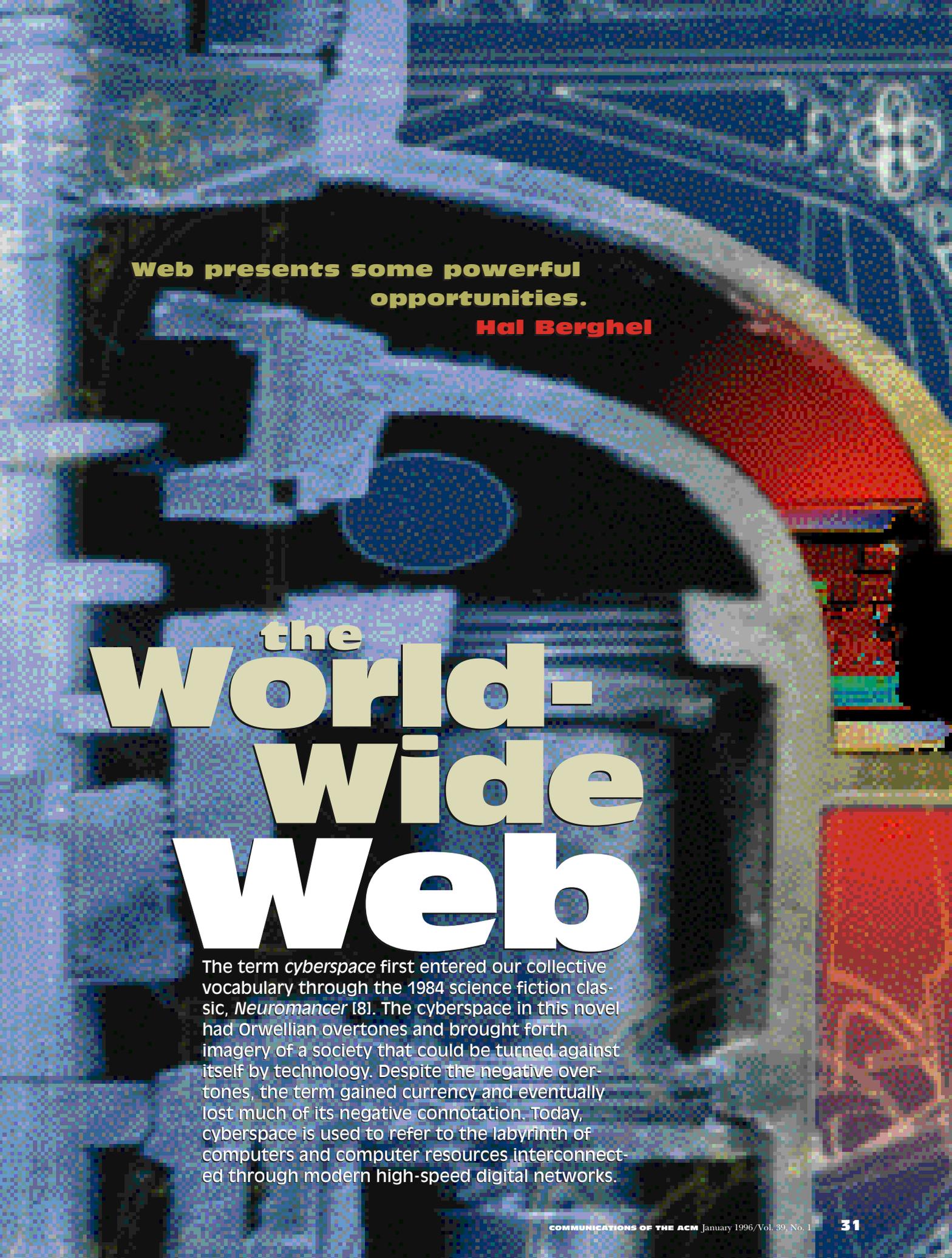


This overview of client

features

on the ever-widening

The
client's
Side
of



**Web presents some powerful
opportunities.**

Hal Berghel

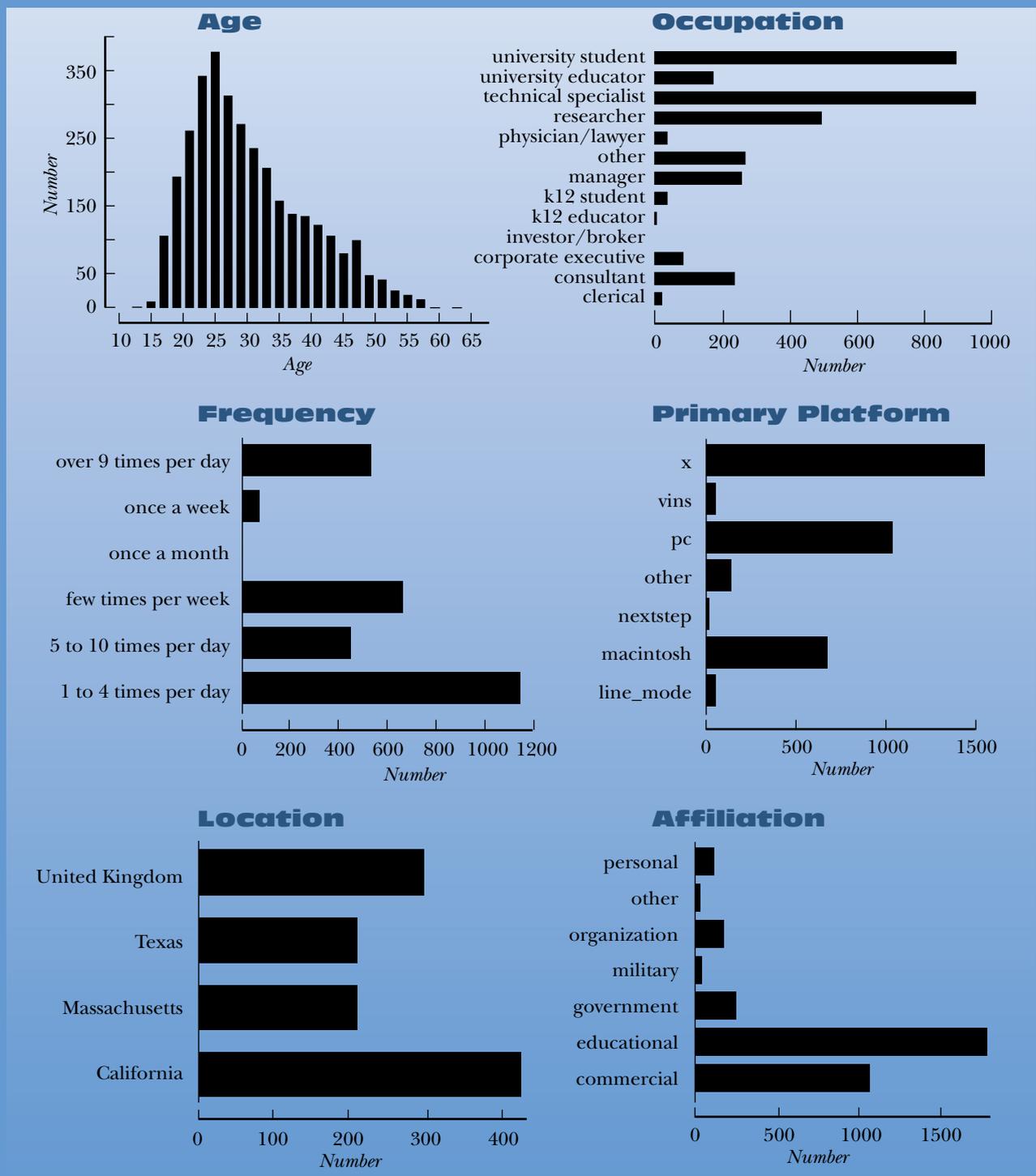
the World- wide Web

The term *cyberspace* first entered our collective vocabulary through the 1984 science fiction classic, *Neuromancer* [8]. The cyberspace in this novel had Orwellian overtones and brought forth imagery of a society that could be turned against itself by technology. Despite the negative overtones, the term gained currency and eventually lost much of its negative connotation. Today, cyberspace is used to refer to the labyrinth of computers and computer resources interconnected through modern high-speed digital networks.

The Client's

The “magic” of this second cyberspace, and the reason for its sudden and overwhelming popularity, is that it is both an enabling and unifying technology and a new, albeit technological, form of social interaction.

The enabling aspect follows from the virtual quality of cyberspace. The unifying aspect of cyberspace results from the fact that its physical infrastructure is a digital common carrier. In principle, anything that can be digitized can become part of the cyberspace. The versatility of this digital, packet-switched communication technology will make it possible to unify digital forms of all media. In just the past decade we have moved from digital text and graphics to rich text accompanied by sound and animation. Force is being added now. New dimensions of interactivity and participatory digital experience will follow.



The Web

Without question the hottest part of cyberspace at this time is the World-Wide Web. The Web is the Internet's "killer protocol" mix: the client-server handshaking and communications protocol, HyperText Transfer Protocol (HTTP), together with a protocol for document formatting, HyperText Markup Language (HTML). The versatility of the resulting tools have made the Web the focus of today's cybernauts.

The utility of the Web is well demonstrated by recent NSF backbone statistics. As this article was being written, the Web became the dominant Internet resource as measured by both packet count and volume, surpassing File Transfer Protocol (FTP), Telnet, and Gopher. In just six months, from December 1994 to April 1995, the Web moved from second place in number of packets moved (11.7%), just slightly behind FTP (19.6%) and ahead of Telnet (10.8%), to first place (21.4%) well ahead of both FTP (14%) and Telnet (7.5%) [9, 10].

According to the third World-Wide Web survey [11], Web users show a strong preference for general browsing (83%), or "surfing" as it has come to be known. Using the Web for entertainment (57%) came next, followed by work-related uses (51%). Not many Web documents are archived. Web browsers are now the primary tools for perusing non-Web documents, including Gopher and WAIS. It is interesting to note that Web volume over the NSFNET backbone surpassed WAIS in early 1993 and Gopher volume in early 1994 [6].

The third Web survey also provides us with a Web user profile. The mean age of Web users was 35. The age range was from 12 to over 75. Web users are generally North American (81%), male (82%), and many hold university degrees (36%). Noteworthy trends from the second Web survey taken six months earlier are that the typical Web user is slightly older, more likely to be from North America, and about as well educated. Median income is between \$50,000 and \$60,000. A breakdown by age, occupation, frequency, primary computer platform, location, and affiliation appears in Figure 1.

Web users also seem to prefer graphically-oriented homepages with meta-indexing and search capability. Multimedia browsing capability remains a low priority for most users, although this will certainly increase as multimedia-equipped workstations and desktop computers become the standard.

The Web from the Client Side

In the 18 months between the first and third Web surveys, the most significant changes had to do with the rapid growth of Web use in non-Mosaic and non-Unix environments. In the first survey, in late 1993,

97% of the respondents used Mosaic and 88% used Unix. By late 1994 when the second survey was conducted, these numbers shrank to 58% and 44%, respectively. By the time of the third survey in April 1995, these percentages fell to 3% [12] and 9%. The client side of the Web at this moment is dominated by Netscape's Mozilla and Microsoft Windows.

The dominance of the non-Unix desktop seems likely to continue to the turn of the century. For one thing, the non-Unix world represents the greatest growth potential for the client side. The few million existing Unix workstations pale in comparison to the estimated 60 to 75 million Windows computers and another 100 million that use DOS, not to mention the tens of millions of OS/2 and Macintosh computers. For the foreseeable future most of the growth in network connectivity will come from the PC arena since that is where the vast majority of connectable, though unconnected, potential customers reside.

Another factor, though less significant, is the fact that the widest range of robust Web clients is currently available for Windows. As Web users become more adroit, their expectations continue to increase and they will demand a wider range of functionality from their clients.

Developers creating products for the largest markets will have a decided competitive advantage. The greater potential return on investment will justify their best and most expeditious efforts. This became evident throughout 1994 with the emergence of the non-Mosaic commercial Web clients for Windows. These clients appear to be charting their own course in defining the cutting edge of Web client characteristics.

One client, Netscape, has even defined its own standard when it comes to HTML compliance. While the short-term effect of this independence seems to be positive because it encourages significant investment in innovation, the long-term effects are not so clear as it undermines the smooth evolution of standards. There is already some strain to be felt on the client side of the Web over HTML compliance and client compatibility [2].

The Web Clients

The two basic ingredients of a Web client are a navigation tool and a document browser. The primary function of the navigator is to facilitate the travel through cyberspace from one resource to another. The browser facilitates the perusal of information thus located. These two functions have become so integrated in modern Web clients that all navigator/browsers are now referred to by the single term, browser. However, it is instructive to remember that these two functions of navigator/browsers are operationally distinct.

When a navigator/browser is designed and engi-

Figure 1.
World-Wide Web user
profile (source [11])

neered well, cyber-browsing can be effortless and enjoyable. When not, cyberspace can appear populated with dead ends and potholes: unrecognized media, impenetrable firewalls, lost or misplaced document names and URLs, seemingly endless access delays and document swapping, window shuffling between client software and spawnable perusers, and so forth. Add to this list the bewildering effect of being lost-in-cyberspace and an ill-conceived browser can take much of the fun away from cybersurfing. Fortunately for the developers, Web surfing is novel enough that the typical end user displays considerable tolerance for abuse.

Figure 2.
Proxy client support for firewall penetration

In the discussion that follows we try to review navigator/browser technology from the point of view of client interface and usability. Our goal is to emphasize what we con-

sheer size of the potential Web client market motivates the developers to try to be the first out with important new features, so nothing stays the same very long on the commercial side. This will likely continue to be the case as long as there remains a large potential market for client software and services and a competitive market exists between several major developers.

For each category of features, we list a “yardstick” by which current products may be compared as of Summer 1995. This is a simple headcount approach to feature analysis and is neither an indication of what is technologically possible nor what is desirable.

Features of Navigator/Browsers Compliance Issues

Client/server compliance relates to the degree to which the operational characteristics of the client matches the expectations of the server to which it is connected. The

Web relies on two client-server protocols, HTML and HTTP, and so it is useful to divide the compliance issues into these two categories.

From the HTTP side, the highest priority is given to *access compliance*. This in turn relates to the environments that one might want to access via the Web. Popular environments are Gopher, WAIS, email, and FTP.

Connectivity (i.e., the breadth of communication environments supported by the client) may also be important. Many routinely use Ethernet and Token-Ring connectivity for direct access to the Web and SLIP, PPP, or X.25 for indirect connectivity via dial-up access through Internet service providers. Some may also add ASI, ODI, NDIS, and PDS for indirect connectivity via local-area networks (LANs).

Proxy client support enables the client to behave as if it is an intermediate foreign server with appropriate permissions so that it may gain passage through computer firewalls (see Figure 2). This is an important feature in industry.

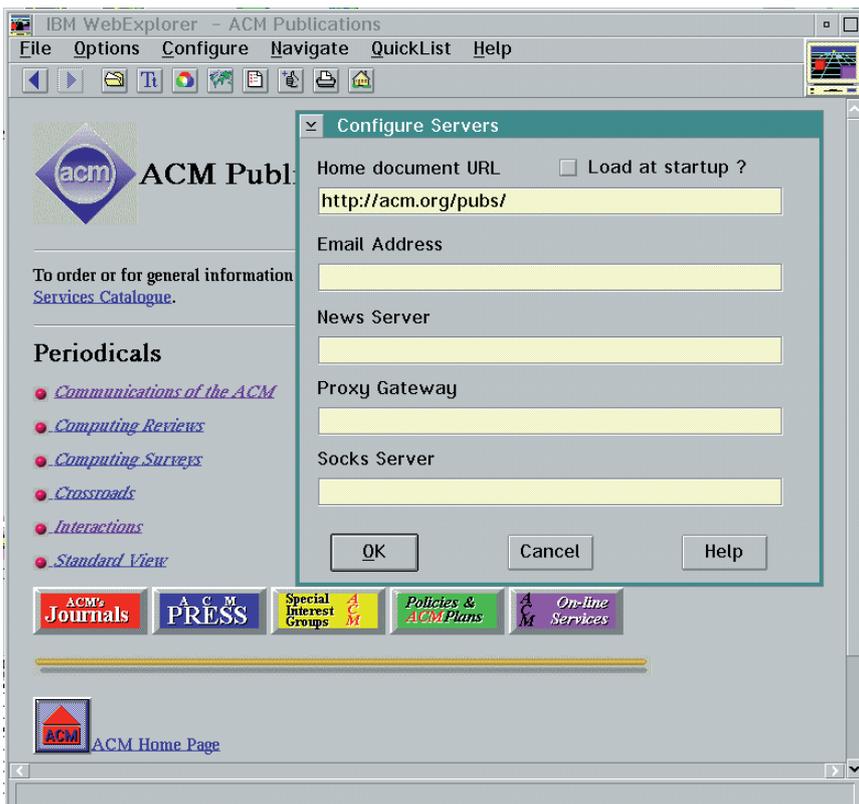
The second category of compliance deals with the HTML protocol. HTML is the “lingua franca” of the

Web. It defines what Web documents may look like and how Web resources may present themselves. The details of HTML are roughly organized by *HTML level*, which is a somewhat confusing mixture of standards and proposals promulgated by the World-Wide Web Consortium (<http://w3.org/>), with independent extensions proffered by the Netscape Corporation (<http://www.netscape.com/>).

One may organize HTML kernel specifications in terms of four levels. Level 0 would relate to specifications for basic HTML structure. Level 1 would

consider to be some of the more important features at this writing and to indicate how likely they are to appear in current Web navigator/browser clients. While we attempt to use standard definitions (such as there are) wherever possible, we will also define new terms that we feel better describe the underlying concepts.

To illustrate the utility of our feature analysis, we also compare a few of the current products—both commercial and freely available. At this moment the Web client topography is undergoing constant change as developers compete vigorously for market acceptance. The



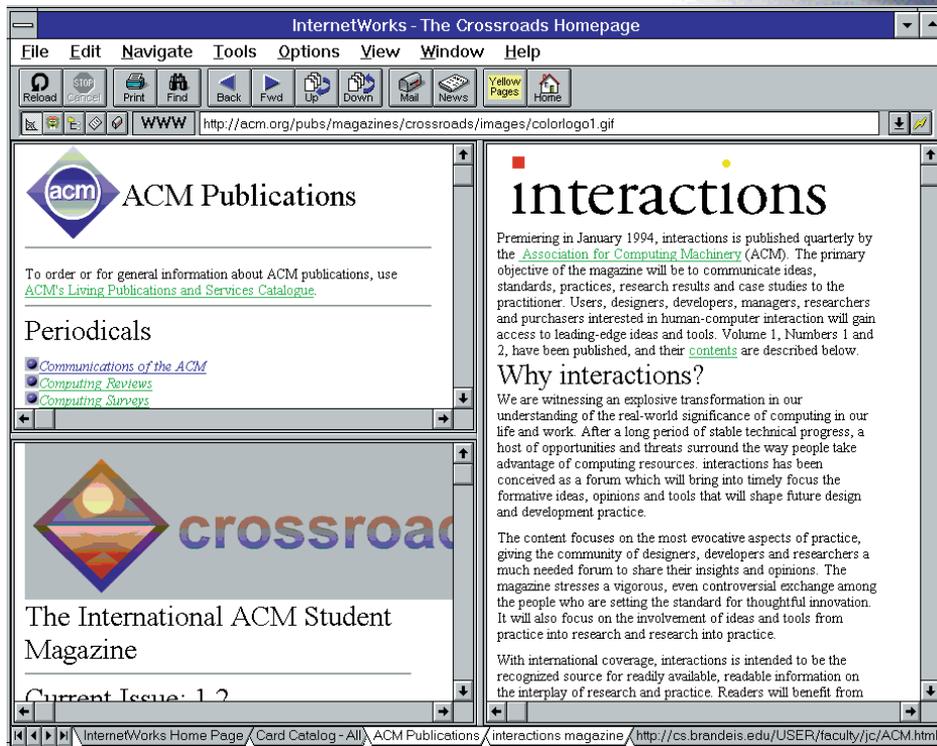


Figure 3.
A multithread-
ing client with
multipanining

define extensions for rudimentary image handling and limited text enhancement. Level 2 would include specifications for forms. Level 3 would provide extensions for tables, a LaTeX-like, ASCII-notation standard for mathematical formulas, and features for additional multimedia support. The HTML version 1 convention includes levels 0 and 1 standards. HTML versions 2 and 3 add to Version 2 the corresponding specification levels. In addition, there are significant extensions for document body proposed by the Netscape developers. These include, but are not limited to, standards for image alignment and resizing and the control of typesize. Web clients differ considerably when it comes to the finer points of HTML compliance.

At this writing, the more aggressive Web client developers are implementing at the HTML version 3 level. Significant differences between browsers [3] may be confirmed with such tools as our Web Test Pattern [2].

Also of critical compliance concern to the commercial side of the Web will be conformance to a received standard for *secure HTTP* transmission, which will involve such things as *authentication* (verifying the message source), *secrecy* (making the message unreadable to unauthorized users), and *message auditing* (creation of a message audit trail from sender to ultimate receiver through digital signatures and watermarks). All will receive increasing emphasis in the months to come.

YARDSTICK: Access compliance is still an issue although it is becoming less critical over time. The weakest link at the moment is Proxy client support, which is critical for those who wish to penetrate corporate firewalls.

Advances in proxy client support will be propelled by corporate and institutional concern for security as a result of the potential risks from *protocol spoofing* (where intruders use a proxy for which they are not authorized) and *hijacking* (where intruders seize the connection of a legitimate client). The evolving complexity of modern firewalls will inevitably make even authorized access more difficult. Where just a few years ago a firewall meant a packet filter on a router or bridge, modern systems are using entire computer systems called “bastions” as sentries of otherwise completely isolated networks. We predict the security issue and the complexity of the modern firewall will also contribute to the rapid advance of proxy technology.

On the HTML side, the robust client navigator/browsers will continue to evolve toward HTML level 3 compliance. Of course, there will inevitably be a level 4 standard that will offset short-term gains. For those who want a fully functional client, there is no escape from eternal vigilance.

Eventually, we expect browser technology to rival desktop publishing capabilities. This evolution will be propelled by such advances as electronic publishing and read/write Webbing. The current proliferation of inexpensive SGML editors will also push this along.

Performance Issues

Performance is becoming the most sought-after feature in modern clients, and one in which the leading developers are investing a lot of time and effort. Performance is critical for two reasons: the bandwidth bottleneck of the Internet and the lengthy load times for multimedia

Performance is critical for two reasons: *the bandwidth*

resources. The seasoned cybernaut will confirm that image, animation, and audio downloads introduce considerable delays in the access of Web materials.

Navigation convenience and efficiency varied widely between modern products until very recently. Techniques used to overcome many of the delays, and much of the discomfort, include the following.

Caching is the basic performance booster in Web products. Caching is a technique where visited documents or pages are retained on the local host so that the time-consuming reloads over the Internet may be avoided. A variation on the theme is the creation of *separate document and image caches* for efficiency reasons since the images are the greater time-drain. A further refinement involves a *cache manager* where the user can fine-tune the local host for maximum efficiency. In one or another form, caching is present in all modern Web navigator/browsers.

One thing to look for is whether the caching is hard or soft. *Soft caching* stays alive for the session. *Hard caching* stores the cache on disk. The problem with the latter is that it leads to what we shall coin “Web guano buildup.” NCSA Mosaic and its clones have been somewhat rude in this respect because the cache accumulated without notice up to a pre-defined maximum. In our view, this forces unnecessary housekeeping chores on the end user and the practice should be discouraged as unfriendly. At a minimum, there should be a way to toggle this feature off for those who don’t want to deal with it.

Even more important, yet less common, is *multithreading*. Multithreading supports multiple, concurrent Web accesses through multiple windows within a single Web client session (see Figure 3). This makes it possible to navigate and browse in several windows at once, perhaps while downloading for other windows takes place concurrently in the background. Its appeal lies in the fact that it enables users to take full advantage of whatever bandwidth they have available to them without launching resource-exhausting multiple sessions of the client navigator/browser. Multithreading will become a sought-after feature of Web clients just as it has in other desktop applications.

The remaining performance boosters tend to cluster around document loading. Some of these features, particularly those which relate to enhanced transfer and caching capabilities, can be quite important. One useful feature found in all modern clients is *load abort*, usually by clicking on some form of stop sign or throbbing, rotating or otherwise undulating icon (these seem to be in vogue at the moment). A refinement of this feature abandons the remaining load but retains what has been loaded up to the point of the abort so that the available links are usable.



Additional performance enhancers fall under the “background processing” rubric. *Dynamic linking* is one such feature that makes links operational even though the load cycle isn’t completed. *Deferred graphics loading* puts the graphics at the end of the load queue since they take the most time. *Image suppression*, which loads only text—kind of like having a built-in Lynx—is probably the most common feature after caching.

Another feature quite commonly used currently is *progressive image rendering*. This gradually fleshes out all of the images in the document as a group rather than each sequentially. Just a few passes usually suffices to render the images as recognizable so that their contribution to the document can be determined. Common progressive rendering techniques are *precise* or *focused interlacing*. Precise interlacing renders the images as they were produced by the interlacing software while focused interlacing displays the imagery in ever-clearer focus as the rendering progresses.

Finally, it should also be mentioned that some products now offer some type of *image enhancement*, which may use a combination of dithering, color palette reduction, compression techniques, or image interlacing to speed up the display.

YARDSTICK: Today the average Web client has some form of temporary caching with at least load abort and image suppression. Not all of the load controls are equally effective time savers. We find dynamic linking and progressive image rendering to be important, effective methods in dealing with the problem of load latency.

At this writing the most popular Web clients with multithreading are available for Microsoft Windows 95 and NT. The competitive advantage of these clients resides in their support of rapid, “parallel” navigation.

Reconfigurability

Software reconfigurability is the ability to change the look-and-feel of some aspects of the software to suit the situation or match the other native desktop applications.

Client reconfigurability may be quite important to some users. The only reconfigurability worthy of the name in our view is *menu-driven reconfigurability*, which is done interactively during a client session. However, some clients still support user-configurability through installation programs or by editing system or client configuration files offline. Those new to Webbing might want to investigate this, for the latter approach is definitely not for the neophyte or typical user.

bottleneck of the Internet and the lengthy load times for multimedia resources.

The things to look for in window-driven reconfigurability include *user-definable default homepages*, which enables the user to “boot” to any homepage of their choosing rather than stare at the developer’s advertising each time the client is loaded; *cache tunability* for balanced resource usage; a *kiosk* or full-screen mode that is especially useful for presentations because it enables the user to quickly move to a presentation mode, avoiding the distraction of the background desktop and/or client interface; and a sturdy configuration utility for *spawnable external multimedia perusers*.

YARDSTICK: In general, the current vendors seem to have taken a minimalist approach to client adaptability. Meager font and color modification of the display, and default homepage selection by editing system files, are about all that you can expect from the typical vendor.

Developers are just now beginning to take on a user-centered view of client configuration. For example, most commercial products now tend to shrink-wrap with local “welcome pages” as default, thereby creating a controlled cybersphere oriented toward their product offerings and services. Many of the “freely available” products still require the editing of system files to tailor the boot configuration. The marketplace will soon force the developers to offer user-friendly, menu-driven reconfigurability for either local or remote homepages. In the future, we expect competitive products will support reconfigurability in much the same way as modern word processing products.

Integration

Integration of the client software with the host desktop may be the last part of the Web client to mature in much the same way that it trailed behind the development of office desktop applications in the 1980s. If history is a good indicator, each of the client components will become more rigorous individually. Then, at some magic moment, the developers will integrate everything into a multimedia, virtual reality, megaprogram bonanza that will require 32MB of RAM (96MB recommended) and 330MB of available disk space. In the words of computer pioneer, Yogi Berra, “this will be *deja vu* all over again.”

The Web experience is so new that it is difficult to predict all of the features that will be integrated into tomorrow’s client. Among today’s necessities are the standard desktop and file management metaphors, cut, copy, and paste to clipboard and drag and drop file management.

Also important is sturdy support of non-native or *spawnable a/v perusers*. The importance of this support is that there is no way to predict future end-user demands for perusers. The computing community is

somewhat capricious when it comes to media formats. Today’s favorites may be tomorrow’s relics. There was a time not so long ago when GEM was a standard graphics format!

An even more dramatic example of the risks attendant upon those who choose to place all of their peruser eggs in one format basket is the recent experience with the Graphical Interchange Format (GIF). CompuServe made this format available to the graphics and network community in 1987. Since that time the format has achieved a leadership position.

While the format was placed in the public domain, the underlying algorithm was not. In reality, the Lempel-Ziv-Welch (LSW) algorithm that drives the lossless compression was patented in 1985 by Unisys and was actually being licensed to the telecommunications community. Once the popularity of the GIF format, particularly on the Internet, became sufficiently great, Unisys sought royalties from CompuServe. CompuServe in turn sought royalties from developers, which produced a developer/end user rebellion worldwide. No one can predict whether this will motivate the development of a new lossless compression scheme, or the acceptance of an existing lossy scheme as a substitute.

The one thing that is certain is that there are many slips between cup and lip in the media peruser business. Those clients that try to handle the formats internally may well place themselves at a considerable disadvantage to those who offer and maintain a versatile and non-exclusive launchpad for a wide range of third-party spawnable perusers.

The integration of a search engine is an important feature. Without search capability, surfing effectiveness declines as cyberspace grows. Search engines are low-level information “agents” that preview resources for content. Although the basic Web search engine is not very sophisticated at the moment, it does help with information filtering to a limited degree. An exception is the more powerful search environment supported by such Hyper-G software as Harmony and Amadeus [7].



Hyper-G broadens search capabilities in two ways. First, it supports searches within resource “collections” which extend beyond document and server boundaries. Second, it is designed to support a form of *virtual database organization* whereby resources distributed across the network appear as a unified whole (the database counterpart to virtual drive mounting through the Network File System). Whether this approach toward integrated searching

will be found to be necessary for generic Web products isn't clear because of the various uses of the Web by surfers with widely differing objectives.

In the future, even generic Web navigator/browsers are likely to support some rigorous form of searching. We predict that these will most likely take on the form of launchable "itinerant agents," which will evolve from today's wanderers in much the same way that the wanderers evolved from such Internet locator/indexers as Archie and Veronica.

Where today's wanderers (aka spiders, worms, crawlers, jump stations) collect URLs based upon

hypertext links, title keywords, document names and document contents, itinerant agents might collect abstracts and extracts of documents, gists or collages of images, or document "information chains" assembled from documents spread all over the Internet. At this writing, however, integrated searching support is rudimentary at best and limited to local searches of active documents (see Figure 4).

YARDSTICK: About all that you can count on in today's typical Web clients is seamless integration of native graphics viewers supporting common image

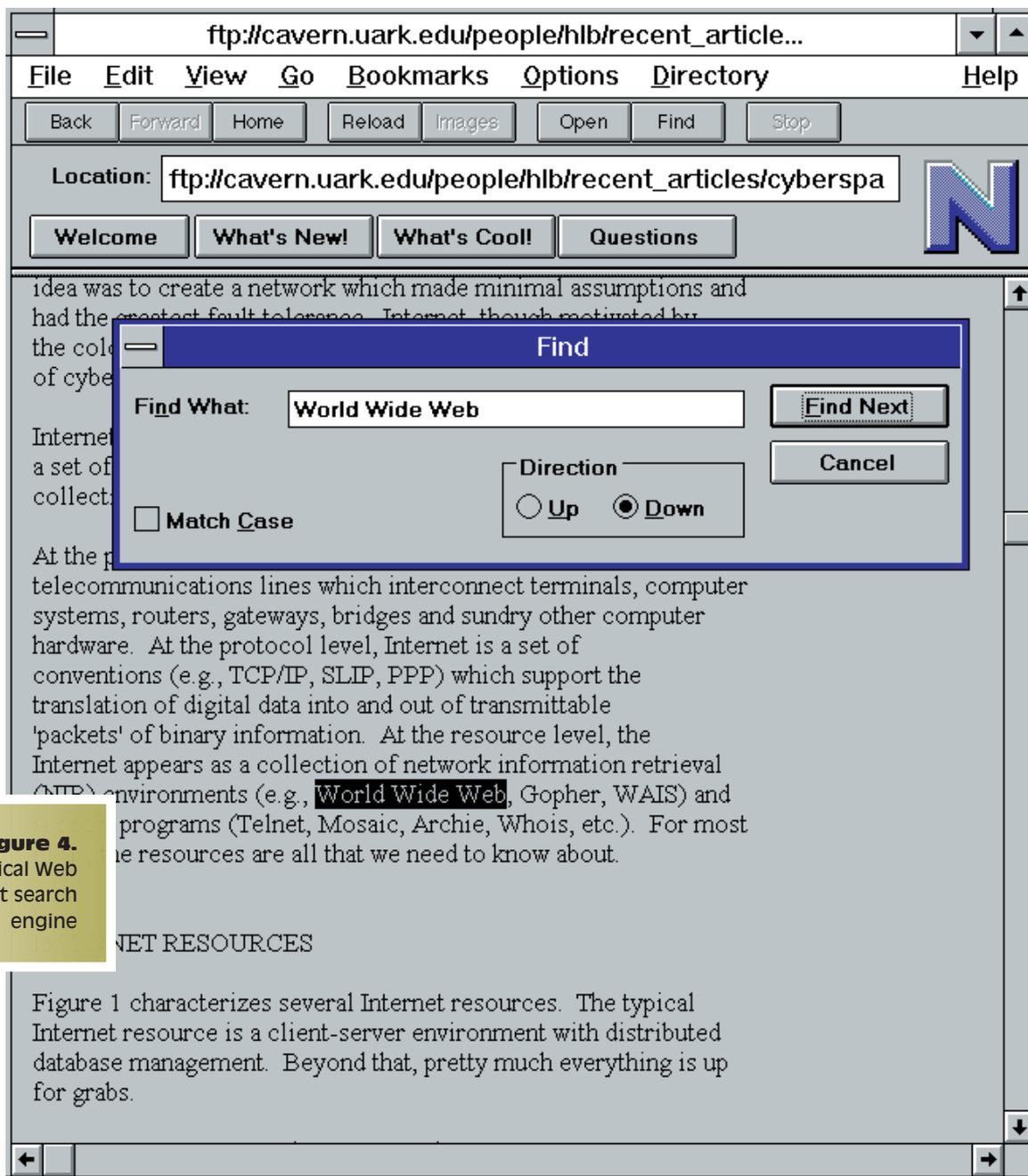


Figure 4.
A typical Web client search engine

Table 1. Navigator/browser feature comparison

	Cello v 1	NCSA X-Mosaic v 2.4	NCSA Mosaic (Win) v.20- alpha 3	Netscape (Win) v 1.0	Spyglass (Win) v 1.02	Air Mosaic (Win) v 3.06	Inter- net- works (Win) Beta 4	Win- Tapestry (Win) v 1.67	Web- Explorer (OS/2) v .91
COMPLIANCE									
proxy	+	-	-	+	~	+	+	~	~
extended html	-	-	-	+	-	-	-	-	-
PERFORMANCE									
multithreading	-	-	-	+	-	-	+	+	-
dynamic linking	-	-	-	+	-	-	+	-	-
deferred image	-	-	-	+	-	+	+	+	-
multi-pane/window	-/-	-/-	-/-	-/-	-/-	-/-	+/-	-/+	-/-
CONFIGURABILITY									
kiosk mode	-	-	-	-	-	+	-	-	+
external players	-	-	-	+	-	-	+	+	+
INTEGRATION									
d & d to clipboard	-	-	-	-	-	+	-	+	-
spawnable players	-	~	~	+	~	+	+	~	+
search engine	-	-	-	+	-	-	-	-	-
NAVIGATION AIDS									
hotlist/bookmark	b	h	h	b	h	h	h	b	h
folders/categories	+/-	-/-	+/-	+/-	-/-	+/-	+/-	+/+	-/-
menu/button bar	-	-	+	-	-	+	-	-	-
import/export	-/+	-/-	-/-	+/+	-/+	+/+	-/-	+/-	-/-
annotation	-	+	+	+	-	-	-	+	-
auto time/date stamp	-	-	-	+	-	-	-	-	-

LEGEND: + indicates that feature is supported in some form
 - indicates that *either* the feature is not supported *or* that we could not get it to function properly
 ~ indicates that the feature's support was weak by current standards

(JPEG, GIF, etc.), multimedia (MPEG, AVI), and audio (WAV, MIDI, AU, etc.) formats and primitive search capability for current document. Beyond that, everything is up for grabs. With regard to integration, we believe that the major deficiency today is absence of a general-purpose launchpad for external search engines such as spiders, wanderers, and worms.

Navigation Aids

Navigation aids are instruments that help reduce the cognitive loads associated with the process of navigation. The root of the problem is that the Web's cyberlinks don't scale well. The aggregate resources links grow exponentially with distance from the starting point. The relation between lack of scalability and such problems as the lost-in-cyberspace phenomenon and cyberchaos have been intensely studied and well-documented, particularly in the context of hypermedia systems [1, 13, 14].

Cyberlogs are itinerary histories (by document name rather than URL) of the recent surfing sessions sorted by date of visit with last visited, first. The logs are automatically created during navigation and subsequently displayable from the main menu. Clickable

entries reload the previously visited document. Cyberlogs obviously can't grow forever, but modern clients seem to retain them well beyond the time at which our interest wanes. This feature helps lessen the disorientation that frequently accompanies long cyber-journeys. We submit that cyberlogs will never become maximally useful until they are editable.

Hotlists and *bookmarks* offer an entirely different way of organizing URLs. These are the Web surfer's Rolodex. Where cyberlogs tell where we've been, hotlists list our favorite haunts. They are created by a mouse click on an "add to hotlist/bookmark" icon or menu item.

Hotlists and bookmarks, like cyber links, don't scale well either. Up to a point, perhaps 50 to 100 items, the non-scalability can be dealt with. Beyond that, the lists become unmanageable and awkward to use.

Two solutions to the non-scalability problem have appeared. One approach is to allow the user to collect URLs into *multiple hotlists* or *bookmark folders*. A refinement to this approach is the addition of cross-indexing by folder name and category. In any event, some organization and structure is required if the scalability problem is to be overcome.

Both hotlists and bookmarks may be annotated.

Annotation may be by comment or time/date stamp. Both require a good editor to be effective. An automated time/date stamp may also be an effective tool in eliminating stale URLs that proliferate like wire coat hangers in a closet. With this feature, the client deletes all URLs that fail the test of recency.

In our view one of the most sorely needed hotlist/bookmark support tools is a general-purpose import/export utility that allows the user to save hotlists from one client and import them into another. The value of this utility will increase as long as there are many different clients in use.

YARDSTICK: Cyberlogs are to be expected from every product. Current products differ widely in terms of the quality of folder management. Automated time/date annotation is still uncommon. Some import/export capability is present in today's better clients, but it is usually limited in scope.

Conclusion

Table 1 compares several current Web navigator/browsers in terms of some of the features discussed. We have emphasized Windows products because they represent the largest Web user community at present.

We provide this comparison only to illustrate the differences that exist between Web products as of this writing (August 1995).

The comparison is not intended to be complete. Some differences may not be important in certain applications. Others may be critical. The intent of Table 1 is to encourage Web users to investigate the capabilities of clients before acquiring them, for substantial differences exist. The appropriate slogan is caveat emptor.

We limited our discussion to features of mainstream Web navigator/browser clients. The two Hyper-G clients, Amadeus and Harmony, were not included because they represent a significant departure from the traditional navigator/browser aims of today's Web clients. Hyper-G is a very different type of client and one that deserves really special consideration.

It should also be noted that many Web navigator/browser clients come bundled in a "suite" of Internet utilities. NNRP-compliant news readers, SMTP or POPx emailers, NFS resource-sharing software, and so forth, may all contribute significantly to the overall usability of the client in particular settings. However, since these utilities are not, strictly speaking, Web clients, we omitted them from consideration. Much the same could be said of today's standalone wanderers, spiders, and worms. Soon these will be spawnable from within the Web client.

In this article we have attempted to discuss the

client side of the Web in an informative and purposeful way. We hope that this overview of Web client features will help focus attention on deficiencies and strongpoints that may help you select the navigator/browser clients best suited to your needs.

Acknowledgments

We wish to express our appreciation to Dennis Bouvier, Jacques Cohen, Robert Inder, Marcus McGuire, Susan Mengel, Dave Oppenheim, Leon Sterling, Gio Wiederhold and the five anonymous referees for helpful comments and suggestions on earlier drafts of this article. Special thanks to James Pitkow for making some unpublished data from his third Web Survey available to us, and to John December for sharing some of his unpublished work on HTML specifications. ■

References

1. Barrett, E. *Text, Context and Hypertext*. MIT Press, Cambridge, Mass., 1988.
2. Berghel, H. Using the WWW test pattern to check HTML client compliance. *IEEE Comput.* 28, 9 (Sept. 1995), 63–65. The Web test pattern URL is <http://www.uark.edu/~wrg/>.
3. Berghel, H. OS/2, Unix, Windows and the Mosaic War. *OS/2 Magazine* (May 1995), 26–35.
4. Berghel, H. and Berleant, D. The challenge of customizing cybermedia. *Heuristics: The Journal of Knowledge Engineering* 7, 2 (1994), 33–43.
5. Berleant, D. and Berghel, H. Customizing information: Part I—*IEEE Comput.* 27, 9 (Sept. 1994), 96–98; Part II—*IEEE Comput.* 27, 10 (Oct. 1994), 76–78.
6. Berners-Lee, T., Cailliau, R., Luotonen, A., Nielsen, H., and Secret, A. The World-Wide Web. *Commun. ACM* 37, 8 (Aug. 1994), 76–82.
7. Fenn, B. and Maurer, H. Harmony on an expanding net. *Interactions* 1, 4 (1994), 29–38.
8. Gibson, W. *Neuromancer*. Ace Books, New York, 1984.
9. Merit NIC Services. NSFNET Statistics, December 1994. URL=<gopher://nic.merit.edu:7043/11/nsfnet/statistics/1994>.
10. NSFNET Backbone Traffic Distribution Statistics. April 1995. <http://www.cc.gatech.edu/gvu/stats/NSF/merit.html>.
11. Pitkow, J., et al. The Gvu Center's Third WWW user survey. URL=[http://www.cc.gatech.edu/gvu/user_surveys/survey-04-1995/\(1995\)](http://www.cc.gatech.edu/gvu/user_surveys/survey-04-1995/(1995)).
12. Pitkow, J. Personal communication, 1995.
13. Rivlin, E., Botafogo, R., and Schneiderman, B. Navigation in hyperspace: Designing a structure-based toolbox. *Commun. ACM* 37, 2 (Feb. 1994), 87–96.
14. Yankelovich, N., Haan, B., Meyrowitz, N. and Drucker, S. Intermedia: The concept and construction of a seamless information environment. *IEEE Comput.* 21, 1 (1988).

About the Author:

HAL BERGHEL is a professor of computer science at the University of Arkansas. **Author's Present Address:** CSCI/SCEN 230, University of Arkansas, Fayetteville, AR 72707; email: h1b@acm.org; URL: <http://www.acm.org/~h1b/>

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.