, level of education, training, physical attributes, habits and capabilities);
- Specific tasks that the end-users need to perform (typical scenarios, frequency and duration of performance etc.); and

- Environment in which the end-users will finally use the mobile application (this included technical network infrastructure, workplace, work practices, environmental disturbances and attitudes towards computers in work).

In phase 2, in order to gain insight into the issues as described above and to carry on with the development, thinking aloud studies were carried out in real-life settings. The following equipment was used:

- Hi8 video camera and recorder on tripod;
- mirror;
- high quality microphone (fixed on tripod, acoustically decoupled from camera to avoid interferences); and
- headphones (monitoring the sound is essential).

In parallel, several issues were also investigated through questionnaires and by orally interviewing the end-users (see section 4 for some experiences and lessons learned).

On the basis of the gained insights, the prototype was implemented following a UCD approach.

Finally, during phase 3, specific tasks were tested, with the involvement of 12 carefully selected end-users (see Sect. 3.5), by comparing the task performance between the desktop application and the mobile application (see Chap. 3.6). For these experiments in real-life we used a stopwatch.

Participants

Previous experiments [66, 46] showed that more than 80% of all usability problems can be found by examining around three to five end-users. Following this guideline, 12 end-users were involved, consisting of 4 novices, 4 intermediates and 4 experts or semi-experts users. Testing was conducted not in a usability lab, but in a real-life setting, which is not easy with mobile devices, but was considered to be absolutely necessary to fully understand the end-users' context as described in Sect. 3.4.

Task performance time

Our task experiments focused especially on:

- number and nature of errors per task;
- number of errors per unit of time;
- number of navigation units per task necessary*);

and most of all on

- time to complete a task (time to perform task, ttpt).

*) An exact definition of any single navigation unit is difficult, since this is dependent on the task or goal. It is generally understood to mean the actions undertaken by the end-user to achieve one step towards the goal.

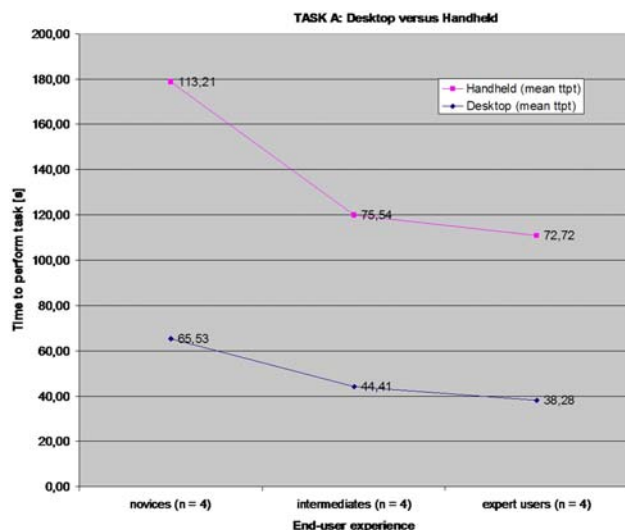


Fig. 3 Time to perform task: desktop versus handheld

The time required to perform a specific task is still the most important factor to measure the value of an application in the work environment [60, 59, 61]. A specific task on handheld devices requires more time to complete the task than the same task on desktop computers (Fig. 3).

Hewlett Packard iPAQs were used for the experiments. These devices come with a 240×320 pixel display. However, with Microsoft's Pocket Internet Explorer (Pocket IE), the user interface (scroll bars, address line, title bar, etc.) further limits the visible content area to 229×255 pixels.

There was a strong correlation between *time* to complete the task and *problems* in locating functions and navigation. Consequently, pages that need horizontal scrolling should be avoided. Most of the end-users overlook the fact that there is more to view in the horizontal direction. Scrolling can be most simply reduced by strictly focusing on the content task.

Due to the limited functionality of the PDA application—in contrast to the Randomizer's native Web interface—navigation paths are short and limited to two levels in depth. Therefore, navigation should not be an issue for the increased time to complete task observed during the experiments.

Lessons learned in developing Web-applications for handhelds

Based on the conducted experiments (Sect. 3.4), it can be concluded that there is a considerable difference between developing Web-applications for handhelds and for desktop

computers [27, 18]. Different factors are to be considered within the following problem areas:

- Information output (chunking, textual information, graphics, Icons, colors, errors, help);
- Information input (tapping, circling, typing);
- Navigation (interaction style, menus, scrolling, browsing, shortcuts); and
- Connectivity (handling connections, synchronizing information, security)—factors which will not be discussed within this paper.

The experience acquired during the development of the Randomizer is presented in the form of condensed guidelines. Although general guidelines are available (see, e.g., the NIH Web design guidelines, available online at http://www.usability.gov/guidelines/guidelines_notice.html), these guidelines are, to the best of the authors' knowledge, the first to specifically address mobile computing in medicine.

Information output on small screens

Chunking information

Humans are able to expand the capacity of their working memory by chunking information together [39]. A chunk is defined ambiguously and subjectively depending on the previous knowledge of the end-users [58].

For example, four letters can be four (separate) chunks (PXHF) or just one chunk (ICON) – in this example the latter is much easier to recognize, because it is a recognizable word instead of a randomly ordered number of characters. As a consequence, a combination of letters which form a *pronounceable* syllable will be more easily remembered than one that is not pronounceable.

Based on the conducted research, the following guidelines are proposed:

- Display the most relevant information first, and allow the users to choose which information they want to view next;
- Highlight relationships between information: proper chunking of information is essential on small screens;
- Present the information in a way which is familiar to the end-users—know your end-users, e.g., present information in the way the end-users need it; base your design on particular experiments within the specific context of use;
- Locate common information in common screen locations;
- Provide frequently used information in a location most obvious to the end-users; and
- Place general items before of specific items; if no order exists, alphabetize.

Textual information

The impact of screen size on the ability to understand textual information was already examined considerably long before the era of the Web [40]. Looking at such studies, it is interesting to note that research on screen reading showed that the reading speed on large screens was approximately 30% less than reading on paper [40].

As computers became more and more widespread, readability on large screens increased, but it is still lower than that of reading on paper [69]. Progress in readability on tiny screens is not expected to follow the same pattern, and low readability will still remain a central issue for devices with limited screen size [69].

An interesting work by Duchnicky & Kolars (1983) [7] considered the effect of window height and line widths on reading. Experiments showed that the full width display was read 25% faster than the screen which was 1/3 the width. However, the impact of varying the display height was much less remarkable.

According to the authors' experiences, the following guidelines should be taken into consideration:

- Keep the amount of text within the application generally to a minimum;
- Consider that the end-users typically make use of the mobile application away from their office; consequently text should be designed for conditions where reading text is difficult;
- Use sans serif fonts (Tahoma, Verdana) and avoid small fonts, be consistent in the use of font, font size, styles, etc.;
- Present the most important content first (Inverted Pyramid Style);
- Do not right justify text (this causes interruption of eye movement and an obstruction to reading);
- Provide good contrast of the text against the background (black fonts on white background is simple and is most often the best; see Sect. 4.1.4 for more details);
- Present text in small interfaces carefully; avoid abbreviations and truncations; and
- Design interfaces that display long text paragraphs in a way which is more suitable for small screens, e.g., by using adaptive Rapid Serial Visual Presentation, which adapts the presentation speed to the characteristics of the text instead of keeping it fixed [48].

Icons

Icons are pictorial representations and suggest a certain meaning. There are many different guidelines for building

icon-based interfaces (examples include [2, 38, 25]). Icon design can be limited by implementation constraints and is sometimes restricted by icons that are in existence already, thereby lacking originality in some sense [10].

Icon design evolved from the concept of signs according to Peirce [50, 18, 17]. Peirce viewed a sign (S) as a product of a three-way interaction between a so-called *representamen* (which represents), the sign's *object* (which is represented) and its so-called *interpretant* (which is the process of interpretation): $S = S(R, O, I)$.

This implies that for User Interface Design, icons alone are meaningless without a particular context and in Human–Computer Interaction the following relationship for icons results [12, 38]: Icon + context + viewer > meaning.

The following design guidelines result from these theoretical considerations in combination with the authors' practical experiences:

- Replace textual information through representative and familiar icons whenever possible;
- Provide familiar icons only, do not force target end-users to learn new icons; if no familiar icon can be provided, use text instead of a new icon;
- Be aware that the meaning of icons is culture dependent, consequently test icons with target end-users; and
- Provide good contrast between the icon and the background, and ensure that icons are distinguishable from each other; shape, form and structure of an icon must always be unique.

Colors

The proper use of color is very important [56]. However, most of the research on color predates the Web [49, 34]. It is very interesting that approximately hundred years after Isaac Newton (1643–1727), Johann Wolfgang von Goethe (1749–1832) examined the problems of color and although his *Theory of Colors* intended to attain “*a more complete unity of physical knowledge*” by including all branches of Natural Science, Goethe approached the subject primarily to gain some knowledge of colors “*from the point of view of art*” [11]. We cannot emphasize enough that color is one of the most efficient techniques for transferring visual information from the computer to the human user. The impact of color, as well as the problems of color perception, has been investigated for a long time [14]. There are general guidelines available, e.g. [57, 67], but unfortunately most guidelines that currently exist are often not based on experimental research [64].

Based on the results of previous experiments on the proper use of color, the following can be recommended:

- Use colors to help the end-users to orient and navigate; never use a color without underlying intention;
- Use a consistent color scheme within the whole application; use colors to group information;
- Make sure that all information conveyed through color is also readable without color;
- Avoid under all circumstances the use of too many colors together; avoid whenever possible the combination of red and green;
- Avoid saturated colors (they cause visual confusion and stimulate the eye enormously);
- Be careful when using blue: thin blue lines are visible but usually difficult to perceive, black and white always provide the best resolution for fine details and small shapes;
- Be careful when using colors of similar brightness or contrast, because they might be difficult to distinguish; and
- Always test the colors used with your target end-users because colors generally have a strong culturally dependent meaning;

Errors

Humans should always be expected to err, and applications must be designed to prevent and tolerate user errors. This is especially true for mobile devices because end-users are far less familiar with mobile devices than with standard devices (see also Fig. 3). Therefore:

- provide an easy “back” button – every end-user action must be reversible;
- accept common misspellings;
- require a confirmation if the requested action causes a change;
- allow expert users to disable confirmation requests by the application settings; and
- provide meaningful error messages in plain language.

Help

Although many end-users feel comfortable having help functions available on request, during the development of Randomizer it was decided that the application must be useable without any help function; consequently no online help was implemented.

Information input

Entering information into handhelds is possible without typing text. There are also other possibilities including voice recognition – which can be interesting in clinical

applications – or separate enhancements (camera, keyboards, keypads, etc.).

However, textual information is still the most important. Common methods are typing via virtual keyboards and selection methods including tapping and circling. With respect to information input, the following guidelines arise:

- Prefer general selection controls instead of text entry (the error rate resulting from selecting is generally lower than the error rate resulting from entering data); consider that the users might use the handheld away from their offices;
- Thoroughly understand the users’ task in order to avoid long or unnecessary text entry or information input at all; text entry is easily left unused when it seems unnecessary to the end-users;
- Allow the application to “learn” from the user input; repeatedly used text can be used to enhance input;
- Prioritize information entry, especially when converting a desktop application to a handheld application;
- Support copy-and-paste and cut-and-paste functionality;
- Allow end-users to increase and decrease numbers by using the navigation key;
- Provide default values whenever possible; use previous entered data as default value; a thorough task analysis is necessary in order to know exactly how the end-users work;
- Provide default values or at least zeroes for numerical data to indicate the data format; and
- Enable the users to exit the application quickly without losing any information; however, the application should remember what the users have done previously and start again at this exit point (when applicable).

Interaction and navigation

Good navigation is the primary key to good usability. Orientation and navigation are considered to be essential for usability issues [3]. Navigational actions compose nearly 90% of all actions performed in a web browser [63]. Consequently, to use any Web-based Information System as effectively as possible, the quality of navigation support is of primary importance (Fig. 4).

Problem areas concerning interaction and navigation can be subdivided into interaction style, scrolling and browsing.

End-users interact differently with handheld devices than with PCs (see for example [36, 26, 5]). For example, while it is acceptable to require keyboard input on desktop machines, typing on a PDA’s soft-keyboard is burdensome. With handheld devices it is much easier to select values from a list rather than enter text in an input field. On the other hand, scrolling, due to the increased Web experience

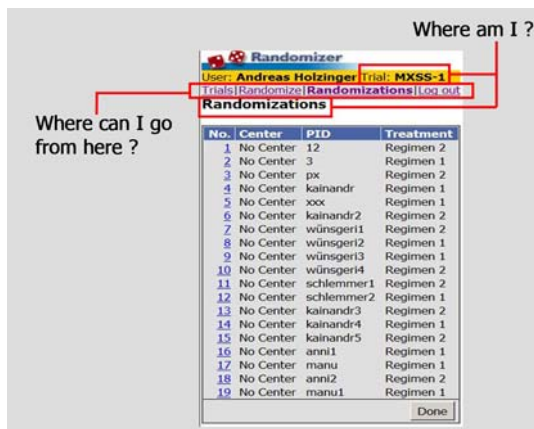


Fig. 4 The most important questions to be answered through the conducted experiments included: “Where am I?” and “Where can I go from here?”

of the end-users, is becoming better accepted than in previous years [28].

Guidelines on interaction and navigation are:

- Provide the core features of an application from its main view;
- Ensure that navigation keys never result in an unexpected action that the user cannot anticipate;
- Provide a context-sensitive menu if there is no single intuitive default action;
- Use soft-keys consistently;
- Use all terms consistently; and
- Avoid scrolling.

Conclusion

The availability of a mobile solution raised the acceptance of the Randomizer system amongst end-users. The experimental results and experiences in design and development provide some useful indications on how Web content can be modified in order to make it *more accessible*, in a universal access perspective, to end-users of mobile computers. It is essential to provide automatic adaptation of content; for example, by automatic preprocessing or providing different versions based on screen resolution and browser capabilities. The findings obtained in the domain of Medicine and Health Care can be generalized to other applications based on mobile computers. In particular, the screen size is limited; for example, a typical handheld, including the Hewlett Packard iPAQ, comes with a 240×320 pixel display. With Microsoft’s Pocket IE, the browser user interface further limits the visible content area to 229×255 pixels. A first experiment—using the standard Web-interface designed for desktop machines—did

not lead to useful results. End-users simply got lost in scrolling horizontally and vertically through the pages. Pocket IE’s Fit to Screen feature also did not help, since the page layout was too complex for reformatting to such a small screen resolution. Moreover, the page design depends on cascading stylesheets (CSS), and the Pocket IE available with the used version of the Pocket PC operating system has no CSS-support built in (XSL stylesheets can also be used). Therefore, detecting PDA-browsers on the server side and delivering pages optimized for a small client form factor is inevitable.

The most important guidelines include: Keep things as simple as possible. Every mobile device has limited resources. It is recommended to use a simple, mainly text-based interface with few small images. Pages must always be designed to allow dynamic resizing, fixed-size designs (e.g., using tables and transparent images for sizing table columns), and pages that need horizontal scrolling must be avoided.

During the study, it was observed that most end-users overlooked information requiring horizontal scrolling. Scrolling can most simply be reduced by strictly focusing on the content task. Flat menu hierarchies and simple single column layouts proved to be appropriate in most cases. Quick and easy navigation for frequent functionalities are an absolute necessity. The obtained experimental results confirmed that extensive text entry is to be avoided and that the priority of shortlists over direct input is important. UCD proved to be an appropriate design technique for this kind of mobile computer interface design, and finally led to a simple and easy to use interface, in accordance with the proverb: less is more.

References

1. Andrews, K.: Web usability on the cheap. In: Holzinger, A. (ed.) Human–Computer Interaction in the 21st Century, pp. 83–95. Austrian Computer Society, Vienna (2001)
2. Benbasat, I., Todd, P.: An experimental investigation of interface design alternatives: icon versus text, direct manipulation versus menus. *Int. J. Man. Mach. Stud.* **38**(3), 369–402 (1993)
3. Benyon, D.: The new HCI? navigation of information space. *Knowl. Based Syst.* **14**(8), 425–430 (2001)
4. Brekka, T.: Select mobile computers tailored to healthcare environment. *Health. Manag. Technol.* **16**(13), **48**, 50 (1995)
5. Brewster, S.: Overcoming the lack of screen space on mobile computers. *Personal Ubiquitous Comput.* **6**(3), 188–205 (2002)
6. Choi, J., Chun, J., Lee, K., Lee, S., Shin, D., Hyun, S., Kim, D., Kim, D.: MobileNurse: hand-held information system for point of nursing care. *Comput. Methods Programs Biomed.* **74**(3), 245–254 (2004)
7. Duchnicky, RL., Kolers, P.A.: Readability of text scrolled on visual display terminals as a function of window size. *Hum. Factors* **25**(6), 683–692 (1983)
8. Field, D., Elbourne, D.: The randomized controlled trial. *Curr. Paediatrics.* **13**(1), 53–57 (2003)

9. Forman, G.H., Zahorjan, J.: The challenges of mobile computing. *IEEE Comput.* **27**(4), 38–47 (1994)
10. Gittins, D.: Icon based human–computer interaction. *Int. J. Man. Mach. Stud.* **24**, 519–543 (1986)
11. Goethe, J.W.: *Farbenlehre* (Theory of colors, English from Charles Eastlake, 1970). Kohlhammer, Stuttgart (1810)
12. Goonetilleke, R.S., Martins Shih, H., Kai On, H., Fritsch, J.: Effects of training and representational characteristics in icon design. *Int. J. Hum. Comput. Stud.* **55**(5), 741–760 (2001)
13. Gould, J.D., Lewis, C.: Designing for usability: key principles and what designers think. *Commun. ACM* **28**(3), 300–311 (1985)
14. Hering, E.: “Principles of a new theory of color sense”. In: Teevan, R.C., Birney, R.C. (eds.) *Color Vision*. van Nostrand, Princeton (1961)
15. Hix, D., Hartson, H.R.: *Developing user interfaces: ensuring usability through product & process*. Wiley, New York (1993)
16. Holzinger, A.: Computer–aided mathematics instruction with mathematica 3.0. *Math. Educ. Res.* **6**(4), 37–40 (1997)
17. Holzinger, A.: *Multimedia Basics, vol. 1: Technology. Technological Fundamentals of Multimedia Information Systems*. Laxmi Publications, New Delhi (available also in German, www.basiswissen-multimedia.at) (2002)
18. Holzinger, A.: *Multimedia Basics, vol. 3: Design. Developmental Fundamentals of Multimedia Information Systems*. Laxmi Publications, New Delhi (available also in German, www.basiswissen-multimedia.at) (2002)
19. Holzinger, A.: User–centered interface design for disabled and elderly people: first experiences with designing a patient communication system (PACOSY). In: Miesenberger, K., Klaus, J., Zagler, W. (eds.) *Lecture Notes in Computer Science*, vol. 2398, pp. 34–41. Springer, Berlin (2002)
20. Holzinger, A.: Experiences with user centered development (UCD) for the front end of the virtual medical campus graz. In: Jacko, J.A., Stephanidis, C. (eds.) *Human–Computer Interaction, Theory and Practice*, pp. 123–127. Lawrence Erlbaum, Mahwah (2003)
21. Holzinger, A.: Application of rapid prototyping to the user interface development for a virtual medical campus. *IEEE. Softw.* **21**(1), 92–99 (2004)
22. Holzinger, A.: *Usability Engineering Methods*. Online at: <http://user.meduni-graz.at/andreas.holzinger/holzinger/usability.html> (last access: November 15, 2006) (2004)
23. Holzinger, A.: Usability engineering for software developers. *Commun. ACM* **48**(1), 71–74 (2005)
24. Holzinger, A., Errath, M.: Designing web–applications for mobile computers: experiences with applications to medicine. In: Stephanidis, C., Stry, C. (eds.) *User–Centered Interaction Paradigms for Universal Access in the Information Society. Lecture Notes of Computer Science*, vol. 3196, pp. 262–267. Springer, Berlin (2004)
25. Huang, S–M., Shieh, K–K., Chi, C–F.: Factors affecting the design of computer icons. *Int. J. Ind. Ergon.* **29**(4), 211–218 (2002)
26. Jessup, L.M., Robey, D.: Issues and challenges in mobile computing: the relevance of social issues in ubiquitous computing environments. *Commun. ACM* **45**(12), 88–91 (2002)
27. Karampelas, P., Akoumianakis, D., Stephanidis, C. (eds.): *User Interface Design for PDAs: Lessons and Experiences with the WARD–IN–HAND prototype*. Springer, Berlin (2003)
28. Kim, L., Albers, M.J.: Web design issues when searching for information in a small screen display. In: *19th Annual International Conference on Computer Documentation*, pp. 193–200. Santa Fe (2001)
29. Kjeldskov, J., Stage, J.: New techniques for usability evaluation of mobile systems. *Int. J. Hum. Comput. Stud.* **60**(5–6), 599–620 (2004)
30. Kleinrock, L.: Nomadcity anytime, anywhere in a disconnected world. *Mob. Netw. Appl.* **1**(4), 351–357 (1996)
31. Konstantakos, A.K.: Personal computers versus patient care: at the desktop or at the bedside? *Curr. Surg.* **60**(4), 353–355 (2003)
32. Krug, S.: *Don’t Make Me Think: A Common Sense Approach to Web Usability*. New Riders, Indianapolis (2000)
33. Lewis, J.R.: Sample sizes for usability studies: additional considerations. *Hum. Factors* **36**(2), 368–378 (1994)
34. Marcus, A.: The ten commandments of color. *Comput. Graph. Today* **3**(11), 7, **12**, 14 (1986)
35. Marcus, A.: Dare we define user–interface design? *Interaction* **9**(5), 19–24 (2002)
36. Marcus, A., Chen, E.: Designing the PDA of the future. *Interaction* **9**(1), 34–44 (2002)
37. Maulsby, D., Greenberg, S., Mander, R.: Prototyping an intelligent agent through wizard of Oz. In: *SIGCHI Conference on Human Factors in Computing Systems*, pp. 277–284. Amsterdam, The Netherlands (1993)
38. McDougall, S.J., de Bruijn, O., Curry, M.B.: Exploring the effects of icon characteristics on user performance: the role of icon concreteness, complexity, and distinctiveness. *J. Exp. Psychol.* **6**(4), 291–306 (2000)
39. Miller, G.A.: The magical number seven, plus or minus two: some limits of our capacity for processing information. *Psychol. Rev.* **63**, 81–97 (1956)
40. Mills, C.B., Weldon, L.J.: Reading text from computer screens. *ACM Comput. Surv.* **19**(4), 329–358 (1987)
41. Moffett, S.E., Menon, A.S., Meites, E.M., Kush, S., Lin, E.Y., Grappone, T., Lowe, H.L.: Preparing doctors for bedside computing. *Lancet* **362**(9377), 86 (2003)
42. Myers, B.A.: Using handhelds and PCs together. *Commun. ACM* **44**(11), 34–41 (2001)
43. Myers, B., Ioannidis, Y., Hollan, J., Cruz, I., Bryson, S., Bulterman, D., Catarci, T., Citrin, W., Glinert, E., Grudin, J.: Strategic directions in human–computer interaction. *ACM Comput. Surv.* **28**(4), 794–809 (1996)
44. Nielsen, J.: Paper versus computer implementations as mockup scenarios for heuristic evaluation. In: *IFIP TC13 3rd Interantional Conference on Human–Computer Interaction*, pp. 315–320 (1990)
45. Nielsen, J.: *Usability Engineering*. Morgan Kaufmann, San Francisco (1993)
46. Nielsen, J.: Estimating the number of subjects needed for a thinking aloud test. *Int. J. Hum. Comput. Stud.* **41**(3), 385–397 (1994)
47. Norman, D.A., Draper, S.: *User Centered System Design*. Erlbaum, Hillsdale (1986)
48. Oquist, G., Goldstein, M.: Towards an improved readability on mobile devices: evaluating adaptive rapid serial visual presentation. *Interact. Comput.* **15**(4), 539–558 (2003)
49. Pace, B.J.: Color combinations and contrast reversals on visual display units. In: *Conference Proceedings of the Human Factors Society, Santa Monica* (1984)
50. Peirce, C.S.: *Collected writings on semiotics*. In: Hartshorne, C., Weiss, P. (eds.) *The Collected Papers of Charles Sanders Peirce* 2, pp. 135ff. Harvard University Press, Cambridge (1932)
51. Rettig, M.: Prototyping for tiny fingers. *Commun. ACM* **37**(4), 21–27 (1994)
52. Reuss, E., Menozzi, M., Buchi, M., Koller, J., Krueger, H.: Information access at the point of care: what can we learn for designing a mobile CPR system? *Int. J. Med. Inform.* **73**(4), 363–369 (2004)
53. Shackel, B.: Ergonomics in design for usability. In: Harrison, M.D., Monk, A.F. (eds.) *HCI–86: People and Computers II: Designing for Usability*. Cambridge University Press, Cambridge (1986)

54. Shneiderman, B.: *Leonardo's Laptop: Human Needs and the New Computing Technologies*. MIT Press, Boston (2002)
55. Shneiderman, B., Plaisant, C.: *Designing the user interface*, 4th edn. Pearson, Harlow (2004)
56. Shubin, H., Falck, D., Johansen, A.G.: Exploring color in interface design. *Interaction* **3**(4), 36–48 (1996)
57. SIGGRAPH.: Physiological principles for the effective use of color. Online at: <http://www.siggraph.org/education/materials/HyperGraph/color/coloreff.htm> (last access: 20030908) (1999)
58. Simon, H.A.: How big is a chunk? *Science* **183**, 482–488 (1974)
59. Stry, C., Peschl, M.F.: Representation still matters—cognitive engineering and task-based user interface development. *Behav. Inf. Technol.* **17**(6), 338–360 (1998)
60. Stry, C., Vidakis, N., Mohacs, S., Nagelholz, M.: Workflow-oriented prototyping for the development of interactive software. *IEEE COMPSAC 1997*, pp. 530–535. Washington DC (1997)
61. Stephanidis, C., Salvendy, G., Akoumianakis, D., Arnold, A., Bevan, N., Dardailler, D., Emiliani, P.L., Iakovidis, I., Jenkins, P., Karshmer, A., Korn, P., Marcus, A., Murphy, H., Oppermann, C., Stry, C., Tamura, H., Tscheligi, M., Ueda, H., Weber, G., Ziegler, J.: Toward an information society for all: HCI challenges and R&D recommendations. *Int. J. Hum. Comput. Interact.* **11**(1), 1–28 (1999)
62. Sutcliffe, A.: *User-Centred Requirements Engineering: Theory & Practice*. Springer, Berlin (2002)
63. Tauscher, L., Greenberg, S.: How people revisit Web pages: empirical findings and implications for the design of history systems. *Int. J. Hum. Comput. Stud.* **47**, 97–137 (1997)
64. Tullis, T.S.: Screen design. In: Helander, M., Landauer, T.K., Prabhu, P. (eds.) *Handbook of Human-Computer Interaction*. Elsevier, Oxford (1997)
65. usabilitynet.org ISO 13407. Online at: <http://www.usabilitynet.org/tools/13407stds.htm> (last access: 20030611) (2003)
66. Virzi, R.A.: Refining the test phase of usability evaluation: how many subjects is enough? *Hum. Factors* **34**(4), 457–468 (1992)
67. VisiBone.: Webmasters Color Laboratory. Online at: www.visi-bone.com (last access: 20030922) (2003)
68. Young, P.M.C., Leung, R.M.W., Ho, L.M., McGhee, S.M.: An evaluation of the use of hand-held computers for bedside nursing care. *Int. J. Med. Inform.* **62**(2–3), 189–193 (2001)
69. Ziefle, M.: User productivity and different screen technologies: CRT screens with high refresh rates versus LCD displays. *Display Search Monitor* **6**(14), 11–14 (2001)