ELSEVIER

Computer Networks and ISDN Systems 27 (1994) 273–280

COMPUTER NETWORKS and ISDN SYSTEMS

Interactive information services using World-Wide Web hypertext

Steve Putz [1]

Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304, USA

Abstract

Most World-Wide Web information servers provide simple browsing access to collections of static text or hypertext files. This paper describes some interactive World-Wide Web servers that produce information displays and documents dynamically rather than just providing access to static files. The PARC Map Viewer uses a geographic database to create and display maps of any part of the world on demand. The Digital Tradition folk music server provides access to a large database of song lyrics and melodies. These applications take advantage of the multimedia capabilities of World-Wide Web to deliver graphical and audio content as well as formatted text. Hypertext links are used not only for navigation, but also for setting search and presentation parameters. In these applications the HTML format and the HTTP protocol are used like a user interface tool kit to provide not only document retrieval but a complete custom user interface specialized for the application.

1. Introduction

World-Wide Web (WWW) is a wide-area hypermedia information retrieval project started in 1989 by Tim Berners-Lee at the CERN European Laboratory for Particle Physics [1]. WWW merges the techniques of information retrieval and hypertext to make an easy to use and powerful global information system. The WWW project is now being carried on cooperatively by several groups and individuals at sites throughout the world. The use of World-Wide Web browsers and servers for a wide variety of applications has increased dramatically over the last year [7].

WWW Hypertext is represented using Hypertext Markup Language (HTML), which is a simple SGML document type allowing structured text with links [2,13]. In WWW, the activation of a link causes the retrieval of a new hypertext document as specified by a Uniform Resource Locator (URL). Each URL specifies the protocol, server, and a document identifier used to retrieve a particular document [3].

A URL may specify any of a number of document retrieval protocols. The Hypertext Transfer Protocol (HTTP) was designed specifically for communication between WWW clients and servers [4]. HTTP has the advantages of being fast, stateless, extensible, and easy to implement for both clients and servers.

HTML and HTTP work together to provide easy access to documents and information in a wide area distributed network such as the Internet. In a typical example, a client program communicates with a remote server program using the HTTP protocol, requesting some information using a URL, see Fig. 1. The server responds with the requested information encoded as an

[1] E-mail: putz@parc.xerox.com; ⟨http://pubweb.parc.xerox.com/hypertext/people/StevePutz.html⟩.

Elsevier Science B.V.
SSDI 0169-7552(94)00075-1

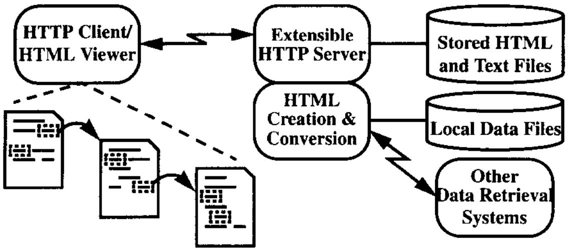


Fig. 1.

HTML document. The client displays the document to the user. Portions of the HTML document may be linked (via a URL) to other documents on the same or different server. The user can follow a link and retrieve the referenced document with a simple mouse click or key press.

Most HTTP servers operate as a hypertext file server, providing access to stored HTML and text documents. Commonly used HTTP server software, such as the CERN httpd, NCSA httpd, and Plexus all provide this capability [5]. However, the HTML hypertext format and the HTTP retrieval protocol can provide more than simple document browsing and retrieval. With additional server software, the same protocol and generic browsers can allow access to many kinds of interactive information sources while providing customized user interfaces.

The document returned by an HTTP server need not be stored in a file. Instead of corresponding to a file name, a URL can cause the server to dynamically generate a new document based on information in the URL. In order to do this, the server must define a naming scheme to define what documents get created for different URLs [6].

HTTP server implementors are now adopting a Common Gateway Interface (CGI) standard so that custom scripts can easily be invoked by a generic HTTP server program for appropriate URLs. The CGI allows a generic server program to process HTTP/1.0 headers, logging of requests, and access to static documents, while custom CGI scripts provide an interface to site-specific services [8].

Two interactive HTTP services are described in this paper. The PARC Map Viewer uses a geographic database to create and display maps of any part of the world on demand. The Digital Tradition folk music server provides access to a large database of song lyrics and melodies. Both of these services are implemented using the *perl* scripting language [15] running under the Plexus HTTP server on a Sun Unix workstation.

2. An Interactive Map Viewer

The Interactive Map Viewer [11] is a World-Wide Web application developed at the Xerox Palo Alto Research Center. This application combines the ability of HTML documents to include graphical images with the ability of HTTP servers to create new documents in response to user input. The HTML format and the HTTP protocol are used to provide a custom user interface for browsing and viewing geographic map data.

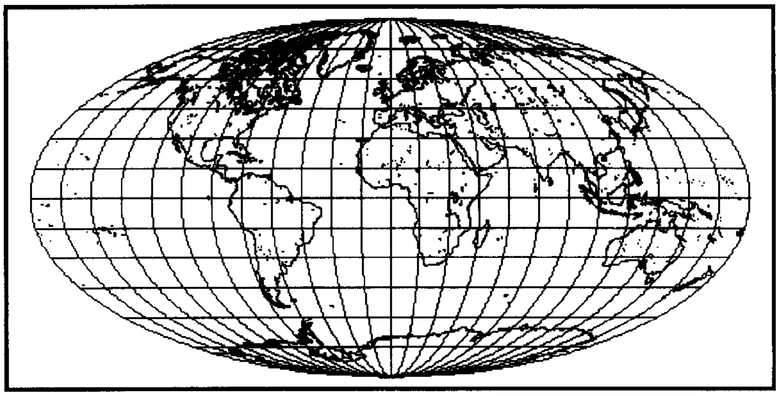
In this application, a computer generated map of the world is embedded into an HTML document using the ⟨IMG⟩ tag for in-line images. Additional links in the document are used as controls allowing the user to change various map rendering options (e.g. pan, zoom, level of detail). Users can interact with the Map Viewer using any WWW browser, that supports in-line images, such as NCSA Mosaic [9]. The Map Viewer server will also provide GIF format map images for use separately or inclusion in other documents.

2.1. The Map Viewer interface

When used as an interactive map browser, the Map Viewer presents the user with a hypertext document consisting of a large graphic image containing the current map. Selecting on the map image causes the map view to zoom in centered on the selected coordinate. Below the image is text which indicates the current and available viewing options. The default world map view is shown in the figure in Example 1.

Example 1 – The default world Map View [2]

Map Viewer: world 0.00N 0.00E (1.0X)



Select a point on the map to zoom in (by 2), or select an option below. Please read About the Map Viewer [3], FAQ [4] and Details [5].

Options:

- Zoom In: *(2)*, *(5)*, *(10)*, *(25)*; Zoom Out: *(1 / 2)*, *(1 / 5)*, *(1 / 10)*, *(1 / 25)*
- Features: *Default*, *All*; *+borders*, *+rivers*
- Display: *color*; Projection: *elliptical*, *rectangular*, *sinusoidal*; *Narrow*, *Square*
- Change Database to USA only (*more detail*)
- *Hide Map Image*, *No Zoom on Select*, *Reset All Options*

Options can also be typed in as search keywords (e.g. "lon = −100", see *details*). Current region is 360.00 deg. wide by 180.00 deg. (12420.00 miles) high.

Preset Coordinates:

- *Globe*, *USA*, *Alaska*, *Hawaii*, *San Francisco Bay*, *United Kingdom*

Since the Map Viewer was created before the forms extension to HTML was available, regular hypertext links are used for changing map rendering options.

2.2. Map Viewer service implementation

The Map Viewer is implemented as an HTTP server running on a Sun Unix workstation using a custom server module written in the *perl* scripting language [15] as part of a Plexus HTTP server. The perl program generates HTML documents and map images on demand, based on parame-

[2] All links (italics) in this example connect directly to the Map Viewer at Xerox PARC: ⟨http://pubweb.parc.xerox.com/map⟩.

[3] ⟨http://pubweb.parc.xerox.com/hypertext/docs/mapviewer.html⟩.

[4] ⟨http://pubweb.parc.xerox.com/hypertext/docs/mapviewer-faq. html⟩

[5] ⟨http://pubweb.parc.xerox.com/hypertext/docs/mapviewer-deta ils.html⟩.

ters encoded into the requested URL. The server responds to two different kinds of map requests. When a GIF format image is requested, the server generates and returns the requested image without any accompanying hypertext document. Otherwise an HTML document is created which includes an IMG element with a URL specifying the appropriate map image. Note that the actual map image is not generated until (and unless) the client browser requests it.

Links in each HTML document (with labels such as "Zoom In") have URLs corresponding to different map rendering parameters, allowing the user to modify the map image by selecting the links. By including the HTML ISMAP attribute in the included IMG element, the server receives image coordinates whenever the user selects a point on the map. The server uses the coordinates to zoom in on the selected location.

The GIF format map images are produced by a separate utility program which is run on the Unix server as a subprocess of the perl script. The rendering options are parsed from the requested URL and converted into command line options for the map generating program, which is invoked using the perl system() function. The output of the map generating program is normally fed directly over the network to the HTTP client without ever being stored in a file. The server does have an image file cache of the most frequently requested images. This cache is checked before invoking the map generating program.

2.3. Map Viewer URL format

There are about two dozen options which can be specified as part of a Map Viewer URL to control various attributes of the generated map image. Since the map server both generates and interprets the map URLs, it is free to choose its own method for encoding the options. Map rendering options are encoded into the URL by separating each option with a slash (/) and within each option separating the option name from its value by an equals (=). Thus the URLs used by the Map Viewer take the form:

http://*hostname*/**map**/*option* = *value*/.../*option* = *value*

where the options specify rendering attributes such as latitude, longitude, map extent, and which geographic features to display. Although the slash character is often used in HTML URLs to define a hierarchical name space (allowing relative path URLs), the Map Viewer's URL name space is not hierarchical and it uses the slash as a simple separator. The order of options is not significant (URLs generated by the map server have the options in alphabetical order) and all options have a default value. An example URL for displaying the Hawaiian Islands is:

```
http://pubweb.parc.xerox.com/
map/lat=21.34/lon= -157.97/wd
=9.0/ht=4.5
```

A list of of the Map Viewer Rendering Options is available in the on-line version of this paper and also in the Map Viewer's on-line documentation [11].

2.4. Incorporating map images in other documents

In addition to providing an interactive user interface for browsing world maps, the map server can also provide separate map images on demand. GIF format map images can be incorporated by reference from arbitrary HTML documents or retrieved from the server and used for other purposes. For example, including the following in an HTML document will insert a small color image with a map of the entire world:

```
<IMG SRC="http://pubweb.parc.
xerox.com/map/color/ iwd=200/
iht=100/format=.gif">
```

It is also possible to link the image back to the interactive Map Viewer interface so that clicking on the image will allow the user to zoom and customize the view.

```
<A HREF="http://pubweb.parc.
xerox.com/map/ border/lat
=41.9/lon=14.8/wd=16/ht=11">
<IMG SRC="http://pubweb.parc.
xerox.com/map/iwd=128/iht=100/
nogrid/ border/lat=41.9/lon=
14.8/wd=16/ht=11/format
=.gif"></A>
```

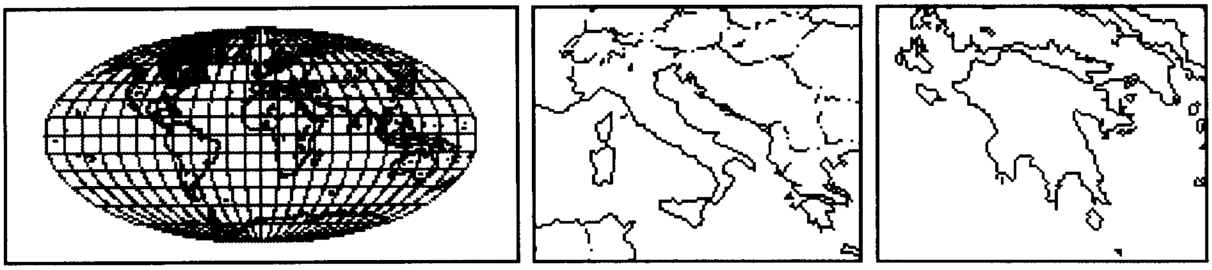


Fig. 3.

2.4.1. Examples of included map images

In Fig. 3 examples are shown of included map images. A particularly innovative use of the map service is the *U.S. Gazeteer* WWW service created by Brandon Plewe [10]. It integrates an existing Geographic Name Server with the PARC Map Viewer. A user simply enters a search query (e.g. the name of a city, county, lake, state or zip code) and a list of matching places is returned as a formatted HTML document. Selecting from the list generates another HTML document consisting of two maps (small and large scale) with the location highlighted (using the Map Viewer's *mark* option). The server in New York does not generate or retrieve the map images, since they are references directly to the HTTP server at Xerox PARC. The user's WWW browser retrieves the map images from the server in California and displays the complete document to the user.

2.5. Map service usage

A significant feature of this application is the ease with which the service was created and its broad accessibility as a network service to users throughout the world. As of February 1994, the Map Viewer server at Xerox PARC was receiving over 25,000 map image requests per week (see Fig. 4).

3. A Folk Music Database

The Digital Tradition Folk Song Server is a World-Wide Web application which provides an interactive interface to a database of 4000 songs. It provides full text search of lyrics, category keyword search, and audio retrieval of melodies [12]. The original Digital Tradition database is distributed as a single user standalone application for the IBM-PC. For the WWW version, the song lyrics and melodies were converted to a simple ASCII format, and a Plexus HTTP server module was created using the *perl* scripting language [15].

3.1. Data representation

Each song has a title, lyrics, and usually one or more category keywords such as "ballad" or

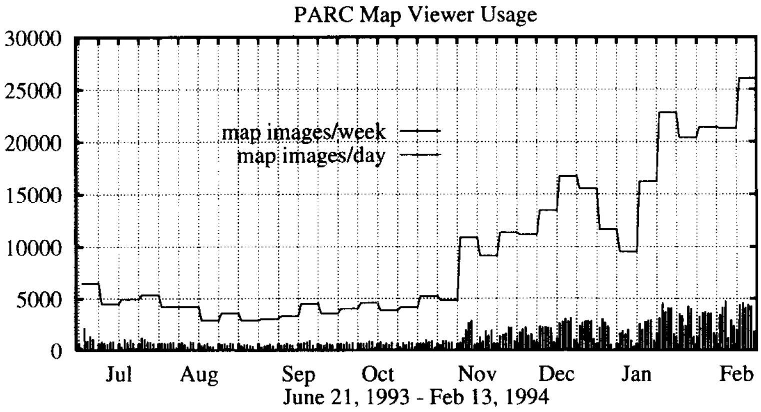


Fig. 4.

Example 2 – Digital Tradition search results [6]

Folk Songs matching: “program*”

Options: _search titles_ or search full text; show matching text or _list titles_ only; list first 50 or _list more_ (100); default settings; or you may enter a _Search Pattern_, or _Start Over_.

11 items matched.

1. _APPLE COMPUTER_ *
   - I wrote my love a **program** that had no end
   - How can there be a **program** that has no end?
   - A **program** with GOTOs, it has no end
2. _I’VE BUILT A BETTER MODEL THAN THE ONE AT DATA GENERAL_
   - Though all my better **programs** that self-reference recursively
3. _GENERATIONS OF CHANGE_
   - from Ed Miller singing at FSGW **program**. learned 1985
4. _JIM SWAGGART, THE PREACHER_ *
   - And they’ve taken my **program** from me.
5. _The Limerick Programmer_ *
   - The Limerick **Programmer**
   - I am a bold **Programmer**, Diamond’s my name,
   - I’ll write you a **program** for cash or for fame
6. _I’M A MAINFRAME_
   - You been runnin’ your **program** on your little PC,
7. _MONTAGNARD SERGEANT_ *
   - Oh, don’t let the **program** go down,
8. _The Programmer’s Lament_ *
   - The **Programmer**’s Lament
   - I met a young **programmer** rolled up in white listings,
   - But now that my **program**’s gone into production
9. _Essay in Creative Specification._
   - And Darkness was upon the face of the **programmers**.
10. _SPERRY RAND_
    - I met a **programmer** a-walking one day
    - I’m not a **programmer**, the truth I will tell
    - I just think about **programs**, the young man did say
11. _WHITE COLLAR HOLLER_
    - And it’s Ho, boys, can’t you code it, and **program** it right
    - **Program**, printout, regress to the mean

"sailor". Most of the songs also have at least one melody stored in the database, represented in a simple ASCII music notation. The lyrics and melodies are stored in two large files, each with a companion table of contents file. Since the total amount of text is less than 6 million characters, a typical full text search can be performed in about four seconds, assuming the text is stored on a local hard disk. Title and keyword searches can be performed in a fraction of a second, since the

[6] All links (italic) in this example connect directly to the Digital Tradition Server at Xerox PARC: ⟨http://pubweb.parc.xerox.com/digitrad⟩.

server keeps the list of titles and keywords in memory. While this kind of linear scan would be too slow for much larger amounts of data, there are many applications for which it is well suited. A more sophisticated inverted word index would be more appropriate for a much larger database.

3.2. Digital Tradition user interface

The Digital Tradition server provides different kinds of information displays depending on what information is being presented. Each display is presented as a formatted HTML document with hypertext links. The two main display formats are a search query and results page and a song lyrics display. There are also lists of category keywords, song titles and song tunes which allow browsing the database without performing a search. In Example 2, a display of Digital Tradition search results is shown.

3.3. Full text searches

This application provides a word-based text search capability. Although the underlying implementation uses regular expressions to find occurrences of search terms, a much simpler query syntax is provided. While regular expressions can be very powerful for specifying search patterns, most end users are unprepared to deal with the confusing and arcane syntax of regular expressions. And the vast majority of users' searches are for simple word or phrase combinations anyway.

The server accepts queries using the standard HTTP convention of ending a URL with a question mark ("?") character followed by the query terms. Adjacent words in a query are interpreted as a phrase (i.e. the words must occur together in the specified order for a match to occur). When the special character " & " is used between query terms (indicating an AND operation), the terms may occur separated and in any order. When the special character " |" is used between terms (indicating an OR operation), only *one* of the terms need occur for a match. Normally each word in a query must exactly match an entire word in the text (ignoring punctuation and case). However the special character "*" will match any sequence of letters of digits. This is most useful for matching variations of a word, such as singular or plural endings.

3.4. Category keyword and title searches

Most songs in the database are categorized with one or more keywords used to group songs by topic or genre. All lyrics and title searches also check each song's list of keywords. It is also possible to search explicitly for keywords by using an "@" character as a prefix (e.g. "@sailor"). This is useful for keywords (such as "father") which occur frequently in lyrics on other topics. This categorization scheme and the use of "@" to mark category keywords was inherited from the original PC-based version of the database, and it works well with the regular expression searches used by the WWW server.

As an alternative to the full text search capability, searches may optionally be performed on just the song titles and keywords. The same query syntax is used as for full text searches. The user can set a display option to specify whether each song's category keywords are listed under the titles.

3.5. Search terms displayed in context

Using HTML for presentation of search query results provides a great deal of flexibility. Using the presentation markup capabilities of HTML, the Digital Tradition server shows lines of text that match a search query below each song title with the search terms highlighted. When the full text is displayed, the search terms are highlighted there as well.

3.6. Audio output

Currently, the only widely supported format for delivering music via World-Wide Web is as telephone quality sampled digital audio (8-bit u-law). When a client requests a melody in audio format (by selecting a link), the server converts the stored note list into audio samples using a public domain music software package called Csound [14].

This is a grossly inefficient method of sending simple tunes across a wide area network. A possible improvement is to provide the melodies to clients in Standard MIDI Format, as software MIDI emulators are now becoming available.

3.7. Using hypertext for documentation and examples

Authored hypertext documents can be combined with the interactive service in useful ways.

Links from the interface point to a variety of pages which use HTML hypertext for documentation about the application. Some of the documentation pages in turn include examples with links that perform sample database queries.

4. Conclusions

A fundamental and important aspect of the design of World-Wide Web is that information references (currently implemented as URLs) can just as easily refer to dynamically generated information as static files. Although most World-Wide Web servers today perform primarily as hypertext file servers, there is an increasing trend towards more dynamic information services, where custom documents are assembled and delivered to a user on request. The presentation markup and hypertext capabilities of HTML make it possible to present dynamically generated information in a way which is both functional and aesthetically pleasing. Authored hypertext documents can be combined with HTTP based interactive services to provide documentation that is well integrated with the application's user interface. Links to appropriate documentation can appear in dynamically generated pages, and static hypertext documents can include sample links which invoke queries or other services.

Acknowledgements

The map rendering software used by the PARC Map Viewer was derived from many sources, dating back to a world map browser created by Al Paeth for the Smalltalk-76 system in 1983, using data from the CIA World Data Bank map data. The author later ported and improved versions of this browser to the Smalltalk-80, Unix, SunView, and X Windows systems in 1989 to 1992. Brian Reid provided new map data and useful data conversion software in 1991. More detailed data for the United States was obtained from a U.S. Geological Survey CD-ROM in 1992.

Thanks to Dick Greenhouse for creating *The Digital Tradition* folk music database, to Dennis Cook, Susan Friedman and many others for contributing songs, and to the Philadelphia Folk Song Society.

Larry Masinter has helped me to explore applications of the Web ever since he first introduced me to Tim Berners-Lee and World-Wide Web in 1992.

Above all, thanks to Tim Berners-Lee for conceiving and implementing the original World-Wide Web system, and spending a month exchanging ideas with us at the Xerox Palo Alto Research Center during the summer of 1992.

References

[1] T. Berners-Lee, *The World Wide Web Initiative*, CERN WWW Documentation, updated 1994. ⟨http://info.cern.ch/hypertext/WWW/TheProject.html⟩.

[2] T. Berners-Lee, *Hypertext Markup Language (HTML)*, Version 1.2 IEFT Internet Draft (work in progress), 1994. ⟨http://info.cern.ch/hypertext/WWW/MarkUp/HTML.html⟩.

[3] T. Berners-Lee, *Hypertext Transfer Protocol (HTTP)*, CERN, IEFT Internet Draft (work in progress), original version December 1991, updated 1994. ⟨http://info.cern.ch/hypertext/WWW/Protocols/HTTP/HTTP2.html⟩.

[4] T. Berners-Lee, *Uniform Resource Locators*,IEFT Internet Draft (work in progress), 21 March 1994. ⟨http://info.cern.ch/hypertext/WWW/Addressing/URL/Overview.html⟩.

[5] T. Berners-Lee, *W3 Server Software*, CERN WWW Documentation, updated 1994. ⟨http://info.cern.ch/hypertext/WWW/Daemon/Overview.html⟩.

[6] T. Berners-Lee, *Making a server*, CERN WWW Documentation, 1992. ⟨http://info.cern.ch/hypertext/WWW/Provider/ServerWriter.html⟩.

[7] K. Hughes, *Entering the World-Wide Web: A Guide to Cyberspace*, Enterprise Integration Technologies, March 1994. ⟨ftp://ftp.eit.com/pub/web.guide⟩.

[8] R. McCool, *CGI specification*, NCSA WWW Documentation, 1994. ⟨http://hoohoo.ncsa.uiuc.edu/cgi/⟩.

[9] National Center for Supercomputing Applications Software Development Group, *NCSA Mosaic Home Page*, NCSA WWW Documentation, updated 1994. ⟨http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html⟩.

[10] B. Plewe, *U.S. Gazeteer*, HTTP Service at the State University of New York at Buffalo, 1994. ⟨http://wings.buffalo.edu/geogw⟩.

[11] S. Putz, *World Map Viewer*, HTTP Service at Xerox PARC, 1993. ⟨http://pubweb.parc.xerox.com/map⟩.

[12] S. Putz, *Digital Tradition Folk Song Database*, HTTP Service at Xerox PARC, 1993. ⟨http://pubweb.parc.xerox.com/digitrad⟩.

[13] D. Raggett, *HTML+ Discussion Document*, CERN WWW Documentation, November 1993. ⟨http://info.cern.ch/hypertext/WWW/MarkUp/HTMLPlus/htmlplus_1.html⟩.

[14] B. Vercoe, Csound – digital audio processing and sound synthesis, Music and Cognition group of the Media Laboratory at MIT, 1991–1993. ⟨ftp://ems.media.mit.edu/pub/Csound/⟩.

[15] L. Wall and R.L. Schwartz, *Programming Perl* (O'Reilly & Associates, 1990).