Mosaic and the World-Wide Web

Ronald J. Vetter
North Dakota State University

Chris Spell and Charles Ward
University of North Carolina at Wilmington

The World Wide Web, one of the newest information services on the Internet, uses hypertext links to other textual documents or files.1 With this method, users can click on a highlighted word or words in the text to provide additional information about the selected word(s). Users can also access graphic pictures, images, audio clips, or even full-motion video through hypermedia, an extension of hypertext.

The World Wide Web project was started by CERN (European Center for Nuclear Research) in an attempt to build a distributed hypermedia system.2 Users access WWW through a “browser program” that can read and fetch documents locally as well as from sites around the world via the Internet. Browsers access files using FTP, NNTP, Gopher, and several other Internet-based protocols (see the glossary of terms). Browser clients can also search remote documents and databases if the associated server at that site has built-in search capabilities.

Browsers can be line-oriented (to work with traditional terminals) or graphics-oriented (to work with more modern graphics display terminals). One of the most popular graphics-oriented browsers is Mosaic, which was developed at the National Center for Supercomputing Applications (NCSA) as a way to graphically navigate the WWW. The first version of NCSA’s Web browser, a WWW client for Unix with X Windows called XMosaic, was made available to the Internet community in 1993. Due primarily to its easy-to-use graphical user interface (GUI), XMosaic soon became the most popular interface to the Web.3

Mosaic is a system for wide-area distributed asynchronous collaboration and hypermedia-based information discovery and retrieval. Mosaic browsers are currently available for Unix workstations running X Windows, PCs running Microsoft Windows, and Macintosh computers. Mosaic can access data in WWW servers, Wide Area Information Servers (WAIS), Gopher servers, Archie servers, and several others. Figure 1 shows an example Mosaic screen (the home page) from the University of North Carolina at Wilmington.

Many commercial and government organizations and universities are migrating to Mosaic because of its many useful features and capabilities.4 It has a consistent mouse-driven graphical interface and can display electronic data in a variety of fonts and styles. Mosaic reads SGML (Standard Generalized Markup Language) and...
Glossary of terms

**Archie** — A system that allows searching of indexes of files available on public servers by anonymous FTP on the Internet.

**CERN** — European Center for Nuclear Research.

**FTP** — File Transfer Protocol.

**GIF** — Graphics Interchange Format.

**Gopher** — A distributed information service that makes available hierarchical collections of information across the Internet. Both the server and client software are available as public domain software from the University of Minnesota.

**HTML** — HyperText Markup Language.

**HTML+** — HTML extension that supports new features.

**HTTP** — HyperText Transfer Protocol, the native WWW protocol.

**ISO** — International Standards Organization.

**Mosaic** — A system for wide-area distributed asynchronous collaboration and hypermedia-based information discovery and retrieval. Mosaic browsers are high-level graphical user interfaces (GUIs) currently available for Unix workstations running X Windows, PCs running Microsoft Windows, and the Apple Macintosh.

**NCSA** — National Center for Supercomputer Applications.

**NNTP** — Network News Transfer Protocol.

**PPP** — Point-to-Point Protocol, which allows a computer to use the TCP/IP protocols with a standard telephone line and a high-speed modem. PPP is replacing SLIP (see below) as a standard.

**SGML** — Standard Generalized Markup Language.

**SLIP** — Serial Line Internet Protocol, which allows a computer to use TCP/IP (Internet) protocols with a standard telephone line and a high-speed modem.


**Telnet** — Lets users connect to any remote system on which the user has an account.

**URL** — Uniform Resource Locator.

---

**HTML and HTML+**

The Web uses HTML for creating and recognizing hypermedia documents. This is a simple markup system, related

![Figure 1. Sample Mosaic screen for the home page of the University of North Carolina at Wilmington.](image)
to the Standard Generalized Markup Language (SGML), used to format hypertext documents. (On the Web, these documents usually carry the "html" suffix.) HTML code describes how textual elements (like paragraphs, lists, numbered and bulleted lists, descriptive lists, and quoted paragraphs) will be displayed. The current HTML standard supports basic hypermedia document creation and layout, but this capability is still very limited. The HTML+ extension supports interactive forms, defined "hotspots" (hyperlinks) in images, formatted tables, and more versatile document layout, formatting options, and styles.

Writing HTML documents. Most HTML documents are written using a standard text editor — for example, vi in a Unix environment — in a plain text format. Although HTML code is a tedious language to write, some authoring tools can now convert plain text to HTML automatically.

One way to learn how to write HTML documents is to look at how others have created their documents. You can do this by using the "source" button (available on most Mosaic browsers) to view the HTML code for a document or page you find particularly interesting. Most HTML statements are fairly straightforward. For example, Figure 2 shows the HTML code for the Mosaic screen image depicted in Figure 1. The code corresponds closely to the final graphic document. If you intend to write HTML code, we recommend that you read the on-line Mosaic document "A beginner's guide to HTML," which is available on the Web (see sidebar on HTML editors and converters).

HTML editors. HTML documents can be created with WYSIWYG (what you see is what you get) editors or with editors that simply assist you by allowing you to select the desired markup tags from a menu. HTML editors are easier to use and interactively display the resulting HTML document on-screen. For example, current versions of the Emacs editor have an "HTML mode" to assist users with writing statements. There is also an editor for Microsoft Windows called HTML Assistant. For X users, tXWWW supports WYSIWYG HTML editing; and since it is a browser, you can try out links immediately after creating them. HotMetal is a professional HTML Plus editor for X and Microsoft Windows. For Macintosh users, the BBEdit HTML extensions allow the BBEdit and BBEdit Lite text editors to conveniently edit HTML documents.

Navigating the Web

Information currently accessible through the Web includes information served through Telnet, Gopher, WAIS, FTP, Usenet news, Archie, and anything in the form of Unix man pages or hypermedia documents. Mosaic lets users effectively navigate the Web and access the
HyperText Transfer Protocol. HTTP is the protocol employed between the server and client. It requires only a reliable connection-oriented transport service, typically TCP/IP. The client establishes a connection with the server and sends a request containing the word GET, a space, and the partial URL of the node to be retrieved, terminated by the carriage return and line feed characters. The server responds with the node contents, which consist of HTML text documents. The end of the contents is signaled by the server closing the connection.

Ways to spin a Web. Creating a Web can be as simple as pointing a hyperlink from an existing Web document to your existing data. Making it public involves running the FTP or HTTP daemon. Any file available by anonymous FTP can be immediately linked into the Web. This method requires only a small start-up effort and allows contributions to be made network accessible via the Internet in a very short period of time.

You can also create a Web by writing your own HTML code and pointing it to databases of interest. Data may come from local archives or be stored elsewhere on the Internet. This requires familiarity with the HTML language and the information resources currently available on the Internet. Existing files (for example, a LaTeX document) can be converted to HTML automatically by using a set of special software tools.

When creating Web documents for public access, we have found the following guidelines to be useful.

- Remember that most users accessing information over the Internet are browsing and are therefore unwilling to wait long for a document to appear on their screen. You should use graphics only to catch the reader’s attention. Keep the size of images on the home page small; they should be used as simple icons that show what the object points to. Wait until the user selects the icon to incorporate high-quality (and large) images into your documents. This will let users access the home page quickly and will ease server and network load. Although each image takes time to process and slows the display, using a particular image multiple times in a
document causes very little performance degradation compared to using it once.

- Documents transmitted over a network will not display as quickly as documents will locally, where you typically develop them. Before making any document public, be sure to test the document’s remote access time to ensure that it is reasonable.

- Always provide an estimate of the number of bytes that will be sent when a user selects a given “hotword” (especially for high-resolution images and digital audio and video). This helps users decide whether to select a given hotword.

- Do not mix a large number of different fonts, colors, and so forth on a single display screen. Use good graphic arts design principles. This is especially important if the screens will be used for educational purposes and projected onto an overhead screen. For example, certain colors that look good together on screen and are easily distinguishable may not be so distinguishable when sent through a projection panel and displayed with an overhead projector. We have provided some guidelines to help developers of multimedia presentations avoid some of the common problems associated with the projection of these materials.

- Do not say “Click Here.” It is redundant and only clutters the screen. Hotwords are normally a different color or font, so it is obvious (especially to frequent users) that “clicking here” will cause a hyperlink to be traversed.

- Put a reference (for example, an electronic mail address) at the bottom of every page telling whom to contact about the document. This can be accomplished by using the HTML Address Tag construct.

- Do not make your pages too long. Rather, create a hierarchical set of menus that group items into meaningful categories. That way only those items of interest will have to be selected.

How does the WWW compare to Gopher and WAIS?

While all three of these information presentation systems are client-server based, they differ in terms of their data models. In Gopher, data is either a menu, a document, an index, or a Telnet connection. In WAIS, everything is an index, and everything that is returned from the index is a document. In WWW, everything is a hypertext document, which may be searchable if it contains text. This means that WWW can represent Gopher and WAIS data models as well as provide extra functionality. Only WWW has hyperlinks.

Although text is supported by all three systems, WWW provides facilities for displaying “richer” text (for example, headings, lists, and emphasized text) in a standardized way. Image, audio, and video data are all supported by external viewers (for example, MPEGplay). There is little direct support for application-specific data in any of the three systems, and all use different methods for expressing type and encoding. Also, all three systems have little support for controlling the presentation of non-text data. For example, backdrops and synchronization in time are not supported, and buttons are only supported in a limited way. (See Adic for a more comprehensive evaluation.)

Limitations. Perhaps the greatest limitation of current distributed hypertext systems is that there is no simple way to find out what information has changed, what information is new, or even what information is out there on the Internet. Mosaic users commonly complain about difficulties finding resources on a particular topic or subject. Unless the user has extensive knowledge of the Internet, it is hard to locate resources of interest on the WWW. A related problem is remembering where resources were found when trying to access them again (hotlists are useful, but it is better to annotate the location and description of the resource on a notepad or set of index cards so that it can later be recalled). Another problem is that there are simply too many links,
<table>
<thead>
<tr>
<th>Operating system</th>
<th>Program</th>
<th>FTP Availability</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td>Cello</td>
<td>ftp.law.cornell.edu/pub/LII/cello</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mosaic for Windows</td>
<td>ftp.ncsa.uiuc.edu/PC/Mosaic</td>
<td></td>
</tr>
<tr>
<td>Macintosh</td>
<td>Mosaic for Macintosh</td>
<td>ftp.ncsa.uiuc.edu/Mac/Mosaic, info.cern.ch/pub/www/bin/mac</td>
<td>The browsers listed here are native NextStep applications. Next systems can also run X-based browsers using one of the widely used X-server products for the Next.</td>
</tr>
<tr>
<td>Amiga</td>
<td>AMosaic</td>
<td>max.physics.sunysb.edu/pub/amosaic</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>The browsers listed here are native NextStep applications. Next systems can also run X-based browsers using one of the widely used X-server products for the Next.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Allows WYSIWYG hypertext editing. Requires NextStep 3.0.</td>
</tr>
<tr>
<td>X/Dec-Windows</td>
<td>Mosaic for X</td>
<td>ftp.ncsa.uiuc.edu/-Web</td>
<td>Unix browser using X11/ Motif; compiled for several popular platforms.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tkWWW Browser/</td>
<td>harbor.ecn.purdue.edu/pub/tcl/extensions/tkwww (followed by an extension dependent on the current version)</td>
<td>Editor for X11 supports WYSIWYG HTML editing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Chimera</td>
<td>ftp.cs.univ.edu/pub/chimera</td>
<td>Unix/X Browser using Athena (doesn't require Motif).</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Line Mode Browser</td>
<td>info.cern.ch/pub/www/src</td>
<td>This program gives WWW access to anyone with a dumb terminal. A general-purpose information retrieval tool.</td>
</tr>
<tr>
<td></td>
<td>Lynx full-screen</td>
<td>ftp2.cc.ukans.edu/pub/WWW/lynx</td>
<td>This hypertext browser for VT100 terminals uses full browser screen, arrow keys, highlighting, etc.</td>
</tr>
<tr>
<td></td>
<td>perl/WWW</td>
<td>archive.cis.ohio-state.edu/pub/w3browser</td>
<td>A tty-based browser written in Perl.</td>
</tr>
<tr>
<td></td>
<td>VMS client</td>
<td>vms.huji.ac.il/www/vms_client</td>
<td>A full-screen client based on VMS's SMG screen management routines.</td>
</tr>
</tbody>
</table>
Obtaining your own browser. The preferred method of accessing the Web is to run a browser yourself. Browsers are available for many platforms, both in source and executable forms. Table A lists some available browsers (a more extensive listing can be found at http://info.cern.ch/hypertext/WWW/Clients.html).

Setting up your own server. Browsers can obtain data from programs supplied by information providers. These programs can either be WWW servers that understand the HyperText Transfer Protocol, “gateway” programs that convert an existing information format to hypertext, or a non-HTTP server that WWW browsers can access — anonymous FTP or Gopher, for example. To learn more about WWW servers, you can consult a WWW server primer available at http://www.vuw.ac.nz/whon/ Nathan.Torkington/ideas/www-servers.html. Table B lists some servers currently available for Unix, Macintosh, Microsoft Windows, and VMS systems. (For more information on writing servers and gateways in general, see http://info.cern.ch/hypertext/WWW/Daemon/Overview.html.)

Creating a public home page. There are several things you can do to publicize your new HTML server or other offering:

- Post it to the newsgroup comp.infosystems.announce. Please read the newsgroup first to get a feel for the contents. You can also crosspost to comp.infosystems.www.
- Submit it to the maintainers of various catalogs, such as the WWW Virtual Library http://info.cern.ch/hypertext/DataSources/bySubject/Overview.html.

Reference


<table>
<thead>
<tr>
<th>Operating System</th>
<th>Program</th>
<th>FTP Availability</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unix</td>
<td>NCSA httpd</td>
<td>ftp.ncsa.uiuc.edu/</td>
<td>Serves WWW and Gopher clients. A good server for migrating from Gopher to WWW, although lacking the NCSA and CERN servers’ script capabilities.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Web/ncsa_httpd</td>
<td></td>
</tr>
<tr>
<td></td>
<td>CERN httpd</td>
<td>info.cern.ch/pubs/</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>www/src.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>GN Gopher/HTTP server</td>
<td>URL <a href="http://hopt.math.nwu.edu">http://hopt.math.nwu.edu</a></td>
<td></td>
</tr>
<tr>
<td>Macintosh</td>
<td>MacHTTP</td>
<td><a href="http://www.uth.tmc.edu/mac_info/machttp_info.html">http://www.uth.tmc.edu/mac_info/machttp_info.html</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td>NCSA httpd (Windows)</td>
<td>ftp.ncsa.uiuc.edu/Web/ncsa_httpd/whipt11a6.zip</td>
<td>The NCSA server has most features of the Unix version, including scripts.</td>
</tr>
<tr>
<td>VMS Servers</td>
<td>CERN HTTP</td>
<td><a href="http://delonline.cern.ch/disk5user/duns/doc/vms/distribution.html">http://delonline.cern.ch/disk5user/duns/doc/vms/distribution.html</a></td>
<td>A port of the CERN server to VMS.</td>
</tr>
<tr>
<td></td>
<td>Region 8 Threaded HTTP Server</td>
<td><a href="http://kogl1.eng">http://kogl1.eng</a> ohio-state.edu/www/doc/serverinfo.html</td>
<td>A native VMS server that uses DEC-threads.</td>
</tr>
</tbody>
</table>
as Hall has pointed out in a recent article. Users of distributed hypermedia systems are often overwhelmed by the large number of possible links and become disoriented when moving between different application packages and information servers.

Steps are being taken to address some of these problems in Mosaic. Several resources now provide information on new and established Mosaic servers by topic. For example,

- The WWW Virtual Library is a good place to find resources on a particular subject, and
- What's New With NCSA Mosaic carries announcements of new servers on the Web.

More recently, several software tools have been developed that create automatic indexes you can search. Two such tools are WebCrawler (http://www.bio-tech.washington.edu/WebQuery.html) and WWW Worm (http://www.cs.colorado.edu/home/mcbrayn/WWW.WWW.html). WebCrawler indexes the contents of the documents, whereas WWW Worm builds its index based on page titles and URL contents. Although tools that automatically traverse the Web are useful, tools that recursively search for information may not notice self-referencing links and may begin to return an infinite number of indexes.

**Popularity.** The WWW is gaining popularity as evidenced by the amount of network traffic (in bytes) across the National Science Foundation's North American Network (NSFNet). In fact, WWW traffic recently surpassed Gopher traffic, according to statistics produced by monitoring traffic on the Internet backbone. (Since WWW browsers can also access Gopher servers, the Gopher traffic may be somewhat inflated.) From January to August 1993, the amount of traffic attributed to Web use multiplied by 414 times, and usage of the server at CERN doubled every four months—twice the rate of Internet expansion. The Web is now ranked eighth of all network services in terms of sheer byte traffic as of May 1994 (Gopher is ranked tenth).

---

**Extending Mosaic to support customized views**

We are building an interface agent for Mosaic that helps users find the various types of Internet data available on the WWW. Our approach is similar to that undertaken by others who have developed autonomous agents to provide personalized assistance with meeting scheduling, electronic mail handling, electronic news filtering, and selection of entertainment. The idea is to augment Mosaic with an intelligent interface agent that can learn the particular interests, habits, and preferences of individual users and then help them obtain the desired information in a way that reduces the number of links offered to the user. We accomplish this by supplying the interface agent with domain-specific background knowledge about user preferences and interests. At runtime, the interface agent uses this knowledge to find and recognize information of interest to the user. The agent must also be capable of acquiring knowledge on its own and building appropriate rules and facts in the knowledge base by monitoring user behavior over time (a form of machine learning).

Our approach aims at having Mosaic customize the link-following processes to suit a user's unique interests, habits, and access patterns. The system customizes the hypermedia environment supported by Mosaic by creating links dynamically, using rule-based algorithms or neural networks. This approach is much more flexible than many of today's closed hypermedia systems and still provides the user with the power to ask for more information when needed rather than statically displaying all available links on a particular topic every time.

Figure 3 depicts the system model we propose. Initially, users (and agents) store information about particular interests in the knowledge base. The rules and facts are used as input to the HTML Language Generator. User-specific HTML code (and links) are then generated at runtime. This requires communication with other Mosaic agents and servers (much like Archie servers are used) to find the relevant information on a particular topic or subject.

The goal of our work is to demonstrate that an intelligent interface agent can be used to satisfactorily limit the number of links offered to the user while simultaneously providing users with a more effective interface to the WWW.

The World-Wide Web is still evolving at a rapid pace. Clearly, distributed hypermedia systems on the Internet will continue to be an active area of development in the future. We believe that the flexibility of the WWW design, its use of hyperlinks, and the integration of existing WAIS and Gopher information resources, make the WWW ideal for future research and study. We also realize that highly interactive multimedia applications will require more sophisticated tools than currently exist. The most significant issue that needs to be resolved is the mismatch between WWW system capabilities and user requirements in the areas of presentation and quality of service. The lack of adequate directory services will continue to plague WWW users, and future work on distributed hypermedia systems will need to address these issues.
References

Ronald J. Vetter is an assistant professor in the Department of Computer Science and Operations Research at North Dakota State University in Fargo, His research interests include high-performance computing and communications, multimedia systems, and high-speed optical networks.

Vetter received the BS and MS degrees in computer science from North Dakota State University in 1984 and 1986, respectively, and his PhD in computer science from the University of Minnesota, Minneapolis, in 1992. Vetter is a member of the ACM and the IEEE Computer and Communication Societies.

Christopher G. Spell is the system administrator in the Mathematical Sciences Department at the University of North Carolina at Wilmington. His research interests include operating systems, communications, and multimedia systems. Spell received his BS in computer science from UNCW in 1993.

Charles R. Ward is assistant vice chancellor for academic affairs and professor of chemistry at the University of North Carolina at Wilmington. His research interests include the application of multimedia technology in chemical education.

Ward received the BS degree in chemistry from Manchester College in 1970, and the MS degree in chemistry from Purdue University in 1975. He received the PhD degree in science education from Purdue University in 1978. He is a member of the American Chemical Society and the Association for Applied Interactive Media.

Readers can contact the authors through Ronald Vetter, North Dakota State University, IACC Bldg., rm. 258, Fargo, N.D., 58105; e-mail vetter@plains.nodak.edu.

October 1994

Call for Papers

**WET ICE 95**

Fourth IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises
April 20–22, 1995
Berkeley Springs, West Virginia

The Concurrent Engineering Research Center (CERC) at West Virginia University is seeking submissions for WET ICE 95, which is sponsored by the IEEE Computer Society. Papers reporting survey, original research, design and development, and applications of enabling technologies for collaboration are sought in the following areas:

virtual team support environments
mediators to support collaborative activities
information sharing in distributed systems
enterprise modeling
process modeling and characterization
integration of heterogeneity and legacy databases
applications of collaborative technologies (healthcare, software engineering, electronic commerce, etc.)
projects and team coordination
requirements, constraints
mobile computing
workflow tracking and management tools
networked collocation
tools for multi-media conferencing
capture of design intent and intelligent retrieval of corporate knowledge
tools for design and system integration frameworks

Papers should be no more than 25 typewritten, double-spaced, single-sided pages including all text, figures, and references. Papers should not have been published or be under submission currently for publication elsewhere. Manuscripts should have a title page that includes the title of the paper and the full name, affiliation, postal address, electronic address, telephone number, and fax number of each author. Authors also are encouraged to write a 300-word abstract and a list of keywords that identify the central issues of the paper. Papers can be submitted electronically as PostScript files (preferred mode) or by paper mail. Accepted papers will be included in the post-proceedings, to be published by IEEE Computer Society Press.

<table>
<thead>
<tr>
<th>Manuscript Due</th>
<th>Notification of Decisions</th>
<th>Final Version Due</th>
</tr>
</thead>
</table>

**General Chair:** K. Joseph Cleetus, CERC, West Virginia University

**Program Chair:** John Callahan, CERC, West Virginia University

**Program and Review Committee:**

Chandraft Bhaj, Purdue U.
Earl Craighill, SRI International
Praun Dewan, U. of North Carolina - Chapel Hill
Mark Fox, U. of Toronto
Sumit Ghosh, Brown U.
Ted Gremann, AT&T
Michael Hubs, MCC
Raman Kannan, Monmouth College
Vinay Kumar, Enterprise Integration Technologies
Jintae Lee, U. of Hawaii
John Lewis, OE Corporate R&D
Felix Londono, U. of East Pratam, Colombia
James Portillo, U. of Maryland - College Park

**Local Arrangements:**

Alex Schill, Technical U. of Dresden, Germany
Dennis Sing, Nanyang Technological U., Singapore
Ram Dasu, IISc, NIST
P. David Stott, U. of North Carolina - Chapel Hill
Glebe Trappe, West Virginia U.
Robert Winner, Ctr. for High Performance Computing

**Point of Contact:**

John Callahan  
CERC, P.O. Box 6506  
West Virginia University  
Morgantown, WV 26506  
Phone: 304-293-7225 ext. 2268  
Fax: 304-293-7541  
Email: e-wkhp@cerce.wvu.edu