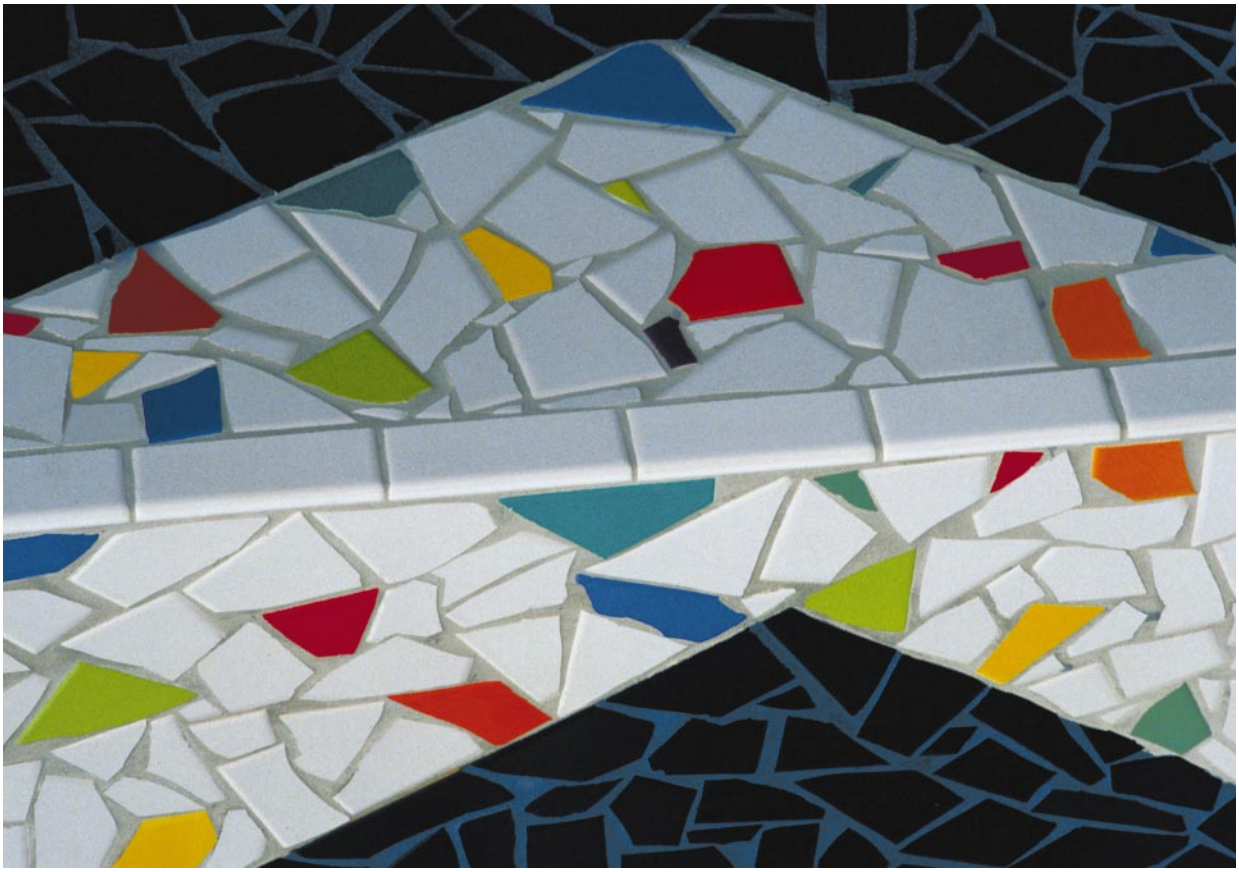


Designing for the Web: A Survey

A majority of the surveys on the Web, for example, the surveys conducted by Georgia Tech's Graphics, Visualization, & Usability (GVU) Center [2]; Nielsen Media Research [6]; and Nua [7], have focused on understanding the profile of the users of the Internet and the Web. As it is important to understand how users interact (i.e., read and navigate) with Web pages in order to design usable Web pages, it is also important to understand how Web authors (or designers) design Web pages in order to design usable Web page development tools.

The May/June Design and Methods & Tools columns have been combined into one as a result of a collaboration between the editors for each column.

Although the 8th GVU survey did have a section for the Web authors, the emphasis was on discovering respondents' backgrounds, development experience, use of Java™, and other information related to Web development. The focus of this survey, on the other hand, is on understanding the process of designing and developing Web pages and to understand better how Web developers design Web pages. Specifically, the survey was designed to achieve the following objectives:

- Construct Web designer profile. Who are typical Web designers? What kind of professional training do they have? How much experience do they have designing Web pages? What types of pages do they design?
- Identify the key components of the Web designer's "toolkit." What kinds of Web development tools do they use? What do they like or dislike about them and why?
- Understand the Web development environment. How much time does it take to develop Web pages? How many pages are developed in that time frame? Do Web designers work in teams? How many people participate in these teams? Do the time available and number of people in the team affect their selection of tools?
- Outline the design process. Do Web designers use style guides/design guidelines when developing Web pages? Do they consider bandwidth and accessibility issues when designing Web pages? How do they evaluate the Web page designs for usability? How do they publish Web pages to the Web server?

Methodology

Survey participants were solicited as follows:

- Announcements on newsgroups: comp.human-factors and alt.hypertext.
- Announcements made to the mailing lists: UTEST, World Wide Web Artists Consortium, WebHCI, Human Factors/Web conference participants, Webgrrls (Chicago Chapter), WebWomen, and Web Design.

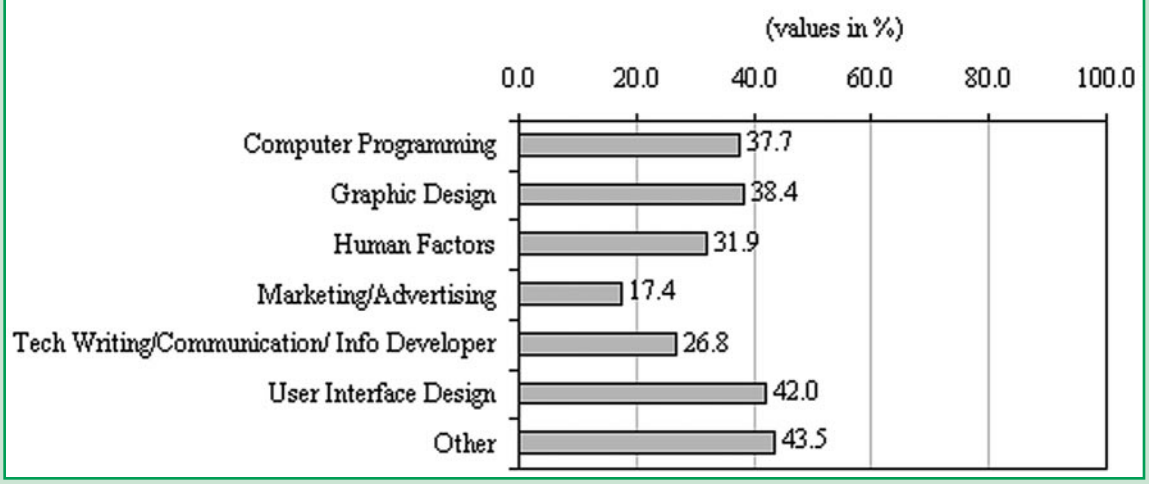
One hundred and thirty-eight people responded to the survey over a 4-week period from December 17, 1997, to January 15, 1998. Like other surveys on the Internet, these respondents were self-selected, and they responded to the survey via

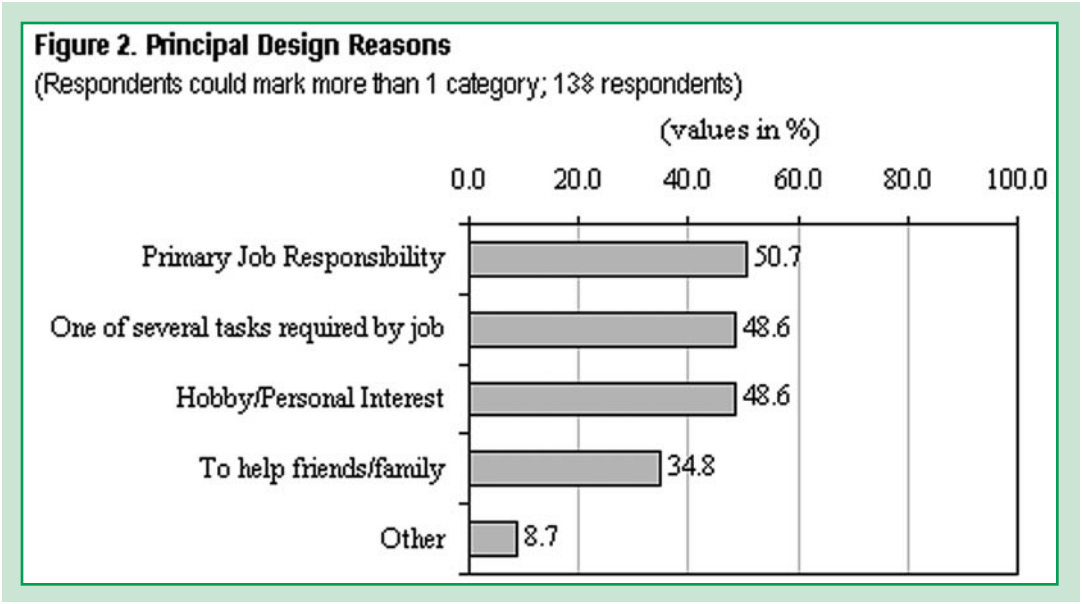
- E-mail because the survey was included in the solicitation message.
- The Web (the survey was made available at the ACM SIGCHI's Web site; URL: http://www.acm.org/sigchi/web/survey/index.html).
- Postal mail.

Survey Results

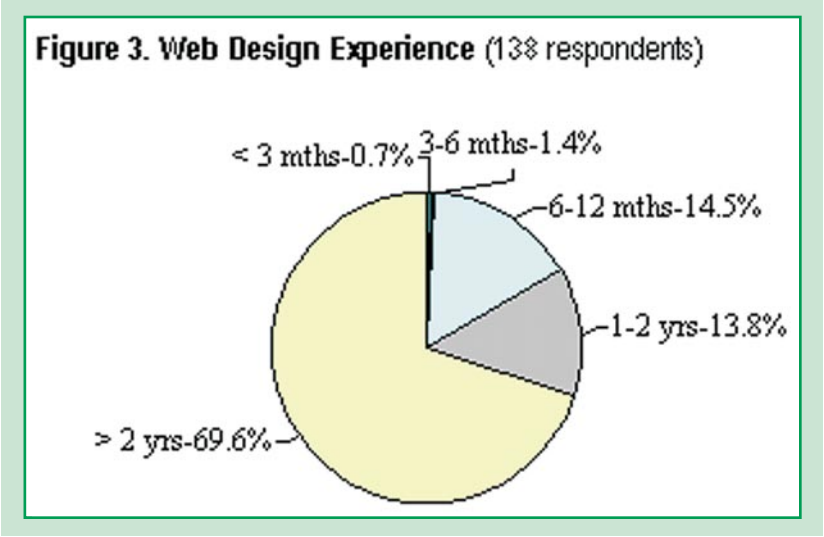
The survey results are organized in the fol-

Figure 1. Professional Background
(Respondents could mark more than 1 category; 138 respondents)





lowing sections: Web Designer Profile, Authoring Tools, Web Development Environment, Web Page Design & Evaluation, Publishing Web Pages, and Overall Comments.



Web Designer Profile

Professional background

The responses indicated that there is no prototypical professional background for a Web designer (see Figure 1). They come from a variety of professional backgrounds, such as computer programming, graphic design, human factors, marketing/advertising, user interface design, and technical writing, and often have additional training in cognitive psychology, multimedia, and education. A relatively high number of responses from people with professional backgrounds in user interface design, graphic design, and human factors, however, were expected because these professions are well represented in the newsgroups and mailing lists from which the responses were solicited.

Reasons for designing Web pages

There was no single, dominant reason for these individuals to be authoring Web pages (see Figure 2). "Primary job responsibility" was cited as often as "one of the several tasks required by job," "hobby/personal interest," and "to help friends, family, and non-profit organizations." Other reasons for designing Web pages included education, communication, community building, artistic endeavors, and to make money (or to engage in a profitable activity).

Web page design experience

Interestingly, a majority of the survey respondents (96 of 138) had more than 2 years of experience (see Figure 3), suggesting a bias toward the "expert" category of Web developers. However, because the survey didn't ask the respondents to give an estimate of the number of Web pages they have designed or the number of Web pages they

design in a month, it is difficult to discern their "expertise" in the area of Web page design truly.

Principal audience

When asked to identify principal audiences for Web pages they designed, authors' responses were overwhelmingly focused on corporate sites for customers and vendors or for personal use by the author (see Figure 4).

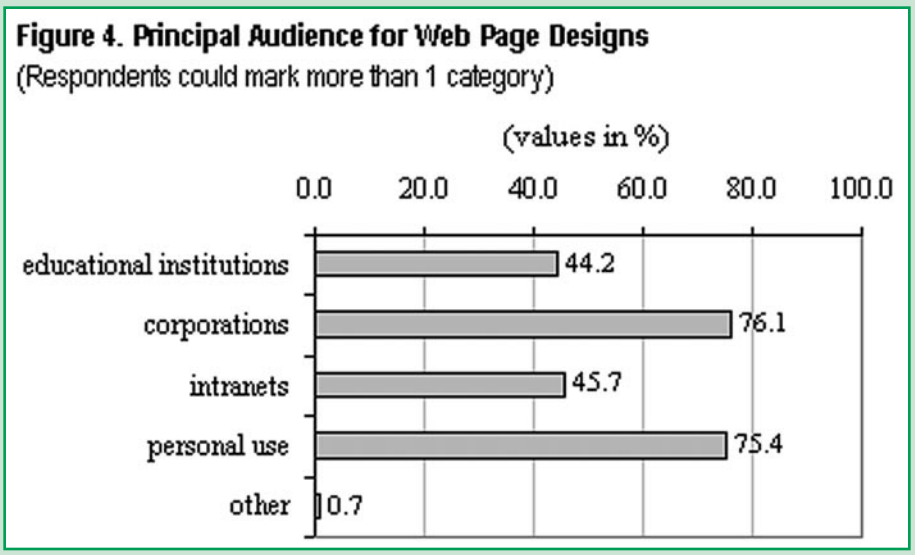


Figure 4. Principal Audience for Web Page Designs
(Respondents could mark more than 1 category)

Types of Web pages designed

Finally, the types of Web pages designed by author respondents are shown in Figure 5. Given that a majority of Web pages were designed for corporations and personal use (Figure 4), the most common Web page types were work/organization information, product/service information, and personal home pages. A small percentage of pages designed was research papers/reports (6.5%), which are more likely to be for educational institutions.

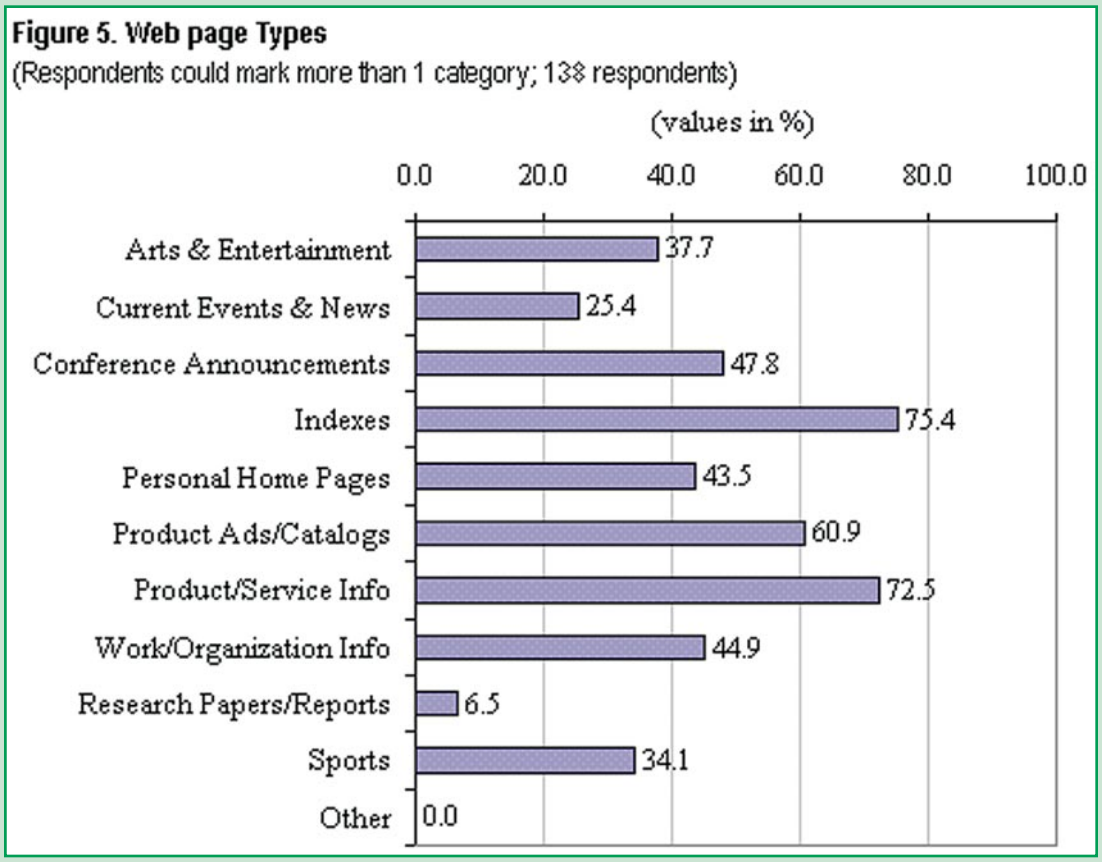


Figure 5. Web page Types
(Respondents could mark more than 1 category; 138 respondents)

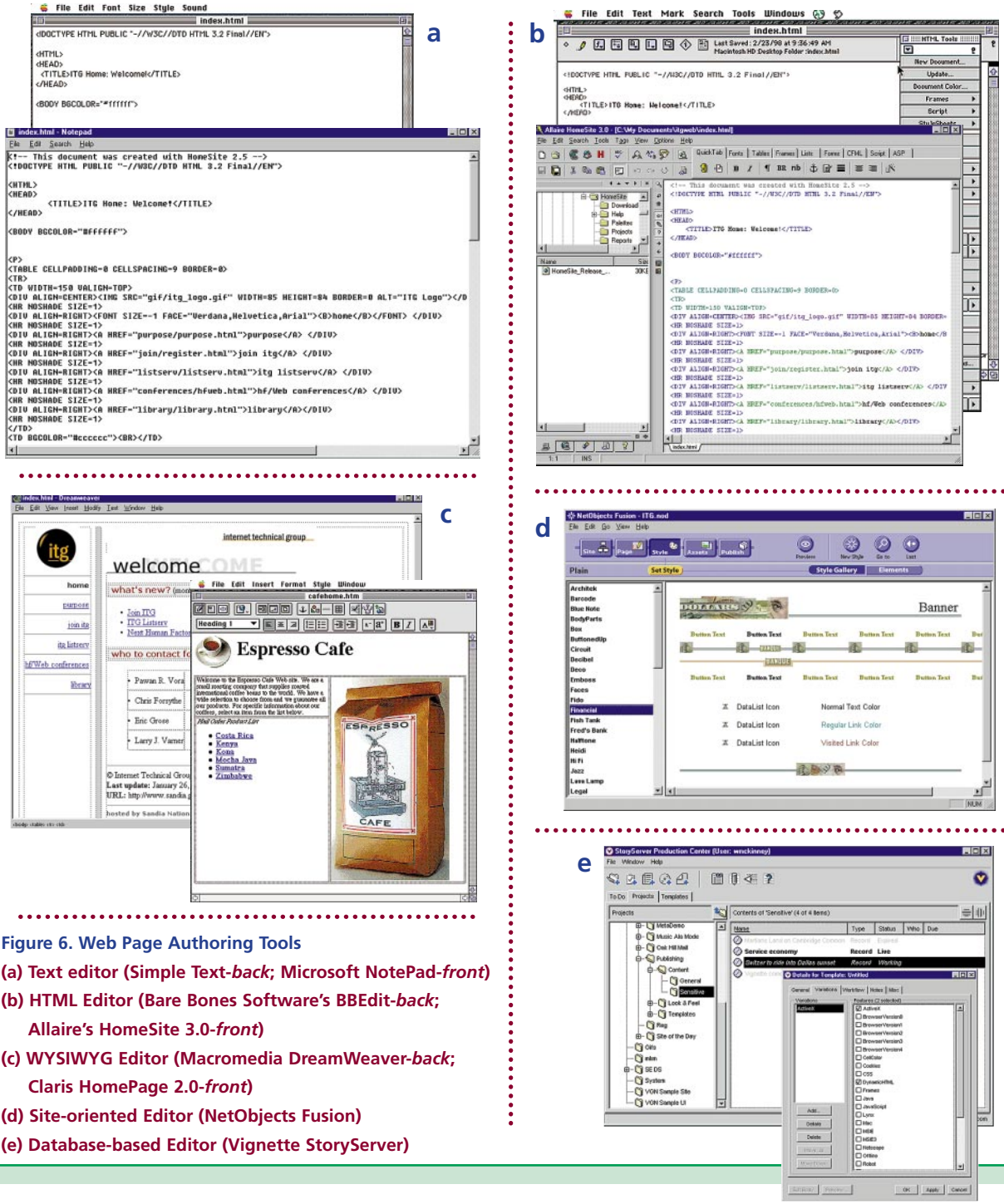


Figure 6. Web Page Authoring Tools

(a) Text editor (Simple Text-*back*; Microsoft NotePad-*front*)

(b) HTML Editor (Bare Bones Software's BBEdit-*back*; Allaire's HomeSite 3.0-*front*)

(c) WYSIWYG Editor (Macromedia DreamWeaver-*back*; Claris HomePage 2.0-*front*)

(d) Site-oriented Editor (NetObjects Fusion)

(e) Database-based Editor (Vignette StoryServer)

Authoring Tools

This section addresses the types of authoring tools used. It is divided into four subsections: Web page editors/generators, HTML (HyperText Markup Language) conversion tools, PDF (Portable Document Format) conversion tools, and graphic design tools.

Web page editors/generators

There are several ways Web designers create and generate Web pages. They often use one or more of the following (see Figure 6):

- Text editors. Because HTML files are ASCII text files, text editors offer the most basic way to design the Web pages.

Pawan R. Vora
U S WEST
Communications
931-14th Street,
Suite 710
Denver, CO 80241, USA
1+303-685-2023
pvora@uswest.com

They do not, however, offer any assistance to Web designers, and it is the responsibility of the Web designers to remember, write, and debug the HTML code they develop. Examples of text editors are NotePad, WordPad, SimpleText, and vi.

- HTML editors. HTML editors support web designers in writing HTML tags using menus, toolbars, and keyboard shortcuts. They do not, however, write or hide the HTML tags from Web designers; Web designers work directly with raw HTML tags. Many HTML editors also offer additional functionality such as HTML validation, link verification, spellchecker, and customized menus. Examples of HTML editors are Allaire's HomeSite, Bare Bones Software's BBEdit, and Sausage Software's HotDog Professional.
- WYSIWYG ("What You See Is What You Get") editors. WYSIWYG editors work like word-processing or desktop publishing programs, allowing authors to lay out pages as they want, and the editors write the necessary HTML code in the background. The Web designers do not really need to know HTML. Examples of WYSIWYG editors are Adobe PageMill, Claris HomePage, Symantec's Visual Page, Microsoft FrontPage, and Macromedia Dreamweaver.
- Site-based editors. Unlike the Web page development tools discussed so far, which are page-oriented, site-based editors such as NetObjects Fusion and GoLive's CyberStudio allow the Web designers to add and rearrange pages in a tree-structured site diagram. The navigation bars and links for the Web site are automatically generated for them. Site-based editors also allow rapid changing of the overall look of a site.
- Database-generated Web sites. Many large Web sites do not build each page manually. They hold the web page elements (text, graphics, headers, footers, etc.) into a database and automatically generate pages by retrieving and combining the elements as the Web browsers request them. Examples of database-oriented Web page–generation tools are Vignette Corporation's Vignette and Lotus Notes (with Domino Web server).

The survey asked respondents to identify the type(s) of Web page editors/generators they had used, to identify specific products used, and to rate, on a scale from 1 to 7 (1 = low and 7 = high), those products in terms of satisfaction, learning, ease of use, functionality, and extensibility. A summary of their responses is shown in Table 1.

The majority of respondents (104) used basic text editors to design Web pages. The most frequently reported text editors were Notepad (42), SimpleText (17), WordPad (14), vi (11), and BBEdit Lite (8). HTML editors ranked second as the authoring tool of choice; 89 respondents reported using them. The most common HTML editors were Bare

Table 1. Evaluation of Web Page Development Tools (Response Scale: 1 = Low; 7 = High)

	Text	HTML	WYSIWYG	Site	Database
Number of Responses	104	89	74	29	19
Satisfaction	4.76 (1.70)	5.93 (1.06)	4.70 (1.59)	4.79 (1.45)	4.83 (1.38)
Ease of Learning	5.41 (1.91)	5.48 (1.21)	5.19 (1.45)	4.38 (1.47)	3.78 (1.80)
Ease of Use	5.40 (1.68)	5.75 (1.13)	5.20 (1.37)	4.79 (1.35)	4.06 (1.73)
Functionality	4.21 (1.82)	5.67 (1.35)	4.49 (1.57)	4.72 (1.62)	5.44 (1.72)
Extensibility	3.60 (1.95)	5.57 (1.52)	4.07 (1.66)	4.24 (1.88)	5.53 (1.23)

Bones Software's BBEdit (38) and Allaire's HomeSite (22). Of the 74 respondents who reported using WYSIWYG editors, the most commonly named were Microsoft's FrontPage (27), Netscape Gold/Composer (15), Adobe PageMill (10), and Macromedia Dreamweaver (10). For site editors, Microsoft's FrontPage (11) was the leader over NetObjects Fusion (8). Finally, for database-based Web page–generation tool, the frequently used were Lotus Notes/Domino (6) and Cold Fusion (4).

By examining Table 1, it is obvious that respondents were quite satisfied with HTML editors, ranking them higher with respect to functionality, ease of learning, and ease of use than the other methods of creating Web pages. WYSIWYG editors probably generated the majority of the "negative" comments where their usability and effectiveness were concerned. Some of the comments were:

Most "WYSIWYG" editors insert too much extraneous code for my taste.

Most tools don't support all of the current HTML tags & keywords—most have property boxes or dialogs, but you have to use the HTML view to edit the code directly to get anything non-trivial done.

Prefer writing code directly rather than using "WYSIWYG" tools because these allow more control of the HTML code and do not add "redundant" code.

I don't use it [FrontPage] anymore, because it is too inflexible/automatic for my needs; e.g. it puts in tags that I don't want, or takes out ones I need, requiring lots of post-editing.

The WYSIWYG HTML editors that I have used are useless, and don't do all the things that HTML can do.

From respondents' comments, it was clear that their annoyances stem from the one or more of the following characteristics of WYSIWYG authoring tools:

- Generation of excessively redundant code.
- Overriding (not respecting) author's HTML code, requiring author to use another editor to finish the work.
- Not allowing authors to see and/or edit the HTML code.
- Not supporting all HTML tags.

The basic premise of WYSIWYG authoring tools is that the Web designer need not know any HTML. They often do not, however, support all the HTML tags and/or do not allow authors to write their own HTML code. When they do allow authors to edit the raw HTML code, the authoring tools often override the HTML code written by the Web designers. Web designers, however, need better control over their code. This is how one respondent commented:

Web Authoring and Web Designing tools should allow maximum freedom to the designer. I think this is the reason why Homesite and BBEdit were successful because they allow maximum flexibility to the user. I think this is also the reason why NetObjects Fusion is doomed because the designer is trapped to the confines of the program's rigid structure.

This doesn't mean, however, that all Web authors want to write HTML. It's just that today's Web authoring tools do not do a good job in delivering what an author needs. This is evidenced in the following comments:

As a designer, I have little interest or talent in writing code, so I depend on a good WYSYWIG editor like NetObjects or FrontPage to help with that aspect. Unfortunately, to date there has been no editor that is as clean, efficient, or as versatile, as hand-coded HTML.

I don't mind doing it, but I think most designers shouldn't have to learn coding to do cool things on the Web.

Interestingly, WYSIWYG tools are used by some authors to design simple pages instead of their proposed use for designing complex pages. As one respondent commented, "Web tools OK for simple pages, often easier, but prefer text editor for anything more complex."

The exception was Macromedia Dreamweaver, a recent entrant into the market of WYSIWYG authoring tools. It received several praises. Obviously, the main source of praise was that Dreamweaver does not override the author's HTML code and lets the Web author use an HTML editor of choice. As one respondent indicated, "This tool gives the user the ease of a WYSIWYG, but the

Design Column Editor
Kate Ehrlich
Senior Research Manager
Lotus Development Corporation
55 Cambridge Parkway
Cambridge, MA 02142
+1-617-693-1899
fax: +1-617-693-8383
Kate_Ehrlich@crd.lotus.com

Methods & Tools Column Editors
Michael Muller
mullerm@acm.org

Finn Kensing
Computer Science
Roskilde University
P.O. Box 260
DK-4000 Roskilde
Denmark
+45-4675-7781-2548
fax: +45-4675-4201
kensing@dat.ruc.dk

control of BBEdit."

HTML conversion tools

Fifty-four respondents reported using HTML conversion tools; their ratings are summarized in Figure 7. The most commonly used conversion tools were Microsoft's Internet Assistants (for Word, Excel, and PowerPoint) and Office '97.

Overall satisfaction with HTML conversion tools was fairly low (mean, 4.17; standard deviation, 1.75), despite their being considered fairly easy to learn (mean, 5.19; standard deviation, 1.51). The main reason cited for low satisfaction was that conversion tools, such as WYSIWYG editors, generate considerable redundant HTML code during the conversion. This makes the file sizes larger and the HTML code difficult to read. In addition, it is cumbersome to change the "converted" HTML documents manually and "risky" to try and make global changes across several documents at the same time.

The reason for the low value for "extensibility" for HTML conversion tools (Figure 7) is obvious when you consider that a majority of respondents used Internet assistants for Microsoft Word, Excel, and PowerPoint for converting the documents to HTML. These tools convert only the documents created using Microsoft Word, Excel, and PowerPoint, respectively. Of course, one can always import the documents created by other business productivity suites in one of the Microsoft Office products and then convert them to HTML.

PDF tools

Approximately 25% of the respondents (34) reported creating PDF documents on the Web. A majority of users reported using Acrobat Exchange (14) and Adobe Distiller (8). Several people also reported using the functionality available in PageMaker and FrameMaker to save files directly as PDF documents. Overall, most users seemed satisfied with the products they were using to convert documents to PDF (see Figure 8).

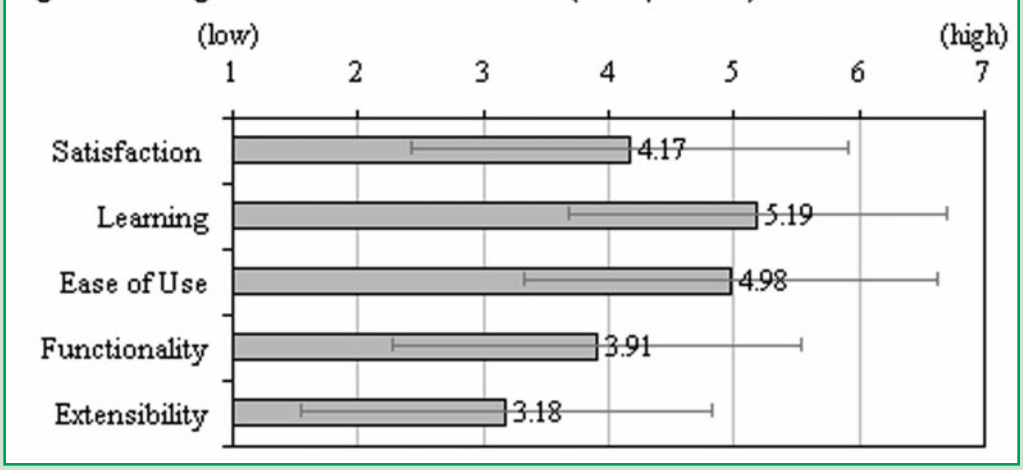
Using a combination of tools to develop Web pages

A majority of users reported that they "always" (51.4%) or "sometimes" (31.9%) used a combination of tools (see Figure 9).

The combination of tools used invariably consisted of one or more of the HTML editors and graphics tools. The most common was a WYSIWYG editor along with a text editor or an HTML editor. A text editor or a text-based HTML editor was seen as necessary in order to "clean up" the code created by WYSIWYG editors. Specific reasons for using a combination of tools are as follows:

- To clean up the code generated by WYSIWYG editors or HTML conversion tools. This comment probably summarizes the most circuitous route: "For text oriented sites, I usually use Internet Assistant for Word to convert text to HTML, then I use BBEdit to strip out all the junky formatting that Word causes. Then I use DreamWeaver to format with Style Sheets, and fine-tune the design."

Figure 7. Ratings to HTML Conversion Tools (54 respondents)



Another comment also provides the necessary evidence: "Pagemill is great for quick pages . . . but it writes sloppy code and unless you use SimpleText or BBEdit are you able to really program or clean code up."

- To support differences in the development platform and production platform. As one respondent described, "I start in VisualPage on the Mac because it's easiest, then tweak it in a text editor and bring it over to my UNIX box. If I then need to update it frequently, like the Weekly Seminar schedule, I update it in Composer, because it runs on UNIX and I don't have to move the files around."
- To take advantage of special features offered by some tools. Several Web design tools offer functionality beyond creation of Web pages, for example, spellchecker, link verifier, HTML validator, and site manager. Web page designers often use additional tools just to take advantage of such specialized functionality, especially when they are not integrated or poorly implemented in their otherwise preferred development tool of choice. One reason for doing so is Web site management: "Often the client wants to manage/add to the site themselves, so things are often designed in BBEdit, and then imported into FrontPage for the 'link management'."
- To create graphics. Although WYSIWYG Web page design tools support conversion of graphics to a Web-ready format, they generally do not provide sophisticated tools for creation of graphics. Most designers use dedicated graphics software to create the graphics and point to them when designing the Web pages.
- To handle multimedia components in a document. Because most Web page development tools do not support creation of multimedia components (audio, video, animation, etc.), separate development tools are required for their creation.

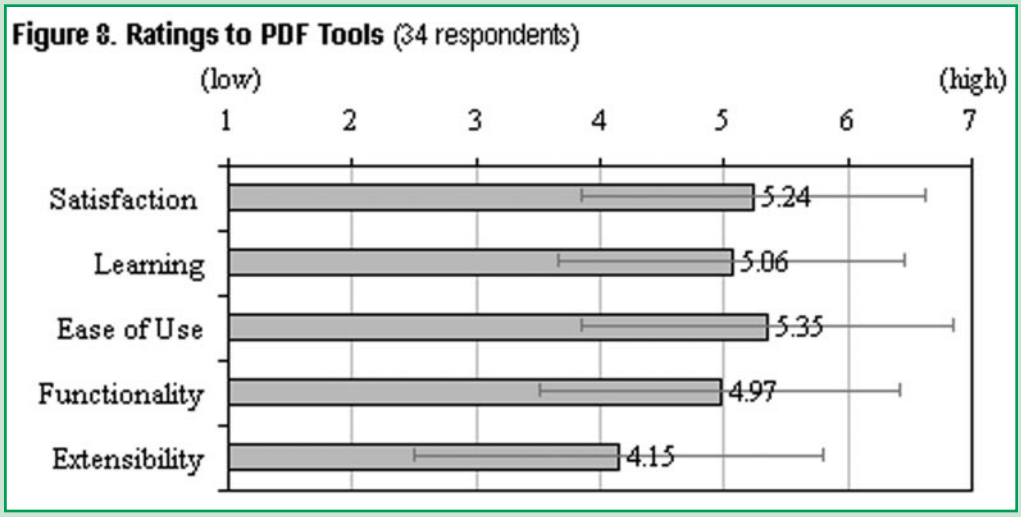


Figure 8. Ratings to PDF Tools (34 respondents)

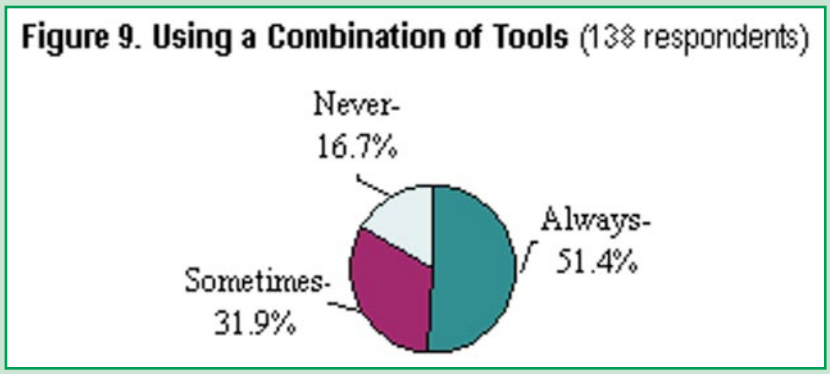


Figure 9. Using a Combination of Tools (138 respondents)

The question, of course, is this: Should there be a tool that allows people to do all they need to do to develop Web pages? Maybe not! As a few respondents pointed out the relevant concerns:

> In this era, serious work in almost any software endeavor requires appropriate tools. Relying on one tool to do everything is now the mark of (a) a trivial task or (b) an amateur.
>
> [B]ecause there is no one tool that does it all, and if there was [I] probably wouldn't use it because [I] am generally not impressed with software that has a bazillion functions.

Designing graphics

A majority of respondents "always" (44.2%) or

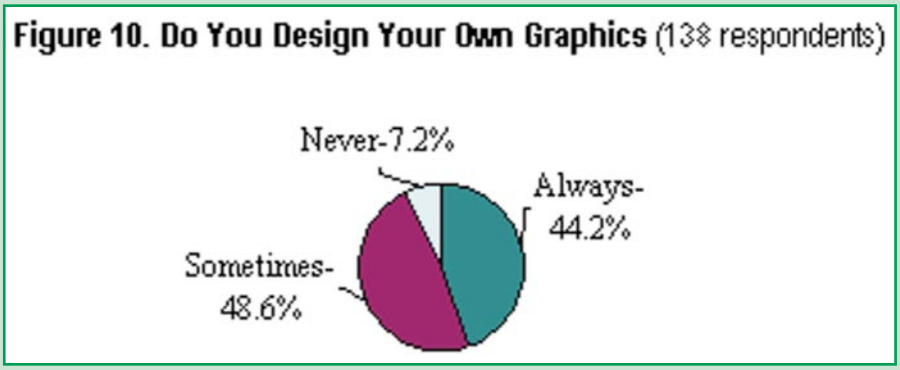


Figure 10. Do You Design Your Own Graphics (138 respondents)

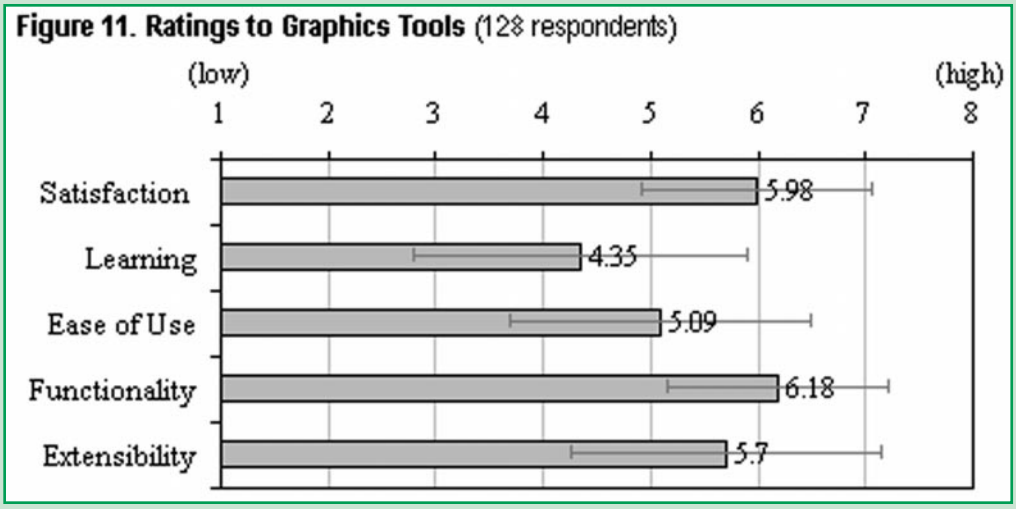


Figure 11. Ratings to Graphics Tools (128 respondents)

"sometimes" (48.6%) designed graphics when they design Web pages (see Figures 10 and 11). The most commonly reported product for developing graphics was Adobe PhotoShop (90 respondents reported to using it.) Interestingly, despite it being noted as relatively difficult to learn (mean, 4.35; standard deviation, 1.55), the satisfaction with Adobe PhotoShop was fairly high (mean, 6.34; standard deviation, 0.73). Other commonly used graphic design products were Adobe Illustrator (21), PaintShop Pro (7), and Debabelizer (6).

Web Development Environment

There were three questions related to the Web development environment:

1. What is the average time to develop a set of Web pages?
2. How many Web pages are designed in this duration?
3. On average, how many people participate in Web development (including yourself)?

The purpose of these questions was to determine if responses to them have any relation to the Web development tools used, use of an in-house Web design style guide, use of and the type of usability evaluation methods used, and other Web design and development variables.

As pointed out by several respondents, however, these questions were ambiguous and ineffective in capturing useful information. Here are some of their reactions:

> *[D]umb question, it depends on if I have to do content, and how many graphics they want. Also what is a set? It depends on if there are a lot [sic] of links.*
>
> *How long is a piece of string?! It depends on the project, who's doing the work, [and] what they want. . . . We recently redesigned our 11 top welcome and contents pages. This project took over a year from design to implementation. I did an entire web site of 11 pages for a health centre in 3 days.*
>
> *[D]on't think it's possible to answer this. It's very dependent on what I'm developing, number and type of graphics, whether I'm creating content or using stuff that's already been written, etc. . . . I'm not sure what you mean by participate. . . . I tend to do most of the coding, but get other people to help with design, testing, content-generation, etc.*

The ambiguity in our questions is obvious from the previous comments. Despite the ambiguities, most people did respond to the questions, and their responses are summarized in Figures 12, 13, and 14. The responses can be summarized as follows:

- A majority of Web development projects

are less than 3 months with many of them less than a month.

- About 50 or fewer pages are developed in this time period.
- Most Web development teams have fewer than five people on their team.

Because of the potential misunderstanding and corresponding low reliability of the responses, however, I have refrained from correlating these with the Web development tools used, usability testing performed, and other Web design variables.

Figure 12. Avg. Time to Develop a Set of Web Pages (130 respondents)

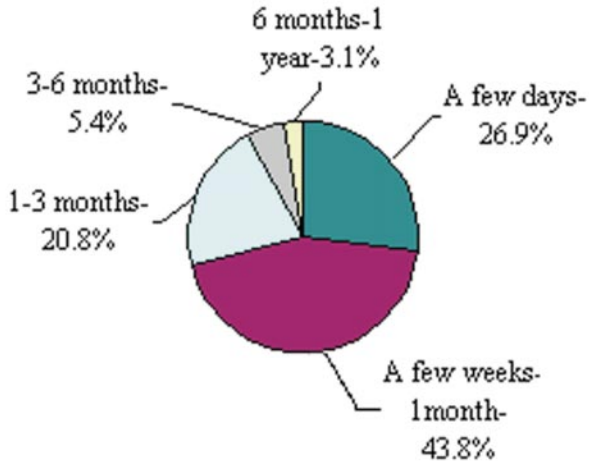


Figure 13. Number of Web Pages Designed (128 respondents)

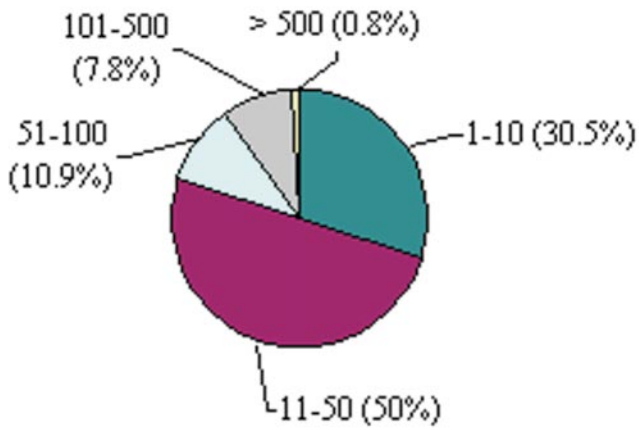
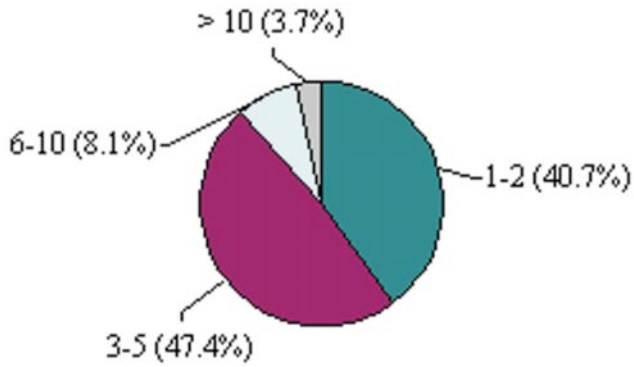


Figure 14. Number of People in the Development Team (128 respondents)

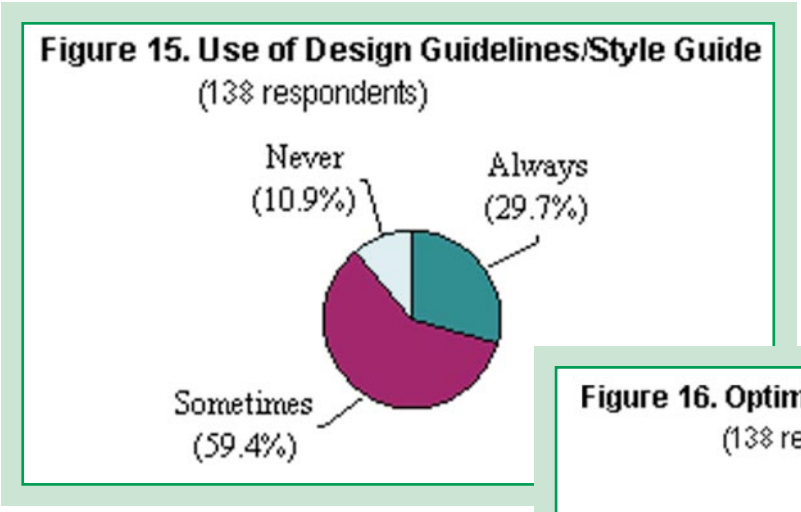


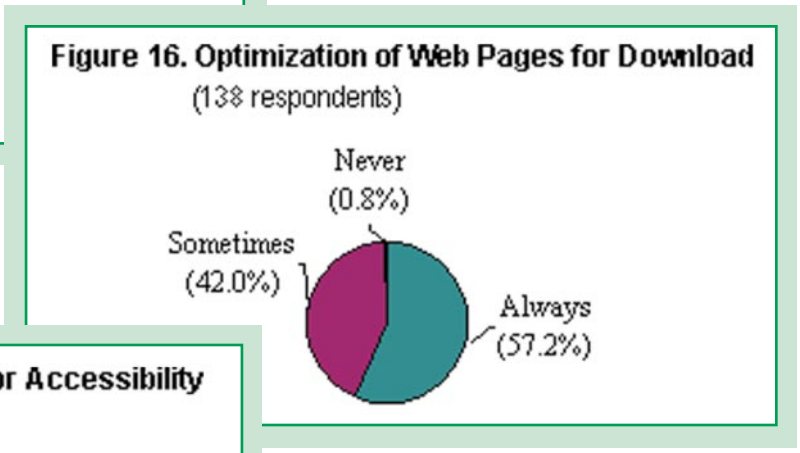
Design and Evaluation

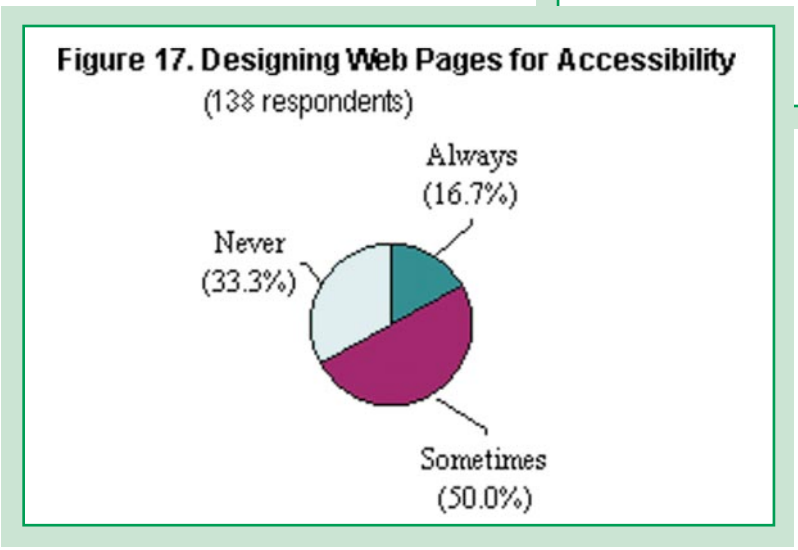
Use of Web design guidelines/style guides

Almost 90% of the respondents reported using Web design guidelines and/or style guides when designing Web pages (see Figure 15). Of these, about 38% used design style guides developed in-house, and about 50% of them used design guidelines available on the Web, as well as those developed in-house.

Although this is encouraging, one can only surmise the extent to which the design guidelines are used in designing Web pages. A similar survey conducted by Mosier and Smith [5] for designing software applications showed that although the user interface design guidelines are considered generally useful, there are significant problems in the practical application of guidelines. The most frequent problem for difficulty in applying guidelines was that the guidelines were too general and







were often not translated into specific design rules that developers can use.

Web design guidelines tend to be more specific as they place emphasis on common look and feel of the Web pages [3] and, therefore, easily implemented. For example,

- ✘ Pages should be time stamped and dated.
- ✘ Every page should have a title.
- ✘ Provide WIDTH and HEIGHT information for images.
- ✘ Do not use "click here" or similar words to designate a link.

Designing for faster download

Most Web designers seemed aware of bandwidth bottlenecks on the Web and the users' dislike about often long waits to download Web pages; GVU's 8th WWW User Survey reported that 63% of the respondents were unhappy with the speed at which Web pages download [2].

Ninety-nine percent of the respondents in this survey indicated that they always or sometimes design pages for faster download (see Figure 16). But, as one respondent pointed out, often the clients drive the decision: "[S]ome clients claim all their visitors will be on T1's, and thus they want mondo [sic] graphics and could care less about folks on modems." Though designing bandwidth-heavy sites may seem justified in such situations (for example, when designing intranets), I have noticed that many clients overlook the needs of telecommuters, who may be using slower modems to access the Web pages.

Designing for accessibility

The responses to this question—Are designs optimized to improve access to people with disabilities?—were not very encouraging. One third of respondents has never considered accessibility when designing Web pages. Only 16.7% of respondents have always designed Web pages to improve access to people with disabilities, and the remaining 50% of the respondents have "sometimes" optimized their designs to improve accessibility (see Figure 17). This is similar to what Daryle Gardner-Bonneau found when she conducted an informal survey on UTEST (a listserver for

usability professionals) in late 1997 on Web site accessibility [1]. She concluded that

- Despite the simplicity of making Web sites largely accessible to people with disabilities, very few were actually doing it.
- People really didn't know much about this problem or the available solutions for it.

Very few respondents answered Daryle's questions about whether and/or how they ensure that their web sites are accessible. She commented:

> *There's a certain hypocrisy (or irony) in the fact that the professionals on this list—which is devoted to usability—are largely ignoring a sizable body of potential clients when developing Web sites (at least commercial sites), especially when the solutions to accessibility problems are so simple, in most cases.*

Of course, Web designers may not always be at fault. As one respondent pointed out,

> *We always use alt tags, all the easy stuff, but few clients are willing to pay for anything that's extra work. . . . I think your survey is ignoring the fact that most of us Web designers are customer driven—i.e. I might want to do something a certain way (like make pages accessible to folks with disabilities) but the client doesn't want to pay for that "extra" so it doesn't happen. Also, even when we want to do extensive usability testing, clients always want the Web site "yesterday" and don't want to wait.*

Though the response rate appears to be fairly good in the current survey about the awareness of accessibility issues, it is not obvious which design strategies Web page authors adopt to design for accessibility. Do they design a separate text-only site? Do they provide "ALT" text description for the image or do they "D" tag the images to link to a page that contains textual description of the images? Do they avoid use of frames, or are they referring to some other design strategies? (For more details on Web accessibility, see Trace Research & Development Center Web site [8] and World Wide Web Consortium's Web Accessibility Initiative (WAI) [9].) To understand the accessibility issues in more detail, especially to understand how people accommodate special populations in their Web page designs, a separate survey looking into the accessibility issues is warranted.)

Usability evaluation of Web pages

When asked "Do you evaluate the Web pages you design for usability?" 70% of the respondents evaluated their Web pages for usability "all the time" or "almost all the time" (see Figure 18). A majority of these usability evalu-

Figure 18. Usability Evaluation of Web Pages (138 respondents)

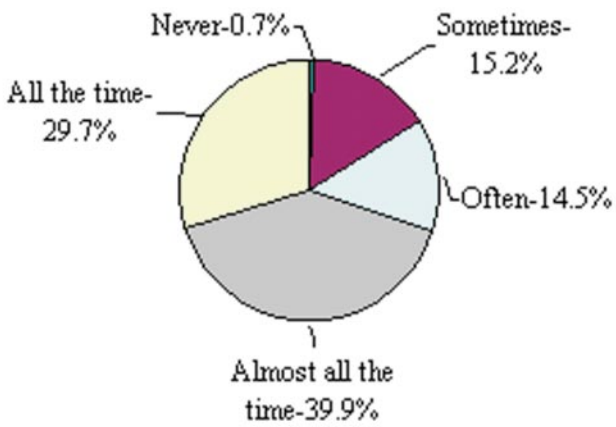
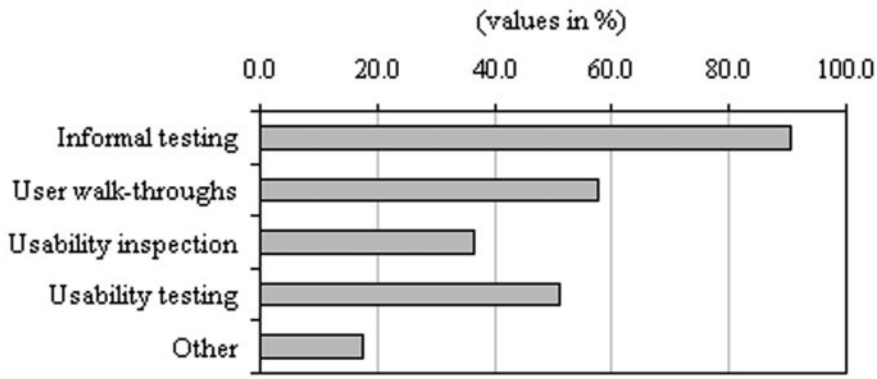


Figure 19. Type of Usability Evaluation Methods Used
(Respondents could mark more than 1 category; 137 respondents)



ations, however, are informal (conducted with colleagues and people in the development team) or user walk-throughs (see Figure 19). Other evaluation methods used by the respondents were user questionnaires and surveys, user feedback forms, remote usability testing, focus groups, and off-line testing (e.g., testing access speeds at 28.8 K modem, testing with a variety of browsers, and analysis of server logs).

What was encouraging was that almost half of the respondents conduct usability testing in a controlled laboratory setup. Although some testing is better than no testing, informal testing does have limitation in that the user's viewpoint is overlooked. As one respondent said, "Many people I work with evaluate the web based on how 'they' surf; disregarding how our typical viewer surfs." And, most people in the Web development teams are less likely to be typical users.

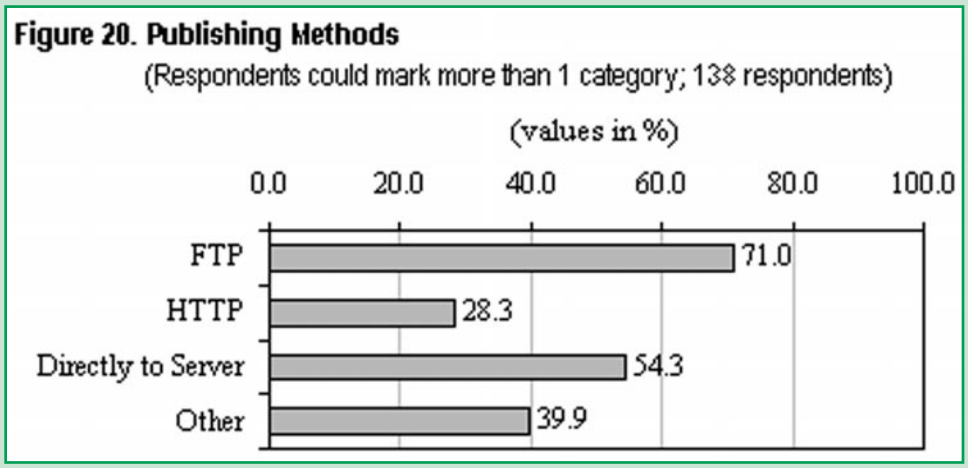


Figure 20. Publishing Methods
(Respondents could mark more than 1 category; 138 respondents)

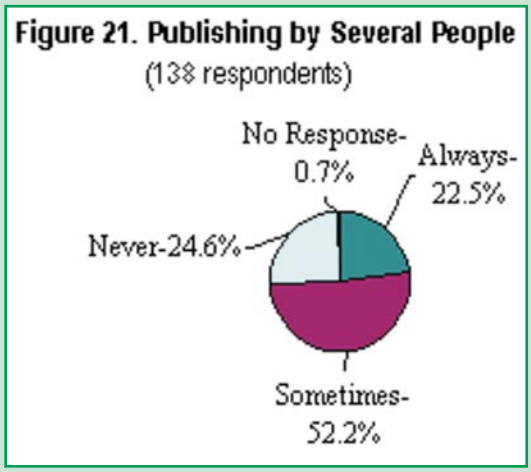


Figure 21. Publishing by Several People
(138 respondents)

Such a high number of respondents conducting some form of usability tests are probably expected as a significant number of the respondents had professional backgrounds in human factors, user interface design, technical writing, and marketing. Consequently, it is possible that usability professionals, who are more likely to be user-oriented, are overrepresented in this survey.

Publishing Web Pages

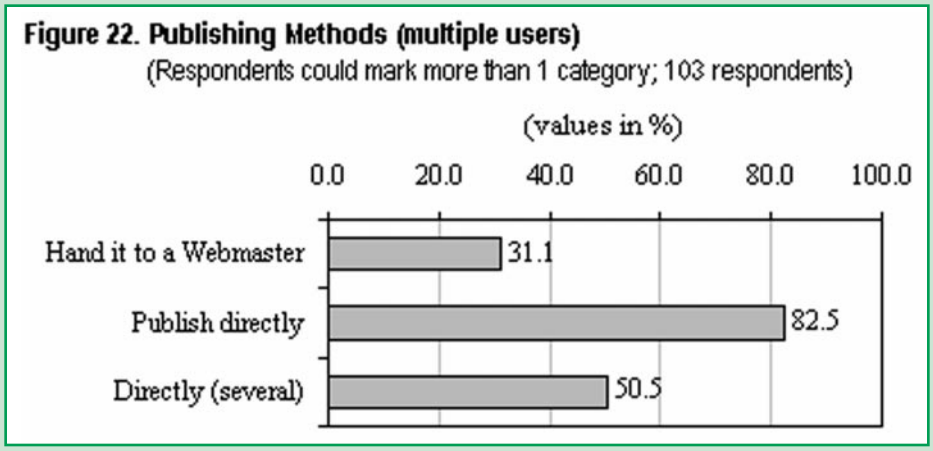
The final step in Web page development is transferring the Web pages to the Web server. There are three main ways to do this:

1. FTP (File Transfer Protocol). By logging into the FTP site of the Internet service provider (whether internal or external) and transferring the files.
2. HTTP PUT. This method also requires log-in, but the file transfer takes place by using the Web's Hypertext Transport Protocol (HTTP).
3. Directly copying the Web pages to the Web server.

As shown in Figure 20, the two most common publishing methods were FTP and copying the files directly to the server. It is not surprising that publishing via HTTP was not that common because few authoring tools support it; one reason is that currently it is not as secure as FTP, unless SSL (Secure Sockets Layer) is used along with HTTP.

The need for several people in the Web development team to publish the files to the server appears to be quite common; about 75% of the respondents have encountered that situation either "always" or "sometimes" (see Figure 21). When several people need to publish on the Web, appropriate measures must be taken to ensure that the authors do not overwrite each other's work or delete the files by mistake. Of course, with the support of check-in/check-out of documents in publishing tools, costly errors could be avoided. Very few Web development tools today support such functionality. To avoid such errors, team members could either hand the work to a Webmaster or a coordinator,

who can then publish the Web pages directly to the Web server. In almost 50% of the cases, however, directly copying to the Web server is the method of choice (see Figure 22).



Overall Comments Offered by the Respondents

Respondents' comments made clear some of their dislikes and annoyances when designing Web pages. Their comments are summarized below.

Annoyances with WYSIWYG authoring tools

As mentioned before, most people were disappointed and/or frustrated with WYSIWYG authoring tools. They tended to follow up their use of WYSIWYG tools with text editors or text-based HTML editors to "clean up" the HTML code and/or to fine-tune the final look and feel of the Web pages.

Inadequate attention paid to the audience's needs

Several respondents felt that adequate attention was not paid to the needs of the actual users.

Far too little emphasis is placed on the design of information for the Web as it is perceived by the target audience. . . . When too much emphasis is placed on the technology, the site becomes a gadget trap for the user; similarly, when too much emphasis is placed on the artistic component, the site becomes something lovely to look at but not a useful tool. A well-researched navigational architecture brings together the technical wizardry and excellent, effective design to create a Web site that is intelligible to the user, easy to use, and as a result accomplishes its mission. This is an essential first step in the development of expansive and complex Web sites. (Note: the original comment was in ALL CAPS.)

Too many web designers fail to consider their potential users when designing their web pages. A good example of this designing web pages that takes remote access users (users dialing in through a modem) several minutes each page to download. This is as much of a usability problem, in my opinion, as poor site mapping or outdated links.

Browser incompatibilities

"Browser uniformity, browser uniformity, browser uniformity," was a comment made by one respondent.

This was probably the most frequent complaint among Web designers. Designing Web pages that not just look good but work for all users has become extremely difficult, if not impossible. This is obvious when you consider the use of proprietary technologies and incompatible support for HTML standards among different browsers. Several respondents voiced their complaints and frustrations about these incompatibilities:

I'd like less divergence between browsers!

I personally am sick of the disparate ways that all the differing browsers "choose to interpret" HTML.

The important thing is to be able to design once, and know that it can run anywhere, and not have to change everything if the ISP changes server platform on you, or if some company goes out of business.

Get the browsers to agree on compatibility issues.

It seems that I spend most of my time adjusting pages so they look even vaguely similar on a version 6.x Mac with Netscape and on a Windows95 PC running Explorer4. If this were true for any other publishing industry, the designers would have rebelled already. MORE COMPATIBILITY!

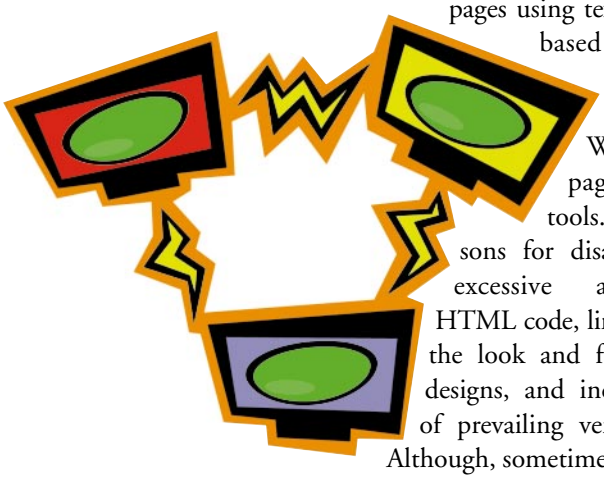
Browser incompatibilities can make a Web developer's life from just a bit annoying to totally frustrating. The Web authors need to know the differences in the browsers not only to design effective Web pages, but also to be able to make effective use of Web development tools. The authors now need to know what features of a Web development tool most browsers support before using them. Increasingly, Web development tools (especially WYSIWYG tools) provide better support for one browser over the other. For example, one could easily create a page with dynamic HTML using FrontPage '98. And, the page will work just fine when using Internet Explorer 4.0 but will not work when using Netscape Navigator 4. The same is true when using Cascading Style Sheets (CSS), which is implemented better on Internet Explorer 4.0 than on Netscape Navigator.

Summary: Profile of a Web Designer

From the results, we can state the following about this survey's prototypical Web designer:

- He or she has typically more than 2 years of experience designing Web pages.
- He or she designs pages mainly for corporation or personal use.
- Web page design is his/her primary task, one of several tasks required by the job and/or a hobby.

The Web author still hand codes Web pages using text editors or text-based HTML editors and is disappointed with the WYSIWYG Web page development tools. The main reasons for disappointments are excessive and redundant HTML code, limited control over the look and feel of Web page designs, and incomplete support of prevailing version of HTML. Although, sometimes, the Web author uses HTML conversion tools, he or she is generally unhappy with the HTML produced and uses text editors for postediting the converted Web pages. Also, the author designs his or her own graphics and is very satisfied with using Adobe PhotoShop.



Because of a variety of Web design projects being worked on, it is difficult for the Web developer to specify unambiguously how long it takes to develop a set of Web pages and how many pages are developed during that time period. Most of the Web page design projects, however, are very likely of very short time periods—less than 3 months. And, while working in teams of less than five people, the Web author probably develops about 11–50 pages during this period.

The Web developer uses design guidelines or style guides when developing Web pages, and the sources of these guidelines are those available on the Web as well as those developed in-house. Although the author designs pages that are optimized for faster download, he or she may not always optimize them to support people with disabilities. The Web designer evaluates the Web pages he or she designs for usability almost all the time, and his or her preferred methods of evaluation are informal methods (with colleagues and people in the development team) and user walk-throughs.

To publish Web pages, the Web author uses either FTP or copies files directly to the Web server. And, because several people in the Web development team need to publish on the Web server, he or she either solicits help of a Webmaster or publishes the information himself/herself.

Finally, the Web developer is quite unhappy with the browser incompatibilities and feels that he or she wastes considerable time making sure that the Web pages designed look good on at least Netscape Navigator and Internet Explorer. Despite misgivings about WYSIWYG Web page design tools, the Web author is looking forward to tools that give the necessary creative freedom and do not override hand-coded HTML pages.

Caveat

Although my interactions with Web designers of varied backgrounds lead me to believe that the sentiments expressed by the survey respondents are quite consistent with the general Web design community, the reader

should be careful in generalizing these results to all Web designers. The previous profile is based on a small number (138) of Web designers who chose to respond to the survey, and in that sense, they were self-selected. Furthermore, the groups from which responses were solicited may be biased in favor of those with backgrounds in human factors, graphic design, user interface design, and computer programming. GVU's 8th Survey found that most respondents had backgrounds in computer science/programming–related fields [2]. Because of these limitations, it is difficult to generalize the results to the entire population of Web designers.

Future Web Development Tools

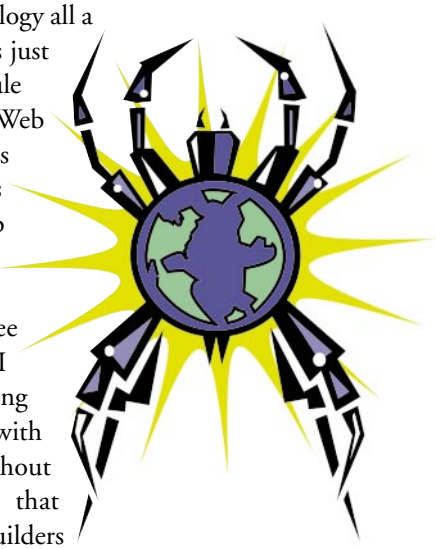
It was clear from the survey that most Web designers were unhappy with the current state of Web development tools, so what will make them happy? Based on the responses in this survey and personal experience, I will venture the following suggestions.

Cooperation Rather than Competition Between Vendors

Competition must give way to cooperation between vendors. I believe that today's competitive products are likely to become complementary working to support the deficiencies of each other. In fact, we see this trend today. Macromedia Dreamweaver, a WYSIWYG Web page design tool, completely supports Allaire's HomeSite and Bare Bones Software's BBEdit as an alternative means for developing Web pages. Similarly, NetObjects Fusion 3.0 (beta) supports third-party tools such as text-based HTML editors and includes Allaire's HomeSite, a text-based HTML editor, as part of the Fusion Package. Both of these tools even combine the two modes so that a Web designer can choose to work either in WYSIWYG design mode or in the text-based HTML mode. This allows Web designers to work the way they want, rather than adapting to the limitations and quirks of the Web development tool.

Increasing Support for Plug-In Architecture: Customizable Web Development Toolkits

Future Web pages will not be built by using only HTML. There will be increasing use of client-side scripting (ECMAScript: a standardized version of JavaScript), Dynamic HTML, CSS, eXtensible Markup Language, database integration, and yet unknown technologies. Web development tools will have to architect their products such that to take advantage of newer technology all a Web designers has to do is just plug in an upgrade module or a new tool to the Web development tool. This will allow Web designers to create customized Web development toolkits with each component of their choice; for example, see Wallop Build-IT [10]. I believe that even proposing such an open architecture with associated protocols (without actual plug-in modules) that Web development tool builders can use will be a significant improvement.



Support for Team Development

Future Web sites will very likely be several hundreds or thousands pages large. Development teams of 1–2 people will be a thing of past. To keep up with their Web sites, companies will push development responsibilities outside of developer teams, letting other departments contribute and manage Web content [4]. Future Web development tools will have to support such collaborative arrangements. This is not to suggest that such tools are not currently available. Vignette Corporation offers a product called Vignette, which is a team-based site-production development tool that allows writers, editors, producers, and Webmasters to collaborate in the content-development process and supports management of medium- to large-scale Web sites without significant performance penalties. Another tool just to enter the market is Net Objects TeamFusion, which also offers check-in/check-out facilities and allows creation of Access Control Lists to prevent changes by unauthorized persons.

Improved HTML Conversion Tools

With the advent of CSS, the conversion tools are likely to become "smarter" and more useful. User satisfaction is likely to increase because the conversion tool can put the style information separate from the main document and keep only structural HTML tags with the main document. Of course, success of such style sheet-based conversion will depend largely on how well (from the style sheet perspective) the original document was constructed. If the original document was not created using style sheets, abundant and redundant HTML code may not be avoidable. It is quite conceivable, however, to create a "smart" conversion tool that identifies the opportunities for using CSS where none exist. For example, the conversion tool can identify consistencies in formatting and suggest the possibility of creating a style-sheet based Web page.

Future Web Surveys

With the advent of new technologies, it will be useful to conduct a similar survey at least once a year to track the changes in development tools and understand the needs of Web designers and how well they are supported with the then available Web development tools. Also, it may be useful to conduct more detailed and independent surveys on areas such as Web development environment, accessibility, usability testing, publishing.

Acknowledgments

I would like to thank the following people:

- Kate Ehrlich and Michael Muller for giving me an opportunity to conduct this survey.
- Kate Ehrlich, Michael Muller, Scott Isensee, Valerie Stateson, and other reviewers for providing useful feedback on the earlier drafts of this article.
- Georgia Tech Research Center and GVU Center for allowing me to use their question on Web page types in the survey.
- Keith Instone for helping me put the Web version of the survey at the ACM SIGCHI site.
- Amy R. Fisher for forwarding the survey to the Chicago chapter of Webgrrls, Linda Tausher for forwarding the survey to WebWomen mailing list, Steven Champeon for forwarding the survey to Web Design mailing list, and JueyChong Ong for forwarding the survey to Worldwide Web Artists Consortium (WWWAC) mailing list.
- All 138 respondents for taking the time to respond to the survey and for providing insightful comments.

References

1. Gardner-Bonneau, D. (1998). Personal Communication.
2. GVU. *Graphics, Visualization, and Usability Center's 8th WWW User Survey.* World Wide Web: http:/www.cc.gatech.edu/gvu/user_surveys/, 1997.
3. Grose, E., Forsythe, C., and Ratner, J. In C. Forsythe, E. Grose, and J. Ratner (Eds.), *Human Factors and Web Development.* Lawrence Erlbaum Associates, Mahwah, NJ, 1998.
4. Hakhi, A. E. Content by committee – Development tools help business units collaborate on Web development. *Information Week* (January 12, 1998), Issue 664, 67–68.
5. Mosier, J. N., and Smith, S. L. Application of Guidelines for Designing User Interface Software. *Behaviour and Information Technology 5,* 1 (1986), 39–46.
6. Nielsen Media Research. Fall 1997 Internet Demographic Study from CommerceNet and Nielsen Media Research. World Wide Web: http://www.nielsenmedia.com/interactive/commercenet/F97/, 1998.
7. Nua. *Nua Internet Surveys.* World Wide Web: http://www.nua.ie/surveys/, 1998.
8. Trace Research and Development Center. *Trace Research & Development Center: Making information technology more usable for everyone.* World Wide Web: http://www.trace.wisc.edu/.
9. WAI (Web Accessibility Initiative). *WAI Accessibility Guidelines. Page Authoring.* World Wide Web: http://www.w3.org/TR/1998/WD-WAI-PAGEAUTH-0203, 1998.
10. Wallop Build-IT. *Wallop Software Home Page.* World Wide Web: http://www.wallop.com/.