Mobile Browsing: Evaluating and Improving an Existing Site

Version 1.0; July 9, 2007

The Mobile Browsing series continues with this article, which takes a look at several case examples of Web design that are **not** mobile-friendly. The problems illustrated are critical issues for mobile users; a site that violates even one of these items is likely to cause the user to leave the site and hesitate to return.

The Nokia Web Browser renders everything in the same way that would be expected on a desktop. However, by making adjustments that take the mobile user into account, the experience can actually be improved for both regular and mobile visitors. Many of the suggestions outlined in this article apply equally to desktop browser performance and, as such, might simply be considered efficient site design.

Developers should take a critical look at their existing site to see if some of the issues presented here apply to them, and then do something about it! Developing mobile-friendly sites is not rocket science, but it does require a new way of thinking and designing for a special group of users. The best way to evaluate a site from the user's perspective is to get a mobile device with a full Web browser and use the site for a week. Does it make the user want to return?

This article uses two site design examples to illustrate the issues discussed here: one shows the problems and the other provides a solution to the problems. Developers should refer to the examples as they read the article.

- Example of problematic design (opens as a popup)
- Improved design (opens as a popup)

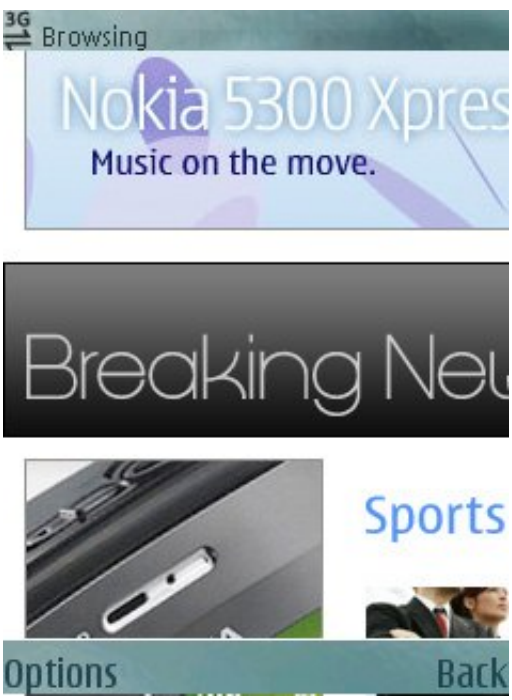
Problem 1: Ignoring the "hot spot," and poor navigation

The most important section of the site is located in the top left corner of the landing page — the so-called “hot spot.” On a desktop browser, the user is able to quickly get an overall picture of the entire site at a glance. It is much more difficult to provide a mental image of the site on a mobile device that does not display the entire page at the same time.

In the problem example, the hot spot does not have the right content. The user is left with little idea of the purpose of the site, and s/he has to search for the navigation aids. The hot spot is filled with nonfunctional content: ad banners take up valuable space.

The navigation of the site is placed on the right-hand side of the page, hidden from the initial view. Also, the navigation is likely to be implemented with images that have to be separately downloaded. Because some users save loading time and bandwidth by turning off images, the failure to use alternative text descriptions for important UI elements may render the navigation completely invisible.

In the improved design for the same content, the navigation is moved to the upper left corner to fully utilize the hot spot. The banners are moved to the right side of the page — visibility of the banners is important, but it should not compromise the usability of the site. Instead of using images, the navigation is built with CSS and text elements, resulting in smaller download rates and ensuring that the navigation is always visible.



The main navigation is boosted with a mobile navigation tool (introduced in the Nokia Browser Design Guide series) for quick access to site content. Additional "Back to Top" links have been added to the bottom of the content areas to help users get back to the beginning of the page without scrolling.

Problem 2: Scalability

Scalability — adjusting to the size of the available browser window — in Web design is an important factor. Scalable Web pages make the user's life easier by removing the need for unnecessary scrolling and by making the best possible use of the available display space.

Mobile use adds a wrinkle to the issue of scalability: fitting the page to the mobile browser display is no longer the best possible option. Attempting to create a fully scalable page is very likely to result in a nonusable and extremely cramped layout:

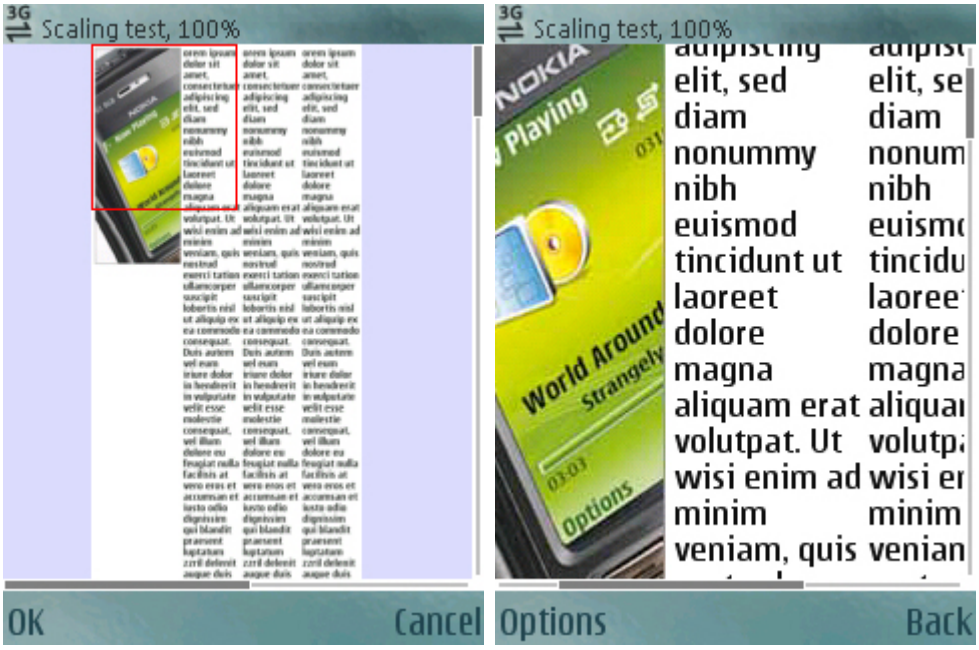


Figure 1: A page overview and a close-up of a cramped page set to full scalability

One solution is to define a range of minimum widths for the page columns. On a desktop browser, this is not likely to result in any problem, but on the mobile browser it can make a world of difference.

In the following example, the minimum width of the table element has been set to avoid cramping the entire view. The min-width property is declared in the style rule:

```
<table style="min-width: 100%; width: 800px">
```

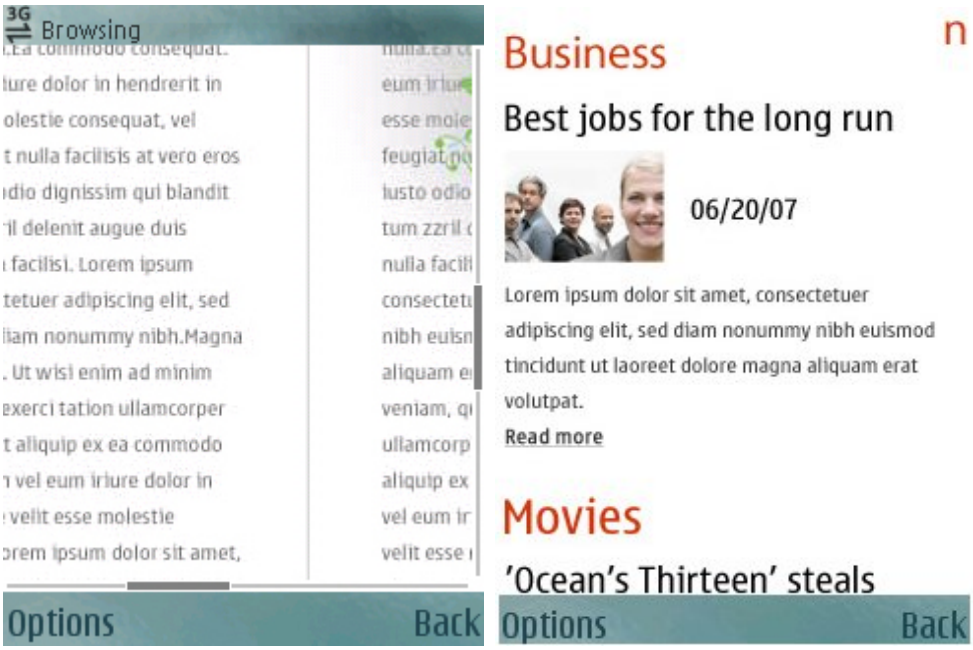
Figure 2: A page overview of a table that has a set minimum width

Problem 3: Content overload causes frustrating scrolling

With a mobile browser, navigating to the content that the user wants is more difficult than on a desktop, where s/he can “close in” on the sections of the page with a simple eye movement. On a limited display, users may have to scroll across the page to get to the content they're interest in.

When a page is overloaded with nonrelevant content, such as a news piece that the user is not interested in, scrolling through the page becomes a burden. Also, while scrolling through vast quantities of text, it is easy to get lost within the page on a small display.

A good solution is to display a brief introduction to a certain topic on the page and allow the user to dig in deeper for the content s/he finds interesting. This can be achieved with, for example, “Read more” links that either go to a separate subpage within the site or expand the text area within the same page. An expanding text area can be implemented with JavaScript™ code; for an example, see the Nokia Browser Design Guide, Article 4, in the mobile browsing series.



Also pay attention to the size of the graphics and banners on a page. A common mobile device display is QVGA (240x320 pixels). If necessary, consider using an alternative page layout or reduced graphics for browser clients with small displays. For examples, see the Nokia Browser Design Guide, Article 4.

Problem 4: Heavy pages and download times

Even though mobile bandwidths and downlink speeds are improving, not everyone has a mobile broadband riding around in their pocket — mobile browsers access the Internet with connections ranging from general packet radio service (GPRS) to WLAN and HSDPA. A heavy page with a lot of images and content can take a long time to load and render on a mobile browser. An additional issue is the mobile data pricing rates. For users with flat-fee pricing, this is not a problem, but a page size closing in on several hundreds of kilobytes or more is a shock for a pay-per-use subscriber.

Try to test how many kilobytes the site’s front page adds up to when loaded fully. Try to keep the download size of the main page down to 100 kilobytes — that should be enough, unless the site is dedicated to showing off high-quality images. Allow the user to dig deeper from the main page and subsequently download only the content s/he wants.

In addition to the bandwidth issue, network latency slows down page rendering. If the Web page is composed of a multitude of individual files (such as images and scripts), they all need to be individually downloaded and rendered. As a solution, make sure to reuse and minimize the number of existing files: use one CSS style sheet instead of several files, and as few script files as possible. Instead of using image files, consider achieving the same result with simple text laced with CSS.

Tips and tricks

Here's a collection of handy tips and tricks related to different fields of page design:

Graphics:

- Optimize graphics to the smallest feasible file size and physical size on the page.
- Use the correct graphics format: JPG for photographic images, GIF or PNG for site graphics.
- Replace graphics with CSS, if possible.
- Remember to use sufficient color contrast.
- Do not assume the user will see the graphics; include alternative text descriptions for images.

Flash and sound:

- Consider using Flash Lite instead of full Flash; most of the same content can be created with Flash Lite, and this will increasingly mean that mobile browsers can view the flash content.
- Avoid Flash altogether in critical page components, such as the navigation or layout.
- Do not use sound on a normal page. Sound causes additional downloads and irritates the user. If sound is used, make it optional.

JavaScript:

- Never rely solely on JavaScript; use JavaScript only to enhance the browsing experience. The basic structure of the page should be built on pure HTML.
- Too much code slows the browser.
- Avoid JavaScript in forms and drop-down menus.
- Check that JavaScript does not render the browser's Back button ineffective.