Minimap – A Web Page Visualization Method for Mobile Phones

Virpi Roto[1], Andrei Popescu[1], Antti Koivisto[2], Elina Vartiainen[1]
[1] Nokia Research Center, P.O.Box 407, 00045 Nokia Group, Finland
[2] Nokia Technology Platforms, 5 Wayside Road, Burlington, MA 01803, U.S.A
[virpi.roto, andrei.popescu, antti.j.koivisto, elina.vartiainen]@nokia.com

ABSTRACT
The Web has become available even on mobile phones, but the current methods to view large pages on small screens have not been highly usable. Current mobile phone browsers reformat Web pages to a single column that fits the screen width. Because not all content is comprehensible in this format, browsers provide a second mode for viewing pages in the same layout as on a PC. We have developed a modeless Web page visualization method called Minimap that shows pages in a modified Original layout. We conducted a long-term usability study with 20 participants to compare the state-of-the-art mobile phone browser with this new method. 18 participants preferred the new method, and it also scored better in more detailed usability ratings.

Author Keywords
Information visualization, mobile Web browser, usability.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces – *Screen design / Interaction styles*

INTRODUCTION
Most people think it is a ludicrous idea to view Web pages on mobile phones because of the small screen and slow connection. We partly agree. There is no doubt that Web content formatted specifically for small screen devices is more usable on mobile phones than content designed for large desktop screens. Sometimes, however, a user needs to access the full Web page because the information s/he needs is available in a full Web page only, or because s/he simply does not know the URL of the mobile friendly site. A large Web page might be needed to find the link to a mobile site. This means we need to provide the user the possibility of browsing full Web pages, but we must not forget that the browser should show mobile sites as originally designed.

Although many mobile phones today are still too limited for Web browsing, high-end phones do provide good quality color displays of 170x200 or more pixels, which enable showing full Web content on the screen. Many high-end phones support 3G (3rd generation mobile phone technology) or even WLAN (Wireless Local Area Network) connections, greatly reducing the response times in browsing.

In addition to the limited screen size and connection speed, there is a third challenge for Web browsers running on mobile phones: the lack of a pointing tool (e.g. mouse or stylus). Most mobile phones provide an input device that allows 5-way functionality: vertical and horizontal movement plus a press as the select action. To select an object on the screen, e.g. a hyperlink, with this 5-way control requires moving the focus to the wanted object with vertical and horizontal movement and then pressing the control. The same control should be used for scrolling the view, so in the most intuitive implementations scrolling and focus moving is done simultaneously.

Zooming is sometimes suggested as an intuitive solution for viewing large content on small screens. Most mobile phones do not, however, provide a dedicated key for zooming, so zooming would need to be hidden behind some number key or it would be a mode. Neither of these options is satisfactorily usable for a visualization method that would rely on zooming as a solution.

Looking to the future, device capabilities will increase and so will the number of phones that are capable of providing a decent user experience for mobile Web browsing. Most of the new Internet users will come from developing countries, and for many of these users, the mobile phone will be their only access to the Web. Because of these developments, Web access via mobile phones will be an increasingly common use case in the near future.

PRIOR ART
Browsing full sized Web content on a mobile device is like viewing a desktop screen through a paper towel tube – it is hard to know where the target content is located and one easily gets lost. Information visualization science has produced various methods for showing large content on small screens [15]. The key is to provide both focus view to show readable content in detail and context view to provide

orientation information on the large data space. In the overview + detail methods, an overview and the detailed view are provided separately. The views can be visible at the same time either in different locations [16,20], with partial overlap [7], or the views occupy the same area and the user switches the view when needed [12,13]. Transparency can be used to show both views at the same time at the same location [11]. Fisheye view is a nice modeless solution to show both focus and context in a single view [8], but requires more processing power than is available on current mobile phones.

The researchers have taken the challenge and invented various methods addressing the screen size and connection speed problems in full Web browsing on handheld devices. Many of these methods require a device with a medium-sized screen, such as a PDA (Personal Digital Assistant) [1,3,19,20,21] and some count on a pointing device i.e. a stylus [1,5,7,10,13]. A few methods are specifically designed for truly small, even black-and-white screens and for a slow CSD (Circuit Switched Data) connection [17]. Especially devices with limited memory benefit from methods that split the pages into smaller pieces and try to identify the main content on the page [1,4,6,17,21]. However, it is very hard, if not impossible, to automatically identify logical pieces of content on all the various layouts and types of Web pages. There are also interesting zooming solutions taking advantage of the fact that overview is mostly needed while scrolling [9]. Solutions such as that are likely to also work fine on mobile phones without a pointing device, but we still have to wait for more processing power to enable these to work smoothly.

A vast majority of current commercial Web browsers on mobile phones provide two alternative methods to view a Web page: an Original layout and a Narrow layout [14]. The Original layout shows the page as it is shown on a PC, in the form that the page author originally designed it (Figure 1). The

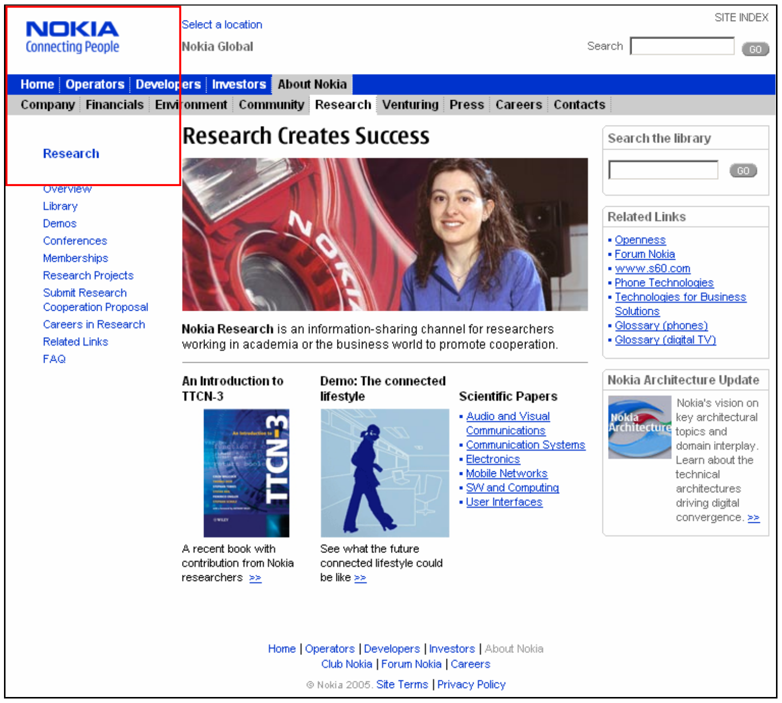
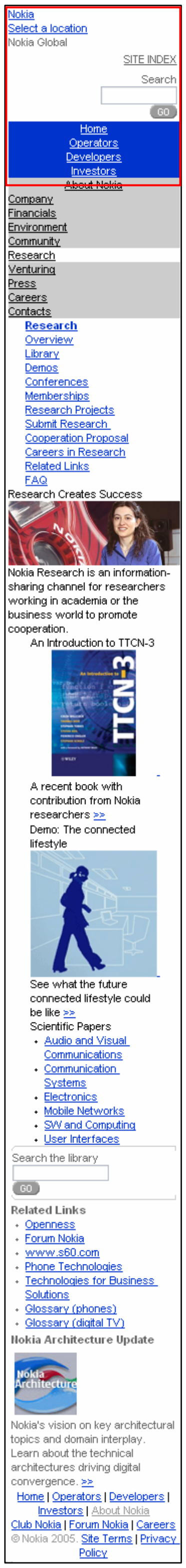

NOKIA Connecting People
Select a location
Nokia Global
SITE INDEX
Search GO
Home | Operators | Developers | Investors | About Nokia
Company | Financials | Environment | Community | Research | Venturing | Press | Careers | Contacts
Research
Overview
Library
Demos
Conferences
Memberships
Research Projects
Submit Research Cooperation Proposal
Careers in Research
Related Links
FAQ
Research Creates Success
Nokia Research is an information-sharing channel for researchers working in academia or the business world to promote cooperation.
An Introduction to TTCN-3
A recent book with contribution from Nokia researchers >>
Demo: The connected lifestyle
See what the future connected lifestyle could be like >>
Scientific Papers
- Audio and Visual Communications
- Communication Systems
- Electronics
- Mobile Networks
- SW and Computing
- User Interfaces
Search the library
GO
Related Links
- Openness
- Forum Nokia
- www.s60.com
- Phone Technologies
- Technologies for Business Solutions
- Glossary (phones)
- Glossary (digital TV)
Nokia Architecture Update
Nokia's vision on key architectural topics and domain interplay. Learn about the technical architectures driving digital convergence. >>
Home | Operators | Developers | Investors | About Nokia
Club Nokia | Forum Nokia | Careers
© Nokia 2005. Site Terms | Privacy Policy

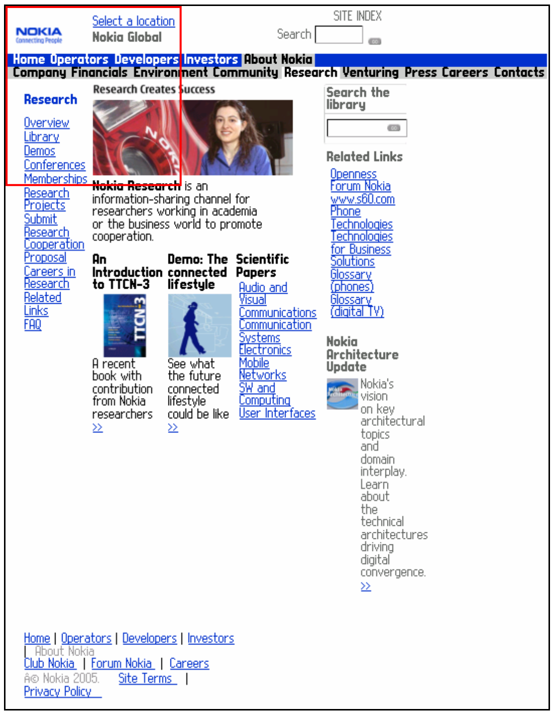
Figure 1. (above)
Original layout of a Web page (www.nokia.com/research).

Figure 2. (on the right)
Narrow layout of the same Web page.

Figure 3. (below)
Minimap layout of the same Web page.

The frame on top of each figure depicts the viewport size.


Nokia
Select a location
Nokia Global
SITE INDEX
Search
GO
Home
Operators
Developers
Investors
About Nokia
Company
Financials
Environment
Community
Research
Venturing
Press
Careers
Contacts
Research
Overview
Library
Demos
Conferences
Memberships
Research Projects
Submit Research
Cooperation Proposal
Careers in Research
Related Links
FAQ
Research Creates Success
Nokia Research is an information-sharing channel for researchers working in academia or the business world to promote cooperation.
An Introduction to TTCN-3
A recent book with contribution from Nokia researchers >>
Demo: The connected lifestyle
See what the future connected lifestyle could be like >>
Scientific Papers
- Audio and Visual Communications
- Communication Systems
- Electronics
- Mobile Networks
- SW and Computing
- User Interfaces
Search the library
GO
Related Links
- Openness
- Forum Nokia
- www.s60.com
- Phone Technologies
- Technologies for Business Solutions
- Glossary (phones)
- Glossary (digital TV)
Nokia Architecture Update
Nokia's vision on key architectural topics and domain interplay. Learn about the technical architectures driving digital convergence. >>
Home | Operators | Developers | Investors | About Nokia
Club Nokia | Forum Nokia | Careers
© Nokia 2005. Site Terms | Privacy Policy


NOKIA Connecting People
Select a location
Nokia Global
SITE INDEX
Search
Home Operators Developers Investors About Nokia
Company Financials Environment Community Research Venturing Press Careers Contacts
Research
Overview
Library
Demos
Conferences
Memberships
Research Projects
Submit Research
Cooperation Proposal
Careers in Research
Related Links
FAQ
Research Creates Success
Nokia Research is an information-sharing channel for researchers working in academia or the business world to promote cooperation.
An Introduction to TTCN-3
A recent book with contribution from Nokia researchers >>
Demo: The connected lifestyle
See what the future connected lifestyle could be like >>
Scientific Papers
Audio and Visual Communications
Communication Systems
Electronics
Mobile Networks
SW and Computing
User Interfaces
Search the library
Related Links
Openness
Forum Nokia
www.s60.com
Phone Technologies
Technologies for Business Solutions
Glossary (phones)
Glossary (digital TV)
Nokia Architecture Update
Nokia's vision on key architectural topics and domain interplay. Learn about the technical architectures driving digital convergence. >>
Home | Operators | Developers | Investors
| About Nokia
Club Nokia | Forum Nokia | Careers
Â© Nokia 2005. Site Terms |
Privacy Policy

benefit is that the page looks familiar and the position of content is easy to find on familiar pages. The drawback is that text columns are very hard to read if they extend beyond the screen width. If the page contains a lot of white space, the user may have to scroll through empty areas and the feeling of being lost is even stronger.

In Narrow layout, the content is formatted to fit the screen width (Figure 2). Although different browsers use a bit different algorithms to produce the Narrow layout, the basic rules are the same. The order of content follows the order it is presented in the markup file, the first piece of content on top of the following piece. Text is wrapped and large images are scaled down to the screen width. The benefit is that text is always easy to read, and the content is compact with not much white space. It is also straightforward just to scroll down the content. However, there are many drawbacks to Narrow layout.

First, content that should remain wide, such as maps and data tables, are often impossible to read in Narrow layout. This is because the text and other fine details in images become too small and distorted after scaling them down to fit the screen width, and the row-wise content of data tables, e.g. timetables, are flowed on top of each other, losing the meaning of the table.

Second, the user cannot navigate by location of the content, because one never knows where the content in the Original layout appears in the Narrow layout. It is also very hard to identify the main content on the page. While scrolling down the content, one needs to pay constant attention to all content scrolled by to identify the interesting piece of it.

Third, it is sometimes very hard for users to realize that after clicking a link the page really changes to another page. On most sites, the top content always includes the same data, such as logos and lists of navigation links. If the user selects one of the links on this list, the beginning of the next page looks exactly the same as the previous one, and often the previously selected link is on the list just like before. The user thinks he did not manage to activate the link and clicks it again, and again. It takes some time to understand that one needs to scroll down, past the links list, to see the content that has changed. For example in figure 2, the changed content becomes visible after scrolling down 3 screenfuls.

Fourth, dynamic Web content is becoming common, where client side scripting is used to modify the document. This trend will eventually make all viewing methods that significantly alter the original page structure unfeasible in real use.

Because of the problems of both Original and Narrow layout, the mobile browsers of today provide the user with both layouts and it is up to the user to decide which method works on which page. Most non-expert users do not know, however, how to control these different views, and it is also laborious for expert users to change the viewing mode.

VISUALIZATION METHOD

Based on our usability evaluations of mobile browsers [e.g. 14], we defined the following usability requirements for a Web page visualization method on mobile phones:

1. Fit more content to screen.
2. Eliminate the need for horizontal scrolling while reading a text column.
3. Provide enough context information to give an idea of page structure and to communicate the current location on the page.
4. Provide all basic functionality such as scrolling and link selection in a 5-way control key.
5. Do all this without destroying the original page layout.
6. Do all this without introducing modes.

To meet the above requirements, we provide a twofold solution that improves the viewing of Web content on a small screen. First, through a process called *layout scaling*, we apply two changes to the CSS (Cascading Style Sheets) formatting model, essentially modifying the size of the text relative to the rest of the page contents and limiting the maximum width of the text paragraphs to the width of the browser viewport. We then render a scaled down version (called *overview*) of the Web page with an indication of the current viewport and we overlay it transparently on top of the browser viewport. This overview is meant to be primarily a navigational aid, giving the user more contexts by allowing her to visualize the current position of the viewport inside the document. It also helps the user locate information inside the page.

We will describe these visualization solutions in detail in the following subsections.

Layout scaling method

Document rendering in a modern Web browser is based on the CSS2 formatting model [18]. The formatting function takes as an input various internal constraints imposed by the structure and style of the document being processed and external constraints imposed by the browser application and environment.

Internal constraints often limit the minimum width of the boxes that make up the generated layout. For example, if a box contains an image, it cannot be narrower than the image. Similarly, if the document sets a table column to some fixed pixel width, it cannot become any narrower or wider than the specified value. Ignoring any of these constraints is violation of the formatting model and will distort or destroy the page layout.

The most important external constraint for the formatting function is the width of the browser viewport (on desktop browser the width of the browser window, on mobile browser the width of the screen, minus UI elements such as scroll bars). The formatting process tries to make the document width match the viewport width while still satisfying all the internal constraints. If that is not possible then the document becomes wider (or narrower) than the viewport and viewing the entire document might require horizontal scrolling.

The text content of a layout box is formatted after the box width is determined. Still the text content provides constraints to the box width. The minimum width of a box containing text is the width of the longest word in the box. The box height, and eventually the total height of the document, is determined by the formatted height of the textual content. Width and height of individual pieces of text depends on the font used and must be known during the formatting process.

This way of content formatting incurs at least the following two usability problems: the general amount of content that fits in the viewport is small (because, for example, many of the images are unnecessarily large) and the text paragraphs are often wider than the viewport width, requiring the user to scroll horizontally when reading. These problems can be observed in Figure 4. The size of the viewport here is 176 pixels wide and 208 pixels high**.**

We propose a document layout scaling algorithm that addresses the two problems mentioned above. This algorithm applies a set of modifications to the normal CSS formatting and painting process of the browser, so that the sizes of the non-textual page elements become smaller (thus fitting more content in the viewport), while ensuring that the widths of the text runs are never larger than the width of the viewport.

Input: current viewport size, scaling factor

Output: a bitmap representation of the document

Algorithm:

1. The viewport width and height constraints are multiplied by a scaling factor.
2. During the formatting step, text metrics (calculated width and height of the text strings) are multiplied by the scaling factor. The formatting is done otherwise normally except that all constraints that depend on text metrics are calculated using the virtual enlarged fonts. For example, if the scaling factor is 2, a text string which measured a width of 150px and a height of 12px is treated as a string of a width of 300px and a height of 24px.
3. If the calculated width of a text paragraph box is wider than the viewport width, multiplied by the scaling factor, then we format the contained inline text exactly to the viewport width.
4. During the painting step, all coordinates and sizes calculated by the formatting step are divided by the scaling factor. These scaled-down coordinates and sizes are used for painting to the output device. For example, if the scaling factor is 2, an image of size 100x100 px at formatted coordinate (0,300px) will be drawn as image of 50x50 px at coordinate (0,150px). Since font sizes were first scaled up during formatting and are now scaled down, the text is actually drawn using the original font size.

In the generated bitmap of the document, images and generally all elements with static sizes (images, HTML

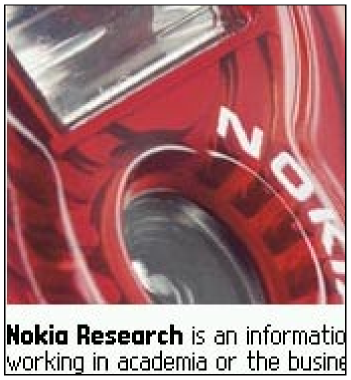
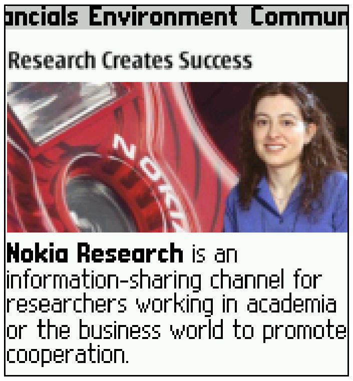

Nokia Research is an informatio
working in academia or the busine

Figure 4. Original layout view on a phone browser.


ncials Environment Commun

Research Creates Success

Nokia Research is an information-sharing channel for researchers working in academia or the business world to promote cooperation.

Figure 5. The page view after layout scaling has been applied.

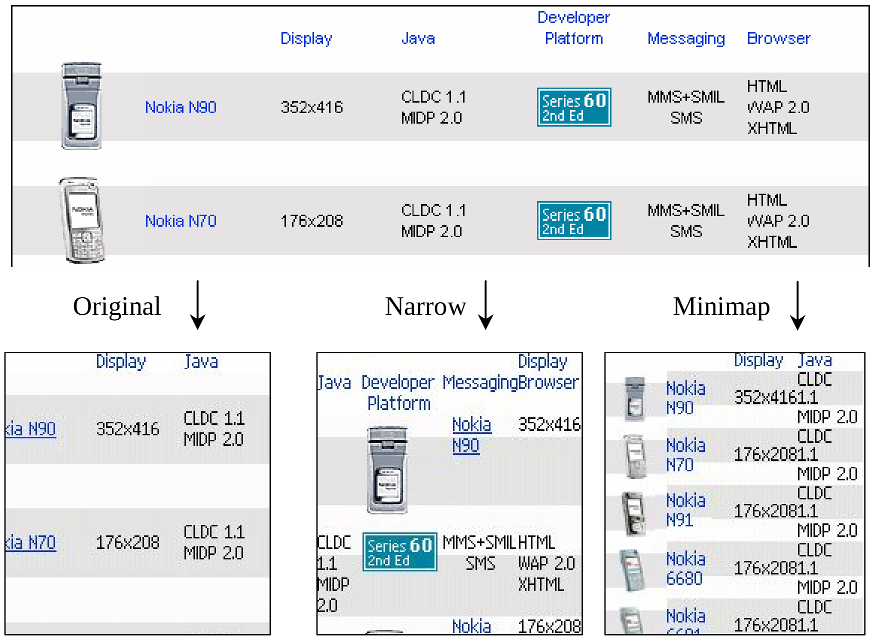

		Display	Java	Developer Platform	Messaging	Browser
	Nokia N90	352x416	CLDC 1.1 MIDP 2.0	Series 60 2nd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML
	Nokia N70	176x208	CLDC 1.1 MIDP 2.0	Series 60 2nd Ed	MMS+SMIL SMS	HTML WAP 2.0 XHTML

Figure 6. A data table viewed on a small screen with different visualization methods: Original layout, Narrow layout, and Minimap view.

tables…) become smaller. For example, a scaling factor of 2 halves the width and height of all images. The font size remains constant to ensure readability. Another important effect of layout scaling is that the maximum width of text paragraphs is equal to the width of the viewport. This effectively eliminates the need for horizontal scrolling during reading. Figure 5 depicts the same page as in Figure 4, this time rendered with the layout scaling algorithm.

We have observed that the vertical size of the rendered documents may slightly increase as a result of layout scaling, since less text fits on a single line. However, often the overall area of the document actually decreases because of the smaller size of non-textual content (Figure 3).

Large data tables are particularly challenging for small screens, yet tabular information such as timetables, schedules, or stock prices is highly relevant for mobile use. The current research tackling this problem counts on a pointing device [19], which mobile phones typically do not provide. In Minimap solution, we aim to preserve the table formatting as well as possible (Figure 6), but making all types of tables work nicely on a small screen is a true challenge. It is very hard to differentiate tables containing tabular data from tables purely for layout, so we handle all tables in the same way.

The page overview

To help the user navigate a Web page, we provide her/him with a scaled down overview of the page. This novel feature, Mini Map™, inspired the naming of the whole method. The Mini Map view is chosen to contain all the content currently visible in the browser viewport, plus significant parts of surrounding areas. Figure 7 shows an example of the browser displaying the overview of a page on top of the viewport. The red rectangle corresponds to the current location at a given moment.

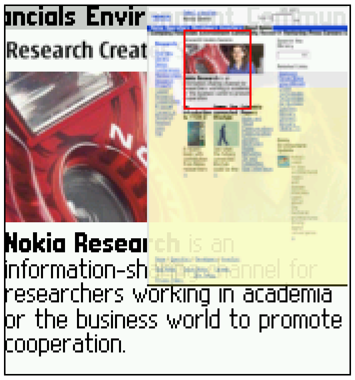


Figure 7. Mini Map overlaid transparently over the viewport.

To prevent the overview from becoming too intrusive, we make it transparent by using alpha blending. The user may adjust the opacity level through the browser preferences. Furthermore, and most importantly, the overview becomes visible only when the user is scrolling the document. Currently, we distinguish between two types of scrolling:

1. Incremental, when the user is scrolling the document in separate steps, generating one controller input (e.g. clicking a scroll key on a phone keypad) for each step.
2. Continuous, when the user is scrolling the document as fast as possible, without releasing the controller (e.g. keeping the scroll key pressed).

The overview is visible only during continuous scrolling. When the user releases the controller, thereby stopping the scrolling, the overview remains on the screen for approximately one second and then fades away.

A red rectangle matching the portion of the document visible in the viewport is drawn on top of the overview. This rectangle is moved accordingly when the user scrolls the viewport. The portion of document shown in the overview is scrolled as needed so that the area of the document visible in the viewport, plus additional surrounding areas, stays visible in the overview.

A slight yellow tint is applied to the overview for a newly formatted document. This coloring is removed for those areas of the document that the user has already seen in the viewport. This mechanism helps a user determine which parts of the document s/he has already visited, which is particularly useful when trying to locate information on a large page.

USER STUDY

We conducted two usability studies to compare our Minimap method and Narrow layout method. The first one was a traditional usability test in a laboratory with 8 subjects, after which we further developed the Minimap prototype. The second was a longer-term field study where 20 participants used Minimap and a Narrow layout browser for 8 days each. In this paper, we concentrate in the second test, but a short summary of the laboratory test is presented below.

In the first Minimap prototype, we had the page overview coming up whenever the user started to scroll the page. Participants complained that Mini Map was covering the actual content and users could not read a longer text block on the page. Since the Mini Map did help the users to navigate on the page, we wanted to find a way to show it less obtrusively. The first option was to provide Mini Map behind a key, so that Mini Map would be shown while the user keeps this key down. The previous experience with hidden key shortcuts has not been encouraging, however, and we wanted novice users to find the Mini Map without learning any key shortcuts. The solution was to detect when the user is likely to be reading, when scrolling a longer way. In the second prototype, we showed Mini Map only when the user keeps the scrolling key down for more than one second, not when s/he scrolls incrementally click by click. The latter is typical behavior while reading.

In both studies, we concentrated on the method of viewing and navigating on Web pages, not so much in the other tools like bookmarks or technical abilities to show the various types of content on Web pages.

Participants

We used 20 subjects with various ages and backgrounds. 12 subjects were male and 8 female, ages 15-50 years with a mean of 30. We divided the participants into two groups of 10 participants each, so that each group had a similar distribution of ages, genders, and backgrounds.

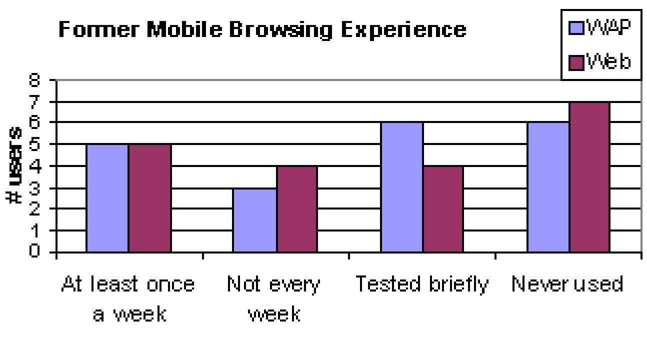


Figure 8. Former browsing experience on a mobile device.

Seven participants had never viewed full Web pages on a mobile device before, and five participants were frequent users of full Web browser on a mobile phone (Figure 8). The participants were paid a small reward after the test period. They did not have to pay the browsing costs during the test period. We did not reveal the origin of either browser during the study, but unfortunately, it was clear that the Minimap browser was a prototype version.

Procedure

Group 1 used Minimap browser first and switched to Narrow layout browser after 8 days. Group 2 used the browsers in the opposite order. The participants were not given instructions on how to use the browser, but only how to copy and paste the URL used in the task to the browser. By not guiding users, we wanted to simulate the situation that real users face when taking a mobile phone with a Web browser into use. With the Narrow layout browser, we instructed the users in link selection, because such a feature should not affect the final rating.

The participants used Nokia 6600 phone for browsing over GPRS (General Packet Radio Service) connection. The display of this Series 60 style phone model is 176x208 pixels, and the main input control is a joystick that can be tilted in 4 directions and pressed for selecting. There are two soft keys, one for Options and the other for Back (or a similar function).

We sent one or two tasks to the participants by text message every morning. Together with the message, we sent 2-4 multiple-choice questions, which they had to answer before the next morning. Below, you can find an example of a task.

Check the main headline of the day from news.bbc.co.uk. Then check what news from Europe the AROUND THE WORLD section provides today.

a) Did you know the pages beforehand?
1=Yes, from PC 2=Yes, from a mobile browser 3=No
b) How easy it was to locate the needed information on the page? 1=Very hard .. 5=Very easy
c) How certain you felt about finding the information needed? 1=I lost my faith .. 5=100% certain

We selected 12 goal-oriented tasks for each period that access many different types of Web pages, both textual and graphical, simple and crowded, with and without data tables, small and large images, images containing detailed information (e.g. text), light and dark background colors, and pages with different number of content columns and page structures. We tried to select tasks that would be somehow relevant for mobile use, so most pages were national ones and the content was fresh.

The tasks for the first 8 days were the same for both groups, to make sure the tasks were equally demanding. We used mostly the same tasks also for the second 8-day period to allow for a comparison of the two browsers. In three tasks, however, we used different Web sites, because we wanted the users to navigate on some unfamiliar pages or find information whose location they did not know with both browsers. If the tasks were exactly the same during both periods, the participants would not have been able to compare how the browser behaved with pages that you do not know beforehand.

In addition to the daily task feedback, the users were asked to keep a diary about their experiences with the mobile browser during the test period. This was in order to gather their insights during the whole test period as well as experiences about their own browsing cases outside the given tasks. They were encouraged to use the browser in places that they would normally use it, but many ended up executing the tasks at home. In the Minimap browser, a log file recorded the functions that the user selected in the browser, which provided us information about the most frequently used functions.

After testing each browser, we discussed the experiences the participants had with the browser in a 2-hour group session. A rating questionnaire was filled-in at the beginning of these sessions.

RESULTS & DISCUSSION

As described in the previous section, we collected various types of feedback from the study participants. In the following sections, we show the results, from the main findings to more detailed ones, and discuss the implications.

Visualization method preference

After using both browsers for 8 days each, we asked the participants to rate which browser they prefer to use for viewing Web content on a mobile phone. We used a 7-point scale, 3 meaning strong preference for either browser and 0 meaning no preference. 18 participants preferred Minimap,

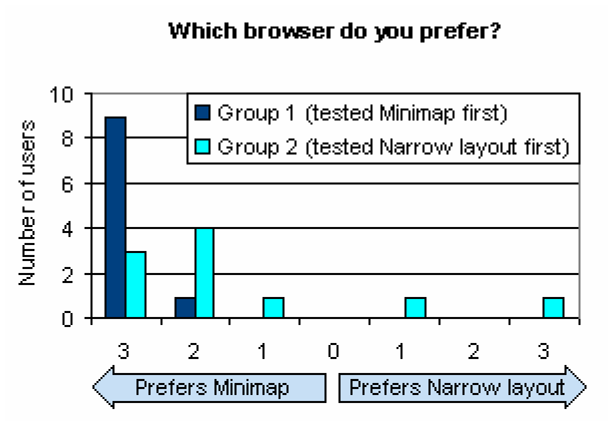


Figure 9. 18 participants preferred Minimap browser, Group 1 more clearly than Group 2.

while 2 users liked Narrow layout browser better. Usually, it is rare to get participants give strong preference ratings, so it is notable that as many as twelve out of the twenty participants used the extreme preference rating for Minimap (Figure 9).

The previous mobile browsing experience did not affect the rating, since all experience backgrounds can be found along the given ratings. However, the order of testing the two browsers clearly affected the rating so that the first browser got the user's preference more easily. All users who first used Minimap browser clearly preferred it, whereas the preference distribution of the other group varied more. The most probable reason is that in the beginning, one is more motivated to learn how to use a mobile browser, but the charm of novelty is gone by the time one has to learn another way of navigating on the large pages. Once you feel you can control one browser, you have little motivation to relearn a new one. Still, eight out of ten participants of group 2 preferred Minimap, although they had to relearn the viewing method.

The two users who preferred Narrow layout commented that they found it easy just to scroll down the content, and eventually the right piece of content will show up.

Why so many participants preferred the Minimap method? Examination of the overall ratings and the task usability ratings will help us to understand the reasons in the next sections.

Overall ratings

We used a two-tailed T-test, with alpha = 0.05, to analyze the statistical significance of the differences between the means of the observed variables.

The results of the overall ratings (Figure 10) show that participants felt Minimap is clearly easier to take into use (p=0.035), and significantly easier to use after some time than the Narrow layout browser (p=0.023). According to the discussions in the feedback session, the reasons were that pages look more familiar on Minimap browser, and the

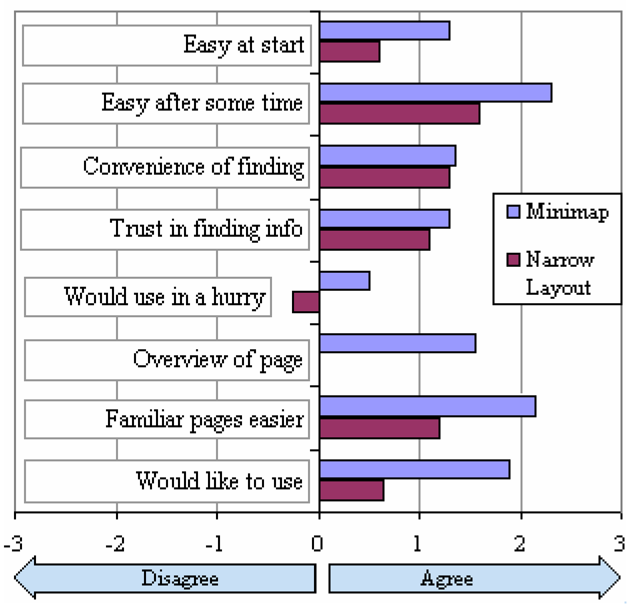


Figure 10. Overall ratings for the browsers show preference on Minimap method.

page overview helps to keep track of the page structure and the current location.

The results for convenience and trust in finding the needed information on the pages did not show statistically significant differences in this subjective rating.

The participants did not see that either of the browsers would be suitable to be used in a hurry, which shows that a small screen makes it hard to just glance at the page and spot the needed content. The GPRS connection is also not fast enough for loading full Web pages in a hurry. The Narrow layout browser scored lower than Minimap (p=0.118, not significant) because it requires more concentration on following the content being scrolled through. Especially timetables are very hard to interpret in a hurry if the rows in a data table flow onto several lines.

The significant difference (p=0.003) in getting the idea on where different types of content are located on the page ("Overview of page") was not surprising, because Narrow layout does neither preserve the layout nor show an overview for the page.

It is very clear from the ratings that on these small screen browsers, familiar pages are easier to use than unfamiliar ones. On the Narrow layout, participants did not see the benefit as big as on Minimap (p=0.012), because you have to follow the content being scrolled through more carefully than in Minimap where you just place the viewport to the location where you know the content can be found.

The last question was about participants' willingness to use this browser for viewing Web pages if they need to access the Web with a mobile phone. Again, there was a significant difference in favor of Minimap (p=0.011).

Participants expressed their surprise on how well they could view the large Web pages on a small screen.

Task-based ratings

After executing one of the daily tasks, participants gave their ratings about how easy it was to locate the needed information on pages, and about task specific usability factors, such as reading text in an image or interpreting the data in a table (Figure 11). A sample rating question was presented in the Procedure chapter.

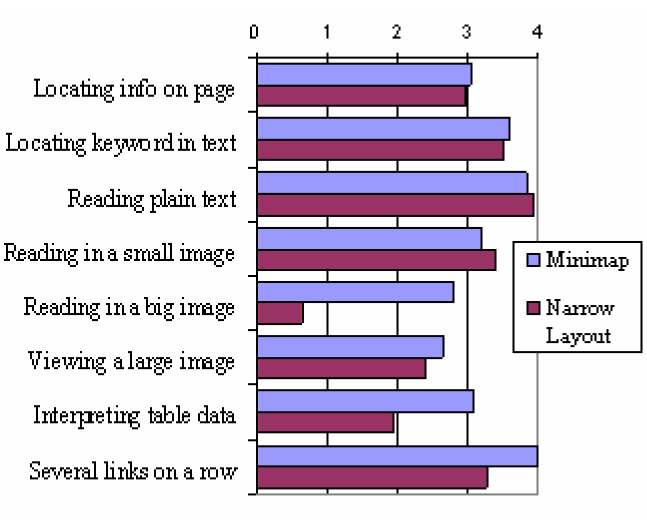


Figure 11. Average usability ratings after executing daily tasks.

The most interesting usability factor in using large Web pages on small screens is how to locate the information needed on the pages. All in all, both viewing methods functioned relatively well here, the scores being almost equal. There were more differences, however, between single tasks. Minimap scored better on pages with big data tables or relatively simple layout, whereas Narrow layout scored better on large crowded pages where the needed information was located near the bottom of the page. With both methods, locating the information was the easier the closer to the top the needed information was located.

Reading plain text was very easy with both browsers. Our hypothesis was that Minimap would have scored worse here, because in Narrow layout, plain text is normally wrapped to the screen width whereas in Minimap method, several columns of text may be visible at the same time. These results do not support our hypothesis, however, because both the ratings and the feedback discussions show that participants saw text reading very easy also with Minimap method. The difference between Minimap and Narrow layout methods is not statistically significant.

In both methods, images are scaled down to better fit the small screen. Downscaling makes it easy to view images at one glance, and to scroll over unimportant images. The downside of scaling is that text and other detailed information in the images may become too small to see. The results show that Narrow Layout has severe problems with large images, whereas Minimap meets this challenge relatively well. The difference in ratings is statistically very significant. One reason is that Narrow layout forces even large images to fit the screen, but in Minimap, images are not scaled down more than 50%. Second, when the participants wanted to see the image in bigger size, they zoomed in the view. In Minimap, the image becomes bigger, but Narrow layout forced the large image to fit the screen width even after zooming in. This was a very irritating feature for the users. Example tasks were to read the Dilbert comic strip of the day and interpret a map. To accomplish the task, the user would be required to change the layout mode from Narrow to Original, but 10 users out of 20 did not find the way to accomplish these tasks during the 8-day test period.

Another clear problem with Narrow layout was the way it shows data tables that are wider than the screen, such as a table with TV programs or a table with public transportation timetable. Participants found it hard to interpret this data, because Narrow layout method wraps the rows onto several lines, and interpreting which information is related to which is very hard (Figure 6). One participant commented in his diary: “If my life was dependent on this data, I would be able to interpret the table, but now, I do not have the motivation.” The solution here would have been to change the viewing mode from Narrow to Original, but as we noted, switching the mode was a feature that only half of the participants found. Large data tables were hard also on Minimap browser, because the column/row headers were often outside the view, but the tables were still significantly easier to interpret than in Narrow layout.

The last rating for daily tasks shows that moving between several links on one row was very easy in Minimap, but clearly harder in Narrow layout. This problem might not be specific to the Narrow layout method itself, but to the browser implementation. We used a Narrow layout browser that focuses links only when the scroll key is used vertically. Horizontal movement is used for speed scrolling: one click scrolls the view down/up almost one screen. This means that when the user wants the link focus to move sideways, s/he should not scroll sideways but vertically. This irritated many users, and even expert users made mistakes in selecting links on one line. Although the content itself does not require horizontal scrolling, it should be possible to move the focus between links on the same row by horizontal move.

Page overview

We did not give any instructions for participants on what kind of features there are in the tested browsers, but encouraged them to explore the functionality. It was interesting to see if they would use continuous scrolling, thereby discovering the Mini Map overview. If some users scroll the pages incrementally (click-by-click) and do not hold the scrolling key down, they will never see the page overview while scrolling. We wanted to know if the Mini Map overview was discovered, so we asked the users on 3rd day whether or not they have found the page overview feature and when they found it.

14 users answered they found the page overview during the first browsing session, and the rest had found it during the first day. This shows that at least when the page content stretches over several screens, users switch from incremental scrolling to continuous scrolling spontaneously. So, according to this study, it is highly likely that users will find the page overview easily without any guidance.

In the feedback discussions, we asked whether the page overview disturbed the participants. The outcome from the discussions was that when glancing though the content on the page, the page overview might be disturbing. However, the pros of the overview overshadowed the cons. For expert users, a best user interface would probably be one where the overview comes up only by pressing a shortcut key.

In our latest study, we have tested a version where the page overview does not appear while scrolling but only when a shortcut key is pressed. We used 20 participants with more experienced mobile browsing backgrounds, but nobody was familiar with the Minimap method beforehand. During a 1.5-hour test session, the participants executed several Web browsing tasks both on this new version without Mini Map overview and on a Narrow layout browser. Most tasks were executed without checking the page overview. We got the same preference result for this study as for the long-term study described in more detail in this paper: 18 participants preferred Minimap, 2 participants Narrow layout. This shows that Original layout with a few formatting changes works better than the current state-of-the-art methods, even without using the page overview.

CONCLUSION

The current state-of-the-art mobile phone Web browsers provide both Narrow and Original layout, and the user should decide which layout works for which content. The layout mode may have to be changed even on the same page: wide text columns can be read only in Narrow layout, but if there is a wide data table or a large map on the same page, one has to switch the mode to Original layout. We have found in various user studies that most users do not know about the existence of the two viewing modes, but they try to zoom the view, and give up when it does not help.

Our Minimap method aimed at solving the Web viewing problems on mobile phones with a modeless solution: all pages are viewable in the same mode, and the worst cases are solvable by having the user zoom the content. The keyhole-viewing problem is alleviated by showing a page overview when the user scrolls the page continuously.

We conducted a long-term field study with 20 participants to find out which Web page viewing method is preferred on mobile phones: the state-of-the-art method, or the new Minimap method. The results show that 18 out of 20 participants preferred the Minimap method, 12 of them strongly. It seems that users like the similar Web page layout as they have seen on a PC, if the usability problems of Original layout on small screens have been addressed. The Minimap method has successfully solved the problems by condensing the page layout to better fit onto the small screen, by forcing all text columns to the maximum width of the screen, and by showing an overview of the page when the user scrolls a longer distance on the page.

Although the Web browsers on mobile phones aim at handling all existing Web content in a usable way, it helps if the site developers take small screens into account when designing their sites. If a site provides relatively light pages without small details in images and with few large components, people will also be able to access the site with their mobile phones without major usability problems.

ACKNOWLEDGMENTS

We want to thank Barbara Hammond, Thomas Stovicek, Guido Grassel, Franklin Davis, and Roland Geisler for their support in authoring this paper. Nokia S60 devices coming out spring 2006 will provide a browser based on the Minimap visualization method.

REFERENCES

1. Baudisch P., Xie X., Wang C., Ma W.: Collapse-to-Zoom: Viewing Web Pages on Small Screen Devices by Interactively Removing Irrelevant Content. *Proc. ACM UIST 2004.*
2. Björk, S., Holmquist, L.E., Redström, J., Bretan, I., Danielsson, R., Karlgren, J., Franzén, K. WEST: a Web Browser for Small Terminals. *Proc. ACM UIST 1999*.
3. Buyukkokten O., Garcia-Molina H., Paepcke A., Winograd T.: Power Browser: Efficient Web Browsing for PDAs. *Proc. ACM CHI 2000*.
4. Buyukkokten, O., Kaljuvee, O., Garcia-Molina, H., Paepcke, A., Winograd, T.: Efficient Web Browsing on Handheld Devices Using Page and Form Summarization. *ACM Transactions on Information Systems*, Vol. 20, No. 1, January 2002, p. 82–115.
5. Chen Y., Ma W., Zhang H.: Detecting Web Page Structure for Adaptive Viewing on Small Form Factor Devices. *Proc. WWW 2003*.
6. de Bruijn O., Spence R., Chong M. Y.: RSVP Browser: Web Browsing on Small Screen Devices. *Personal and Ubiquitous Computing* (2002) 6:245–252.
7. Fulk, M.: Improving Web Browsing on Handheld Devices. *Proc. ACM CHI 2001*.
8. Furnas, G. Generalized Fisheye Views. *Proc. ACM CHI 1986*, p. 16–23.
9. Igarashi, T., Hinckley, K. Speed-dependent Automatic Zooming for Browsing Large Documents. *Proc. ACM UIST 2000*, p. 139-148.

10. Lam H., Baudisch P.: Summary Thumbnails: Readable Overviews for Small Screen Web Browsers. *Proc. ACM CHI 2005*, p. 681-690.

11. Lieberman, H.: A Multiscale, Multilayer Translucent Virtual space. *Proc. IEEE Information Visualization 1997*, p. 124-131.

12. MacKay B.: The Gateway: A Navigation Technique for Migrating to Small Screens. *Proc. ACM CHI 2003*.

13. Milic-Frayling, N., Sommerer, R.: SmartView: Flexible Viewing of Web Page Contents. *Poster paper at WWW 2002* (http://www2002.org/CDROM/poster/172/).

14. Roto, V., Kaikkonen, A.: Perception of Narrow Web Pages on a Mobile Phone. *Proc. Human Factors in Telecommunications 2003.*

15. Spence, R.: Information Visualization. ACM Press (2001), p. 111-133.

16. Thunderhawk. http://www.bitstream.com/wireless.

17. Trevor J., Hilbert D., Schilit B., Koh T.: From Desktop to Phonetop: A UI For Web Interaction On Very Small Devices. *Proc. ACM symposium on User interface software and technology table of contents, 2001*. p. 121-130.

18. W3C: Cascading Style Sheets, level 2 Specification, W3C Recommendation, http://www.w3.org/TR/REC-CSS2/, 1998.

19. Watters, C., Zhang, R., Duffy, J.: Comparing Table Views for Small Devices. *Proc. ACM Symposium on Applied Computing (SAC) 2005*.

20. Wobbrock J., Forlizzi J., Hudson S., Myers B.: WebThumb: Interaction Techniques for Small-Screen Browsers. *Proc. ACM UIST 2002*.

21. Yin X., Lee W.: Using Link Analysis to Improve Layout on Mobile Devices. *Proc. WWW 2004*.