Navigating in a Mobile XHTML Application

Anne Kaikkonen
Nokia Research Center,
P.O. Box 407, 00045 Nokia Group, Finland
+358 504837400
anne.kaikkonen@nokia.com

Virpi Roto
Nokia Research Center,
P.O. Box 407, 00045 Nokia Group, Finland
+358 504836631
virpi.roto@nokia.com

ABSTRACT

The Internet has been a great success in the fixed world, whereas WAP (Wireless Application Protocol), the mobile Internet, has not fulfilled its promise. However, now the analysts have started to believe in a rise of the mobile Internet again. WAP 2.0, with XHTML Mobile Profile as its standard language, will enable sites to function both in the fixed and wireless worlds. In this paper, we analyze different ways to navigate XHTML sites with mobile phones and base our analysis on two usability evaluations with a total of 30 subjects from various countries. The results show that due to limitations of mobile devices (the limited display size, pointing methods, and bandwidth), not all navigation guidelines of the fixed Internet are applicable to the mobile Internet. It is important for developers to realize the effect of these limitations in order to build usable XHTML sites also for mobile use.

Categories & Subject Descriptors: H.5.2 [Information Interfaces and Presentation]: User Interfaces — evaluation/methodology, Interaction styles, Styleguides; H.1.2 [Models and Principles]: User/Machine Systems — human factors, human information processing; H.4.3 [Information Systems Applications]: Communications Applications — Information browsers; H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia — Navigation, User issues; C.5.3 [Computer System Implementation]: Microcomputers — Portable devices

General Terms: Design, Experimentation, Human factors

Keywords: Navigation, XHTML Mobile Profile, WAP, mobile Internet, wireless Web, mobile phones.

INTRODUCTION

In this paper, we investigate the usability of navigating XHTML Mobile Profile (MP) sites with a mobile phone. Navigation is defined as the path and actions needed to find a piece of information on a site and get back when needed.

XHTML Mobile Profile is the new language of WAP (Wireless Application Protocol) version 2.0 [1]. The biggest change from the old WAP system is that an XHTML MP site can be viewed with both mobile devices and an ordinary Web browser. When developers start to build their sites for both mobile devices and the fixed Web, they will need to know which kind of navigation works in both.

There are plenty of navigation guidelines for the Web, but not so many for WAP. In this study, we examined if the guidelines for the Web are applicable to mobile browsing.

Mobile devices differ from fixed systems, like PCs, dramatically [2,4,6,15]:

- Display size is small
- There is a lot of variation in display dimensions
- The number of color displays has just started to grow
- Text input is slower than with a full PC keyboard
- Usually there is no mouse for activating an object. This limits the possible user interface components and slows down object activation
- Some devices support only vertical scrolling
- Softkeys are commonly used for activating commands, but the number and purpose of softkeys varies between devices from different manufacturers
- Accesskey functionality allows users to activate commands also with the phone’s number keys
- Data transfer between the terminal and the server is slow
- The amount of cookie data that can be stored in a mobile device is very limited
- Context of use is harder to predict than with an office PC application
- The user may have to pay for each piece of data

One cannot access all the information available on the Internet with a small mobile device. Also, the mobile browsers running on mobile devices are different, and they interpret the standards and protocols in different ways.

CHI 2003, April 5–10, 2003, Ft. Lauderdale, Florida, USA.

Even the basic link activation differs considerably according to the manufacturer of the browser software.

Because of the small display size, there is not always space for a navigation bar or other navigation aids on the page. Current WAP applications use a simple tree hierarchy for sites to ensure streamlined navigation.

Because of the limited bandwidth of the current wireless networks, the slowness of WAP is a similar problem for WAP site users as Web slowness was in the early days of the World Wide Web.

HISTORY AND RELATED RESEARCH

The number of PCs with an Internet connection has been growing quickly, and so has the number of WAP-enabled mobile phones. In late 1990's, there was an expectation that with WAP the mobile Internet would soon grow much faster than the fixed Internet. WAP was advertised to be 'Internet in your pocket', but the reality was different. Although the technology behind WAP was similar to that of the Web, small devices with a slow connection speed and only a few available WAP sites did not fulfill the users' dreams of mobile Internet access.

Mobile Internet applications have been in public use for a few years. The protocol used in applications in most parts of the world is WAP (Wireless Application Protocol). The first commercially available applications were developed for WAP 1.1; later, the mobile browsers started to support WAP 1.2 features. The markup language for both protocols was defined to be WML (Wireless Markup Language). The former WAP Forum specified the language for WAP 2.0 to be XHTML Mobile Profile [1,6,7]. This change of the markup language brings the mobile Internet closer to the fixed Internet, since the same sites will be available for both fixed and wireless devices.

Several guidelines exist for designing WAP applications. Some guidelines are provided by browser or device manufactures [6,15], and some by network operators or service developers [12]. Even though there are not many XHTML-enabled mobile devices on the market at the time of this study, guidelines for developing XHTML applications are already emerging [7,14,15].

It has been hard to validate these guidelines, since there are few studies publicly available [9] about the exact methods and results used in guideline development. With this paper, we hope to initiate public discussion of the methods used in guideline development.

Because XHTML Mobile Profile is very similar to HTML, the guidelines for developing sites for mobile use could also be very similar. There are a lot of instructions and guidelines for Web site development [e.g.10,11,13], which are now even more relevant in WAP site development.

There are some general Web navigation guidelines that clearly are applicable also for WAP:

- Navigation is a necessary evil that is not a goal in itself and should be minimized [10]
- Focus on user goals and needs [13]
- Customize service [10]

There are also some guidelines that, because of the nature of mobile browsing, are not necessarily important in the mobile environment:

- "Make it obvious what's clickable" [11] is not so important with devices that do not provide a pointing device. This is because the focus moves from link to link, like when using the Tab key on a PC to move on the page, and it is the browser's task to make the focus obvious enough. The clickability information, however, helps the user to predict where the focus will move next and thus to follow the system more easily.

The guidelines that need to be questioned in the mobile environment:

- "Show current location at two different levels:
  Relative to the Web as a whole, and relative to the site's structure" [10].
  How much location information should be presented on a small screen to make the current position clear?
- "Support search and multiple views" [10].
  Writing search keywords is laborious, so should we simplify and save screen space by leaving a search out?
- "Be small and cheap" [10].
  What is the optimal size for a WAP page?

OBJECTIVE OF THE STUDY

The main objective of this usability evaluation was to make guidelines for developers that develop XHTML applications and services. The basis of these guidelines is in studies made for WML services and on the guidelines for Web sites.

In this paper, we focus on the following navigation issues:

- Knowing where you are
- Finding your way forward
- Finding your way back
- Long pages and flat hierarchy vs. short pages and deep hierarchy
- Navigating in a list of items
- Minimizing steps in navigation

Although navigation goes hand in hand with the site structure, we did not analyze the different site structures here. The sites we tested were basically structured as a tree hierarchy, with some linear parts included.

We did not test navigation via site maps here, because the site map is hard to implement nicely on a small screen. It might be interesting, however, to test if the users would like site maps for WAP sites.

SERVICES TESTED

The user interfaces used in the test were specified and developed just for this evaluation. Only the parts needed to carry out the test tasks were implemented.

All the user interfaces contained the same data, but the way the data was presented, navigation, and the usage of the elements varied in each user interface style. To avoid associations that would make one style better than other, we named the styles according to fruits:

Banana “long content”	Long pages, flat hierarchy; Selection lists; Layout tables; Images
Orange “slices”	Short pages, deep hierarchy; Multi-page forms; Choice for text input or value selection; Data tables; Small images
Apple “for experts”	Keyword search prioritized; Accesskey shortcuts; Textual input; No images

The structures of the 3 styles are presented in figures 1, 2 and 3. The number of users’ steps for finding information about a basketball game can also be seen in the figures.

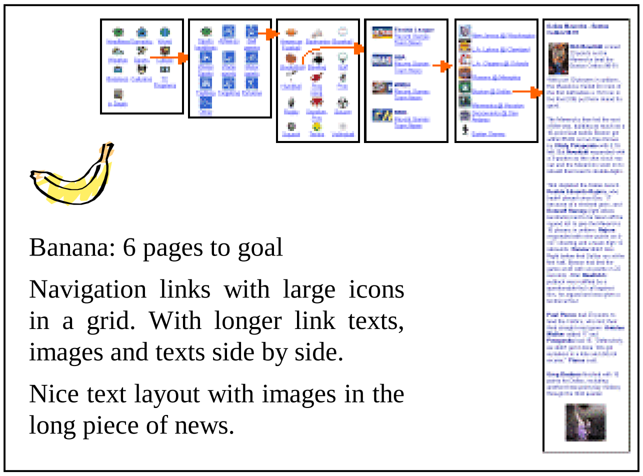
METHOD

The first test was carried out with Nokia 6510 (Europe) and 6590 (U.S.A.) mobile phones (referred to as a "small phone") and the second test with Nokia 7650 ("big phone"), all running a prototype version of the Nokia mobile XHTML browser. The applications were used via GPRS connection as WAP services, and the implementation was pure XHTML Mobile Profile. Pure XHTML Mobile Profile is the safest way for developers to ensure that the service functions in all mobile devices using XHTML.

In the first test, the number of subjects was 20: 12 in Helsinki, Finland, and 8 in Boston, U.S.A. The subjects in Finland were from various European countries and from Japan. The subjects varied from active users of the current mobile Internet to ones that had never used it. All but one user knew, however, at least the principle of either WAP in Europe/USA or Japanese mobile Internet (i-mode, J-sky or EZ-web).

In the second test, with the Nokia 7650, the number of test persons was 10. Eight users were Finnish, one was Romanian, and one Indian. In the first test, we found that experienced with mobile application usage than the users in the first test. In the first test, over half of the users had used WAP or other mobile Internet, while in the second test only four users had gained such experience.

All subjects in both tests used a mobile phone daily, and they knew how to type with a mobile phone keypad; they had either inserted names in the phonebook of the phone or written text messages.

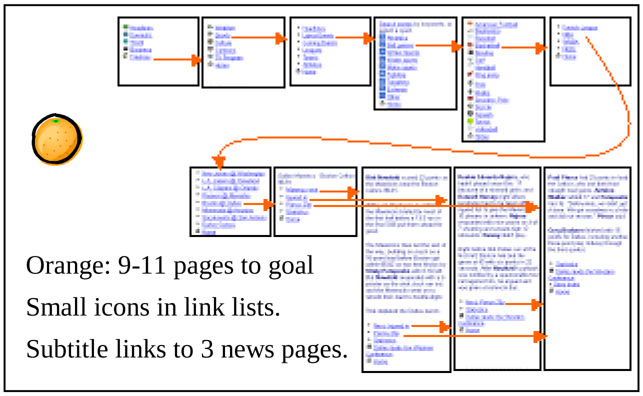


Banana: 6 pages to goal

Navigation links with large icons in a grid. With longer link texts, images and texts side by side.

Nice text layout with images in the long piece of news.

Figure 1: Banana style for finding sports news

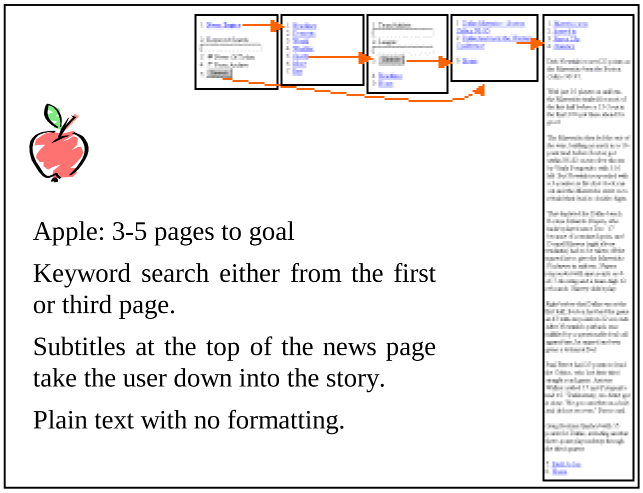


Orange: 9-11 pages to goal

Small icons in link lists.

Subtitle links to 3 news pages.

Figure 2: Orange style for finding sports news



Apple: 3-5 pages to goal

Keyword search either from the first or third page.

Subtitles at the top of the news page take the user down into the story.

Plain text with no formatting.

Figure 3: Apple style for finding sports news

The picture of the mobile phone screen was captured with a mini-camera that was attached to the phone. Only the phone screen and the button area were recorded, together with the finger movements of the subject. Users were asked to think aloud, and their comments were recorded together with the video image on a digital video camera. The test moderator could follow the exact user's actions from the LCD screen of the video camera.

The applications in both tests were almost the same. Some obvious usability flaws were fixed in the second test, like the all too short pages of the Orange style. We also added the images on pages only in the second test.

The order of the services and user interface styles was counterbalanced.

The tasks users were asked to perform in the News service:

1. Finding the weather forecast for a specified location and a specified period
2. Finding information about the performance of one basketball player in a specified match

Tasks in the Auction service:

1. Deleting an offering from the list of own offerings
2. Modifying an offering on the list of own offerings

Each user performed all tasks three times, once per user interface style. They were allowed to use the time they needed to execute the task; there was no time limit. Most users performed all the tasks with all the user interface styles within 2 hours, but some users did not have the time to complete all the tasks.

After testing each service with a certain user interface style, users were asked to rate the user interface style on a scale of 1 to 5, 1 meaning very bad/difficult and 5 very good/easy.

The questions were:

1. Did you always know how to proceed?
2. How easy it was to locate/ recognize components?
3. How quickly did the system let you execute the task?
4. Did you like the visual outlook?

We analyzed the times in specified tasks and user satisfaction ratings. We also performed a qualitative analysis of the user performance in test tasks.

RESULTS AND DISCUSSION

Task Execution Times

In the News service, the Orange user interface style with split pages was slower than the Banana and Apple styles. The difference was statistically very significant. (F_2=16.57; p<0.001). The differences in Auction task times were not statistically significant.

Small screen phone	Banana News	Apple News	Orange News	Banana Auction	Apple Auction	Orange Auction
average (sec)	291	249	**789**	450	388	326
st. dev.	109	85	**418**	216	140	124

Table 1: Task execution times in the first test

There were not significant differences in performance times in the second test with a bigger phone, Nokia 7650.

Big screen phone	Banana News	Apple News	Orange News	Banana Auction	Apple Auction	Orange Auction
average (sec)	377	225	341	409	381	394
st.dev	91	89	170	171	138	235

Table 2: Task execution times in the second test

Subjective ratings of the different user interface styles

When using the services with a phone that has a small monochrome display, there were differences in users' experiences of the user interface. In the News service, Orange news (split pages) was clearly worse when it came to knowing how to proceed (F_2 = 15.09, p< 0.001) and in the ease of locating items (F_2=12.09, p<0.001) (Figure 4).

In the auction service, users preferred the Apple style with the Search function in the ease of locating items (F_2 = 4.05, p<0.001) and in the look of the pages (F_2 = 3.94, p<0.05) (Figure 4).

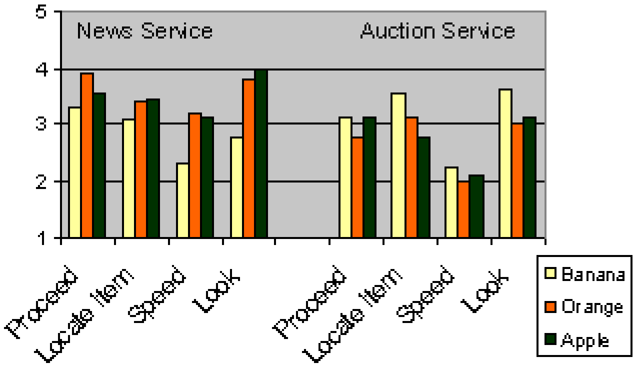


Figure 4: Subjective ratings of the styles (1st test)

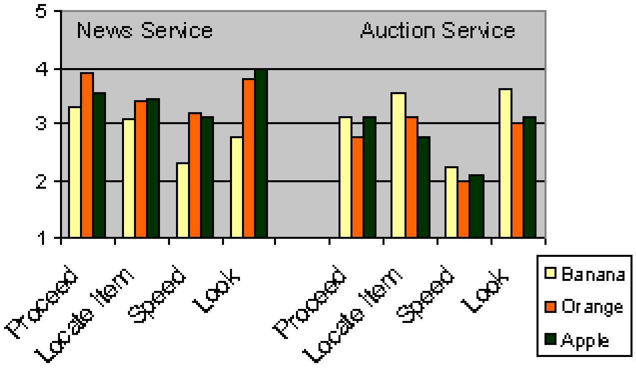


Figure 5: Subjective ratings of the styles (2nd test)

In the second test, the phone display was bigger, and the Orange user interface was redesigned so that the too short pages were combined into longer ones. These changes made the Orange style more successful, and there were no statistically significant differences between the styles (Figure 5).

Knowing where you are

In mobile services, location information is normally put in the page title, and the title is shown on the first line of the display. We tested providing either the site subsection name (e.g. Sports or Weather) or a unique page name (e.g. Basketball or Forecast) in the page title.

All the users noticed the page titles when they wanted to understand their location in the site structure. This was most obvious when they estimated how much they still have to scroll to find the item they were looking for, for example, in the Apple style of the Auction service. There the offering items were organized on 5 pages, and the page title showed the current position: "My Offerings 2/5". Over half of the users spontaneously articulated their location according to the page title at some point of the test, for example “So now I’m on a Sport page” when the title of page was SPORT.

There were no significant differences in navigation success between the service with subsection titles and the one with unique page titles. However, when the users wanted to go back by selecting a page from the History list, it was clearly easier to navigate back with the unique page titles.

On Web pages, the location information cannot be shown in the page title only. We found that in WAP, however, the page title is an adequate way to communicate the current location to the users.

Finding your way forward

The users could proceed towards the target page by selecting hyperlinks on the page. The link hierarchy constituted the tree structure of the site. In the Apple style of the News service, users were forced to use the Search function at some point to get to the target page.

In the Orange style, the users could select between the Search function and the hyperlinks. The user could select which one s/he wanted to use. Our hypothesis was that users would prefer the hyperlinks over the keyword search, because it is relatively difficult to write text with a normal phone keypad. There also exists a WAP design rule: "minimize the need for text entry in mobile services" [6].

In our test service, the system always provided two matches for a keyword search. That gave the impression of efficiency. In some cases users could alternatively make selections and go forward by selecting items with a radio button or check box.

All the users in tests managed to activate the links on the page without problems. Links were preferred over radio buttons and submit buttons, which require more action from the users than link selection. All the users in the test thought that the item they first selected with a radio button remained selected when they selected a new item.

Users were also pleased in using the efficient Search function, and complained about the styles that did not offer such a tool. The way the Search function was provided affected its popularity a lot: if the keyword field was right on the page and not behind a link, users selected it surprisingly often.

Understanding Accesskey shortcuts

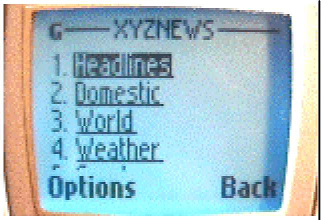
In XHTML Mobile Profile, it is possible to define an Accesskey for each interactive component on the page. This means that the user may activate a component by pressing the corresponding key on the phone keypad. This way, s/he does not need to click her/his way to a hyperlink at the bottom of the screen but can activate it with a single key press.

How does the user find out if this kind of functionality exists and which key to press for a specific link? There are guidelines to provide numbered link lists and to use the keys consistently for standard functions [6]. It is assumed that the user will realize the Accesskey feature either by reading the phone manual or by guessing.

We found that the users did not find the Accesskey functionality easily. In our first test, it was hard for most users to interpret the meaning of the numbers (Figure 6). If there were more than 10 links on the page, not all links could be assigned an Accesskey, and the users wondered why some links are numbered and some not. Only 4 users spontaneously pressed a key to activate a link, but 16 out of 20 users did not find an appropriate explanation for the numbers during the test.

In the second evaluation, we provided users an image of a number key in front of each link, hoping that this would give them a clear enough hint about key pressing (Figure 7). We did not succeed, because now only five out of ten users figured out the Accesskey functionality. Many users complained, however, about the download time of the images. Accesskeys should make the usage quick, and waiting for the images to download is not acceptable here.

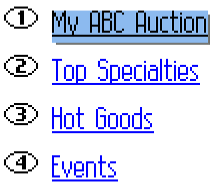
Unfortunately, this prototype version of the browser did not respond to key pressing appropriately, and we could not evaluate how frequently the users would have used the Accesskey shortcuts. According to our test, it is hard for the users to figure out the meaning of numbered link lists.



XYZNEWS

1. Headlines
2. Domestic
3. World
4. Weather

Figure 6: First evaluation, Numbered Accesskey links



① My ABC Auction

② Top Specialties

③ Hot Goods

④ Events

Figure 7: Second evaluation, Accesskey links as images

Finding your way back

In the first test, the Back function was provided in the rightmost softkey of the phone, right below the display. The browser provided no way to jump back several pages. In the second test, the same softkey opened up a History menu containing the Back function together with a list of visited pages.

The Apple and Orange styles provided links back to the Home page and sometimes to a subsection main page, at the bottom of the pages. In the Banana style, the browser functions were the only way to move back.

In the first test, all users demanded to have the backwards links on the pages, because there was no other way to jump back several pages at one time. In the second test, users utilized the History list of the browser, and did not complain about missing links on the page. Still, four users out of ten in the second test jumped back with the Home link at the bottom of the page. As the link did not seem to disturb other users, we recommend providing the Home link at the end of the pages.

Long page and flat hierarchy, or short pages and deep hierarchy?

Currently, WAP services are often used over a low bandwidth connection, and the download time is an issue. The old WAP with WML provides decks of cards instead of single pages. The system shows the first card to the user and downloads the next cards of the deck automatically in the background, so that moving from card to card is relatively quick. In that case, it is easy to minimize scrolling by providing short pages and quick movement from card to card. In XHTML, the pages are downloaded one by one, and the system does not start to download the next pages until the user initiates the download. The browser could, of course, start to download the linked pages on the background automatically, but this does not fit into the billing model where the user pays for data and not for connection time.

Sliced pages to minimize scrolling?

In the first study, we implemented the Orange style of the News service so that the pages fit on the screen, and the user was provided a quick way to move to the next page where the same content continued. The Banana style provided the same information on long pages with less navigation levels.

The Orange style with sliced information and no need to scroll was very irritating for all users. The main reason was the delay in getting the next piece of information onto the screen. There was no single user in the test that would have liked the idea of splitting close-knit content onto several pages. Scrolling down the page is quicker and the users seem to be more confident of their location in the application than when jumping from page to page.

The less information there was on a single page, the more the users complained about the download time of that page. In reality, long pages downloaded almost as fast as short pages, but the users seemed to cleverly estimate the acceptable download time based on the amount of expected data. When the pages were short, users did not accept as long download time as they did for long pages.

According to our evaluation, it is evident that splitting close-knit information to several short pages does not function well in WAP 2.0. It is better for users to scroll a little than download new pages.

Long pages with scrolling?

The longest page in all the user interface styles was a piece of news about a basketball game. In the small phone used in the first test, the text took 20 screens. None of the users complained about the text length of this story.

The Apple style provided within-page links at the beginning of the story to make it quicker to jump to the interesting part of the text. Twenty users out of thirty used the within-page links, but half of these users tested the links only after finding the information from the page by scrolling.

Half of the users complained about the need for scrolling up and down in the Orange style of Auction, where there was a 6-screen-long table with edit buttons at the bottom.

We found that the optimal length of a page depends on the page type: an interactive page has to be much shorter than an information page. In this test, six screens was all too long for an interactive page, but 20 screens fine for a text page. Within-page links may help the user to locate information on a long text page.

Navigating in a list of items

One common user interface issue is how to provide a usable way to check and/or manipulate a list of items, each of which contain some properties that would be too long to show all on a single page. In our sample service, the items were 10 stamps that the user had put for sale in the mobile auction service. A similar user interface solution will work also for the items in my mobile shopping cart, bills queuing to be paid, trains from place A to B on a certain day, or basic news headlines.

We evaluated three ways for the users to manipulate their offerings. In the Apple style, the user was provided tools to search and sort the items, and then view 2 items per page in alphabetical order. The default was to show 2 items per page in alphabetical order, and the link to the next 2 items was at the bottom of the page.

In the Orange style, we provided the offerings in a wide table, where the user could check the properties of all items on one page, and when s/he wanted to manipulate an item s/he selected it with a radio button and then selected an appropriate function button. Property editing was performed on a separate page.

The third way to manipulate the list of offerings was presented in the Banana style: the offering names were provided as hyperlinks on one page, and each offering was then presented on a separate page. From the item page, the user could move either to the previous or next offering, or back to the list of all offerings.

We asked the users first to delete a named offering (6th item), and then to halve the price of those offerings that nobody was willing to buy. In the first task, the user could navigate by item name, but in the second task they had to check a property named Best Bid to find the items that did not have any bids.

Deleting a named item

The item deletion task was easy in the Banana style, because there was a compact list of item names, and the user could spot the item there easily. In the Apple style, users that used search to find the item by name (15 in first test and 6 in the second) were very pleased with the functionality. Unfortunately, the rest of the users browsed the offerings in the default alphabetical order, and were frustrated when they had to browse through 2-5 long item pages until they found the offering they were looking for.

The item table in the Orange style was the least usable solution for this task, although from the navigation perspective it was the easiest: you did not have to leave the page to delete the item. The first difficulty was to realize the procedure to delete the item. Because the text on the table was wrapped onto several lines, only 2-3 radio buttons out of 10 were visible on the screen at a time. It was surprisingly hard even for the expert users to realize that the empty circles were radio buttons. The second difficulty was to find the deletion function after the item was selected. The table took up several screens and the Cancel Offering button was down there. Many users wanted to find the command right after selecting the radio button from the Options menu. When they eventually scrolled down, the first button that came into view was Edit Offering. Many users selected this one instead of the deletion button, because they did not scroll down enough to see the correct button.

Navigating to an item having a specific property value

The price-halving task was much harder than the first task in the Banana style, because the compact list of items did not reveal which items had no bids. The had to browse through all the items until they found what they wanted. Now, many users commented that the main properties of the item should be brought up to the list of items, as tiny images, for example.

Half of the users in both tests moved from item to item directly by using the previous and next triangles on the item pages. The other half did not utilize these shortcuts, but they returned to the list of items and then selected the next offering on the list. The reason for not using the Next link is most probably that the triangle design was not understandable enough in this service. Price editing itself was done on the item page, so that did not cause any navigational problems.

The Apple style provided may be the most elegant way for price editing, but it was not a trivial one. The optimal way was to sort the items by Best Bid values, in ascending order. The items that had no bid at all were listed first, and the users could edit the prices right on the first item page. Again, only 7 users realized that there was an elegant sorting option available. They were very happy with the user interface. The other 23 users had to struggle through long item pages to the right position.

Minimizing steps in navigation

In a mobile context with a low bandwidth connection and often a limited time for the service usage, it is very annoying to see needless pages along the navigation path. It is easy to believe the mobile service guidelines that recommend minimizing the number of steps to get to the goal. On the other hand, in the fixed Internet world, many have adopted the guideline “It doesn't matter how many times I have to click, as long as each click is a mindless, unambiguous choice” [11]. We wanted to test how many and what kind of steps the users were willing to take when using a mobile service.

Login

In the auction service, the user had to log in before being able to edit the offerings. All subjects accepted this step in this service. Three users commented spontaneously, that it is irritating to have each mobile service ask you to login to their site, and you will no more be able to remember the user names and passwords for each.

Login should be asked only when the service contains very confidential information, or deals with user's money. Content providers' need to know when and why a specified user is accessing the service is not a valid reason to force a user to log in. Sometimes the service needs to know the user only to provide a customized service view to be user-friendly. In such cases, the developer should find a solution that does not require any actions from the user, like providing a unique link for each user and ask them to bookmark that page. Then, login is not needed, but the service provider can still identify the user.

Confirmations

In the auction tasks, we wanted to test different kinds of confirmation procedures for deletion and price change. The hypothesis was that the users would like one extra step after item deletion to confirm the action, but two steps would be too much.

In the Orange style, we asked the user to confirm deletion, but not the change of prices. After deletion, we did not inform the user if the item was successfully deleted. 21 out of 30 users wanted to take the time to check from the list of items if the change really took place.

In the Apple style, we only showed a message “Your offering was deleted” with OK and Undo buttons. Only 7 users read through this message; the rest just selected the OK button. The ones who read the text were off balance because they expected the system to confirm deletion before doing so. It was clear that this type of information, even with an Undo option, frightened the users.

The Banana style with maximal confirmations was clearly preferred. After deletion, the user was first asked to confirm the deletion, then the system informed that the deletion took place successfully and provided several alternatives to proceed. After the price change, the system asked for confirmation and then returned to the same page.

We found that confirmations are extra steps, but must be there. No participant complained about them. However, if the confirmations were missing, the users needed to go to check if the change really took place. That was much more laborious for the user than to select OK on the confirmation page.

CONCLUSIONS

According to our studies, the guidelines for the Web differ from the guidelines for WAP, and the guidelines for WAP 2.0 differ from the old WAP guidelines.

Knowing where you are: We found that unique page titles are the primary way to communicate the location information to users.

Finding your way forward: The keyword search was surprisingly popular in our study. This finding is conflicting with the current guideline to minimize text input in WAP services. On the other hand, users did not have problems with streamlined tree navigation with links.

Finding your way back: The browser Back function is the most important way to go back. When users need to go back several steps, links to home and subsection main pages are useful.

Long page and flat hierarchy, or short pages and deep hierarchy: Close-knit information should not be split on too many pages. The optimal length of a page depends on the type of the page (informative vs. interactive).

Navigating in list of items: A compact list of items on one page was clearly preferred by the users. From the list of all items, the user should be able to go to a detailed view of one item.

Minimizing steps in navigation should not increase the users' feeling of insecurity.

NEXT STEPS

There is very little research available on mobile Internet usability. A lot of mobile Internet usability questions remain still unanswered, like longer-term Accesskey usage, ways to show current location both in Web and WAP, breadcrumb navigation on a small screen, and a broader question of what is the applicable consistency level of the pages viewed through a PC and a mobile device. Long-term field tests should be carried out to get information beyond the first time use. We look forward to see more research on these topics.

REFERENCES

1. Wireless Application Protocol Forum Ltd.: XHTML Mobile Profile Specification. http://www.wapforum.org/what/technical.htm
2. Kaikkonen, A., Williams, D.: "Here, there, everywhere": Designing Usable Wireless Services, Interact 2001 Tutorial Notes (2001)
3. Kaikkonen, A., Törmänen, P.: User Experience in Mobile Banking, HCI2000 Proceedings, BCS (2000)
4. Kaikkonen, A., Williams, D.: Designing Usable Mobile Services, CHI2000 Tutorial Notes, ACM (2000)
5. Kaasinen, E., Kolari, J., Laakko, T.: Mobile-Transparent Access to Web Services. Interact 2001 Proceedings (2001)
6. Nokia Corporation: Mobile Application Development Guidelines. http://www.forum.nokia.com (2002)
7. Nokia Corporation: XHTML Guidelines. http://www.forum.nokia.com (2002)
8. Kaikkonen,A., Roto,V.: XHTML in Mobile Application Development. Mobile HCI 2002 Proceedings (2002)
9. Chittaro, L., Dal Cin, P.: Evaluating Interface Design Choices on WAP Phones: Single-choice List Selection and Navigation among Cards. Mobile HCI 2001 Proceedings (2001)
10. Nielsen, Jakob: Designing Web Usability: The Practice of Simplicity. New Riders Publishing (2000)
11. Krug, Steve: Don't Make Me Think: A Common Sense Approach to Web Usability. New Riders Publishing (2000)
12. Serco Usability Services: Designing WAP Services: Usability Guidelines (2000) http://www.usability.serco.com/research/suswapguide.pdf
13. Fleming, J., Koman, R. (editor): Web Navigation: Designing the User Experience. O'Reilly & Associates (1998)
14. Sprint Spectrum L.P.: PCS Web Style Guide, Vision Edition (2002). http://www.sprintpcs.com/
15. Openwave Systems Inc.: XHTML Mobile Profile Reference (2001). http://www.openwave.com