

Designing Web Pages for Handheld Devices

Karen Tusack, Instructional Technology Consultant
University of Wisconsin-Madison

The opportunity to engage in interactive “learning on demand” is quickly becoming a reality with the advent of wireless networking and portable wireless devices such as PDAs (Personal Digital Assistants), the ubiquitous cell phone, and the recent merging of the two in the form of the “smartphone”. Very soon the ability to surf the internet and communicate with the world will be in the palm of our hands every waking moment of the day, and for educators it is an opportunity not to be missed. The technology for this development is changing rapidly: what I write here will be hopelessly out of date in six months. Rather than wait until the dust settles, start designing your web pages with handheld wireless devices in mind now, to take advantage of that elusive “anywhere, anytime” learning opportunity.

Alphabet Soup

Desktop and laptop computers use IP (Internet Protocol) for delivering information from web servers, and HTML (HyperText Markup Language) to display those materials. IP/HTML capabilities of the first handheld devices measured from poor to non-existent, which prompted entirely new markup languages: WML (Wireless Markup Language), HDML (Handheld Device Markup Language), and WBMP (Wireless BitMaP), as well as a new protocol: (WAP Wireless Access Protocol). WML is a subset of standard HTML, as is HDML. WAP is a protocol that allows servers to deliver web materials in a manner that will not overwhelm handheld devices. WBMP is a part of WAP, a compressed form of graphics files optimized for mobile computing devices. Current wisdom holds that the future of the web lies in XML (eXtensible Markup Language) or XHTML (eXtensible HTML) and CSS (Cascading Style Sheets), separating the form from the content of web sites, and will allow one to display any given information in any format.

Handheld Limitations

1. *Screen size and resolution.* PocketPC devices display 176 X 220 to 320 X 320 pixels. Palm OS devices display 150 X 140 to 240 X 320 pixels. Keep in mind that this is the **total** screen size, and that some of it will be used for the devices own tools bars and scroll bars. The typical cell phone displays 96 X 96 pixels, or 4 lines of 15 characters each. Research has shown that users will still be able to accomplish tasks in a reasonable amount of time even if they have to scroll vertically more than on a desktop, but scrolling horizontally. (Jones et al., 1999, p. 3).
2. *Handheld browsers.* On PocketPC devices, the current built-in browser is Mobile Internet Explorer version 2003, which is much more capable than the previous Pocket IE version 2002. On Palm devices, it is mostly Web Pro, Netfront, Blazer, or AvantGo, or numerous proprietary browsers that ISPs distribute to customers when they sign up for wireless service. Opera or customized versions of browsing software are usually shipped with devices run on the Symbian OS (smartphones, for example). The latest browsers for PocketPC and Palm devices not only support most HTML 3.2 standards, but also some HTML 4.1 features, such as XML, CSS, and SSL (Secure Socket Layer), but these are nowhere near as numerous as older devices and operating systems in use today, so keep that in mind when designing.
3. *Device input* ranges from handwriting recognition systems, on-screen keyboards, and "thumb" keyboards to add-on full-sized keyboards (not to mention the soon to come “virtual keyboard“for handhelds!). None of them is an easy solution: keep user input to a minimum when you can.

“Sherman Tank” Solutions to Designing Your Web Page

If you want your web materials to be fully accessible on a handheld and expect the main users of your site to be handheld users, you can hire a programmer to develop a complete WML site that parallels the information on your web site. Or, you can use special software to perform server-side HTML to WML on-the-fly transcoding, use software such as Argogroup’s WAPTool to convert HTML pages to WML, or transducing software which splits a web page into multiple sub-pages (each better suited for the smaller display) and adds navigation links to navigate around the sub-pages. (Trevor, 122). On the other hand, one could rely on the users to find software solutions, such as web-clipping software, which downloads selected web sites to a handheld via a desktop computer, adjusting the HTML for the capabilities of that particular handheld device. Far from elegant, these solutions may disappear as handheld browsers improve and smartphones take the place of text-only cell phones.

The “Ideal” Solution

As the multitude and variety of web-enabled devices increases (yes, you may someday soon be reading the morning newspaper on your toaster) and as XML (eXtensible Markup Language) and CSS (Cascading Style Sheets) gain in popularity, it is becoming evident that the ideal solution is to separate content from form, so that one need only record information once and use style sheets to customize the way it is displayed on an infinite variety of devices. This sounds like the ideal solution, but is a little ahead of the game: most PDAs and few, if any, cell phones support XML or CSS at this time. The learning curve for using XML with CSS is also quite steep, and educators rarely have the opportunity to hire programmers to design their web pages. I’ve no doubt that if XML and CSS are the perfect solution there soon will be software available to make authoring XML/CSS web sites as easy as creating standard HTML sites, without the need to understand the underlying code (Dreamweaver, FrontPage, etc.).

The “Flyswatter” Solution

A much more affordable, practical, and pragmatic approach to the problem is available that does not require a programmer, learning a new markup language or the inefficiency of maintaining separate sites for desktop and handheld users. It is a design method which merely keeps in mind the limitations of handheld devices and tries to alleviate the most problematic aspects of web display by handheld devices, while allowing you to produce pages primarily for desktop/laptop users. The following tips are gleaned from numerous online articles and tutorials, periodical articles, and from personal experience gained from participating in an HP grant to experiment with mobile computing on campus and in the classroom.

General Design Tips for the "Handheld Friendly" Site

Since handhelds differ from desktops by having 1) low bandwidth, 2) slower processors and less memory, 3) clunky input methods, and 4) a much smaller display, think "minimal links, minimal margins, minimal clicking". (Pollock, p. 10).

- Focus on content, not display. Browsing patterns of people using handhelds are quite different from those using full size devices. They are very task-oriented, and if your site is visually complex they might not be able to get a good “mental map” of your pages, taking longer to complete their tasks.
- Use a search feature at the top of your page, if you think the site is large enough. 80% of small screen devices users will begin their use of a web site with a search feature, if offered (Jones, p.6).

- No amount of programming will ever be able to fully compensate for a small screen (we are a long way away from folding screens) so the upper left-hand corner of your web page will always be your most valuable screen “real estate”. Keep fixed navigation or important information at the top of your pages. Consider the primary reasons students will be visiting your pages to help you decide what your "key" information is. Don't, for example, put an "About Us" or "Contact US" link at the top of your page because your users probably aren't coming to your site for those reasons. Also, the second most valuable real estate will be at the bottom of the page. Never put your most important links or information in the middle of your page as searching for it will require more scrolling.
- Use internal navigational features when appropriate to minimize vertical scrolling. In fact, you may want to design your site to be mostly links on your main page (a “site map“), allowing users to quickly navigate to their destination.
- Anything you can do to speed the loading of your pages, such as optimizing graphics, or getting rid of unnecessary html code (deprecated code) will make using your site easier (and cheaper for phone users!). Keep it under 100KB per page.
- Keep the width of your page under 172 pixels so users don't have to scroll horizontally. They really, really hate that.
- According to research, the optimal length of your web page should be between 273 and 300 words. (Albers, p.199).
- Use no more than 2 frames, if you really have must use frames. Users would rather have the entire screen to view the main text of the page, the only exception possibly being navigational links along the left side of the page.
- Avoid unnecessary graphics, and always ALWAYS use an "alt" tag for the graphics you do use, even if they are buttons. (Users may turn off graphics all together.) Optimize your graphics, both in pixel size and file size, for faster loading. Remember that “sending eight 1KB images takes longer than sending four 2KB images” (Pollock, p.3) so keep the number of images on your page small as well.
- Tables may be supported or not, so avoid them if you can. Never use tables for formatting.
- Avoid excessive use of italics: it looks awful on a small screen and on some devices it will be translated into bolded characters. Which brings us to fonts. Just don't. I'm not kidding. Don't use the font tag. The largest number of fonts supported by any device is three. Phones have only one.
- Use only heading tags for different font sizes.
- Do not use color for meaning (such as in charts and graphs). Black and white devices will display as grayscale so you also want to avoid tactics like using yellow text on a white background (not enough contrast when it's all turned to grayscale).
- Support for Javascript, CSS, XML, dynamic pages, animated gifs, shockwave or flash animations (although there are now flash players available for Pocket PC and Palm devices) multiple windows, Java applets, php, forms, and <mailto> tags is uneven and spotty. This is one area where things are changing rapidly, however, so you might want to experiment with these features

soon.

- Finally, make sure you plan for that “ideal solution” just around the corner by making your pages XML friendly as well: take a look at <http://www.wdvl.com/Authoring/Languages/XML/XHTML/> to see what you need to do so that the pages you’re coding today won’t have to be recoded later.

Check Your Work

For mobile phones, there are several good emulators on the web: for iMode devices try <http://pukupi.com/big/tools/mimic/> or use the Reqwireless WebViewer Demo at <http://reqwireless.com/demo-webviewer.html>. These two tools will give you a good idea of what phone users will be seeing.

For PDAs, it’s much more problematic. If you have access to several PDAs, both Palm and PocketPC, then you’re all set! I wish there were a web applet for Palm and PocketPC devices that’s comparable to the above phone applets, but there just isn’t...yet. Instead, one has to download and install SDKs (Software Development Kits) for these devices from the manufacturers, which will include software “emulators” that run on your desktop and will let you see what a web page would look like on the device. This is WAY too much trouble, and way too much software to install. Instead, try setting the screen resolution on your web authoring software (such as Dreamweaver or Frontpage) to 150 X 140. This is the available screen pixels for web browsing on an older Palm, which should be the smallest of the PDAs. If you want to design for newer PDAs try a resolution of 240 X 320. Or, you could try and set your browser to mimic a handheld screen (try 11 lines of 65 characters each).

Conclusion

Unless one is anxious to capture the handheld user market for economic reasons, one can create a perfectly useable, handheld-friendly web site with just a little forethought and attention to detail. The added advantage of building a handheld friendly page is that modifications that make a page more useable on a handheld will also make it more accessible according to the Section 508 Web Accessibility Standards.

References

- Albers, M., & Kim L. (2000). Implications of the wireless web for technical communicators: User web browsing characteristics using palm handhelds for information retrieval. *Proceedings of IEEE professional communication society international professional communication conference and Proceedings of the 18th annual ACM international conference on Computer documentation: technology & teamwork*. Retrieved June 1, 2004, from the ACM Digital Library.
- Jones M., Mardesen, G., Mohd-Nasir, N., Boone, K., & Buchanan, G. (1999). Improving Web interaction on small displays, *Proc. of the 8th Intl. WWW Conf.* Retrieved June 1, 2004, from the ACM Digital Library.
- Microsoft Corporation. (2003). Designing Web sites for Pocket Internet Explorer for SmartPhone. Retrieved June 1, 2004, from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/guide_sp/html/designing_web_sites_for_internet_explorer_for_pocket_pc_mgvc.asp

Pollock, H. (2004). The end-all guide to small-screen Web-dev. Retrieved June 1, 2004, from <http://hotwired.lycos.com/webmonkey/04/12/index4a.html?tw=design>

Trevor, J., Hilbert, D. M., Schilit, B. N., & Koh, T. K. (2001). On the move: From desktop to phonetop: A UI for Web interaction on very small devices. *Proceedings of the 14th annual ACM symposium on User interface software and technology*. Retrieved June 1, 2004, from the ACM Digital Library.

Biographical Sketch

Karen Tusack is an Instructional Technology Consultant with the Letters & Science Learning Support Services, University of Wisconsin-Madison, where she has helped instructional staff use technology inside and outside the classroom for 15 years. She has Masters degrees in both Slavic Languages and Library and Information Studies.

Address: 1220 Linden Dr. Rm. 279
UW-Madison
Madison, WI 53706
E-mail: karen@lss.wisc.edu
Phone: 608.262.4471