World-Wide Web:
An Information Infrastructure for High-Energy Physics

T.J. Berners-Lee, R. Cailliau, J.-F. Groff, B. Pollermann
C.E.R.N., 1211 Geneva 23, Switzerland

ABSTRACT

The World-Wide Web (W^3) initiative encourages physicists to share information using wide-area networks. The W^3 software provides easy hypertext navigation and information retrieval in a consistent manner to a vast store of existing data and future hypertext. The client-server architecture uses global conventions for document identifiers, a set of common access protocols, and an ever-widening set of transfer formats. The HTTP protocol is introduced which allows servers, sometimes simple shell scripts, to provide data and take advantage of a range of hypertext browsers on many platforms. Existing data may be put on the “web” by a gateway without affecting data management procedures. Internet archives, news, “WAIS” and “Gopher” systems are already included in the web. The future will see multiple data formats being handled by negotiation between client and server, and hypertext editors bringing collaborative authorship in the information universe.

The need

In few disciplines is the need for wide-area hypertext so apparent and at the same time so soluble as in particle physics. The need arises from the geographical dispersion of large collaborations, and the fast turnover of fellows, students, and visiting scientists who must get “up to speed” on projects and leave a lasting contribution before leaving. Fortunately, the community necessarily has a good computing and network infrastructure.

Much information is in fact available on-line, but references to it involve complicated instructions regarding host names, logon passwords, terminal types and commands to type, sometimes needing the skilled interpretation of a network “guru”. W^3 replaces this with a point-and click interface which anyone can use.

The technology

The W^3 reader uses two operations to access the world of information. The first is to jump across a hypertext reference. On a graphic terminal, a reference is represented by a sequence of highlighted text, or an icon. The user clicks on it with the mouse, and the referenced document (or part of a document) appears. On a line mode terminal, a reference is represented for example by a number in the text: the user types the number to follow the reference.

Hypertext links allow many existing structural forms to be represented: references

Page 1 of preprint of T.J. Berners-Lee, R. Cailliau, J-F Groff, B. Pollermann, CERN, "World-Wide Web: An Information Infrastructure for High-Energy Phsyics", Presented at "Software Engineering, Articicial Inteligence and Expert Systems for High Energy and Nuclear Physics" at La Londe-les-Maures, France, January 1992. Proceedings to be published by World Scientific, Singapore, ed. D Perret-Gallix.

between academic papers, pointers from a table of contents to the text, “See also” references and such like within books. Systems of nested menus which characterise many online help systems can also be represented: each menu page is a hypertext document pointing to a set of other menu pages or documents.

An index may also be regarded as a hypertext document, but searching an index by hand is time consuming. Futhermore, text retrieval software uses sophisticated search techniques which are out of the question for a human being. Therefore, the second operation W^3 allows is for the reader to present a query to a remote search engine. The engine itself, and the particular search it will perform, are represented by a virtual “cover page” document. This index document may itself be found by following hypertext links or indeed by a search.

The architecture (Fig. 1)

It can be seen that whereas following a hypertext link requires intelligence on the part of the application used to present the data, a search must be performed at the site where the data resides. The architecture therefore involves two programs, a client “browser” and a server, communicating across a network. The “information bus” which connects clients consists of a set of standards and conventions. At this level, a conceptual world is composed of named documents which may or may not be searchable indexes. The browser is free to present the documents and the relationships betwen them in any appropriate way to the user. The server is free to generate the documents either by sending real files, or by generating virtual hypertext on the fly in reponse to a request. This is how a “gateway” server can provide a hypertext image of all the data in another system.

The three conventions of the information bus are Universal Document Identifiers, a set of protocols, and a set of data formats.

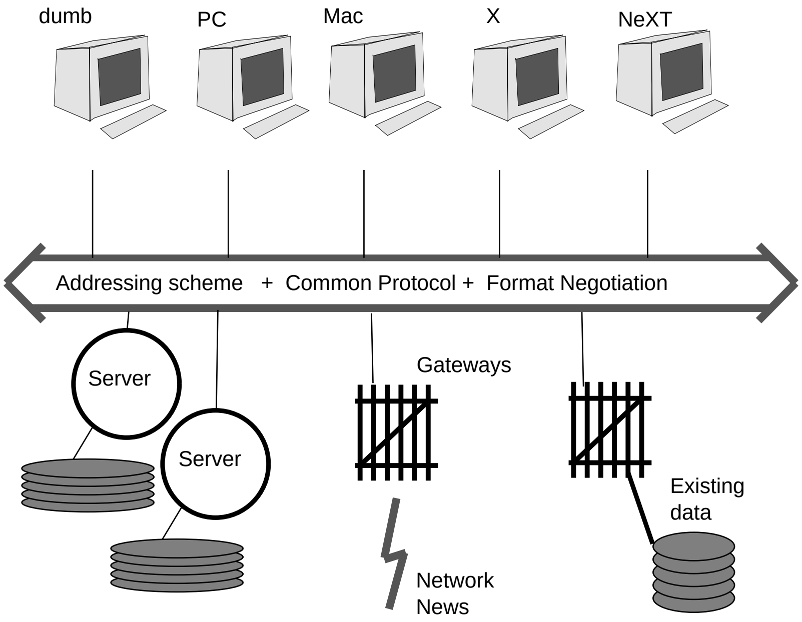

Fig. 1. The W^3 architecture in outline.

Protocols

A number of existing protocols, and one new protocol, form the set with which a W^3 browser is equipped. Standard protocols used are

- The FTP protocol [Post85] allows access to the internet archives of software and other information. Directories are browsed as hypertext. The browser will notice references to files which are in fact accessible as locally mounted (or on DECnet™ on VMS™ systems) and use direct access instead.

- The NNTP protocol [Kant86] allows access to news groups and news articles. News articles make good examples of hypertext, as articles contain references to other articles and news groups. News groups are like directories, but more informative.

Some nonstandard protocols are in use by other experimental systems, and W^3 browsers can access two of these:

- The Wide Area Informtion Server (WAIS) protocol [Kahl90] provides search and retrieve by database. Although not currently specified to allow links to other servers or data accessible by other protocols, it is close to the functionality required by W^3. A gateway server provides access from any W^3 browser to any WAIS server.

- The “Internet Gopher” protocol [Albe91] provides a distributed information service consisting of interlinked menus and files. As the menus map onto hypertext and the Gopher protocol is simple, W^3 browsers can browse the gopher world as part of the web.

A new protocol, HTTP [Bern91], was defined in order to allow, like WAIS, document retrieval and index search. The “HyperText Transfer Protocol” is a protocol for retrieving anything (not necessarily hypertext) as fast as is needed in response to a hypertext jump. HTTP is a very simple internet protocol, similar in implementation to FTP and NNTP (in that respect it is like Gopher). The HTTP client sends a document identifier with or without search words, and the server responds with hypertext or plain text. The protocol runs over TCP, using one connection per document request. The browser acts as a pipeline, so that as the bytes arrive from the server they can be presented to the reader as soon as they arrive.

Universal Document Identifiers

The power of the system lies not in a complex protocol, but in the Universal Document Identifier (UDI) [Bern92]. The UDI, while compact and printable, contains fields allowing a server to be identified, a document to be identified on that server, and a search to be performed if necessary. It may have a field specifying a particular part of a document to be selected when the document is presented. Some examples are

`http://info.cern.ch/hypertext/WWW/TheProject.html`	A document about the WWW project ;
`http://crnvmc.cern.ch/FIND?sgml`	A list of documents about SGML in the CERN computer centre index. ;
`news:comp.sys.next.announce`	An internet newsgroup ;
`news:ddrg1423f@cernvax.cern.ch`	An (imaginary) news article ;
`file://info.cern.ch/pub`	An anonymous FTP directory ;
`Overview.html`	A file in the same directory as the current document ;
`Overview.html#proof`	A part named “proof” of that document ;
`#proof`	A part named “proof” of the current document.

Data formats

As formats for representing data will continually evolve, one cannot restrict them in any way. The W^3 architecture proposes a negotiation between client and server to agree on a document format for transmission. This has not to date been implemented, but its importance will become increasingly apparent. Current W^3 browsers handle either plain text or simple hypertext, and in fact the great majority of information currently available in the world is representable in this way, as is evidenced by the growth of the web during the last year.

Plugging data into the web

In these early stages of the web, the most practical use of effort is to make small servers which provide existing data as part of the web. Now there are many examples of such servers, including in particle physics,

- “Phone book” indexes of people’s coordinates (CERN and NIKHEF) ;
- Documentation indexes for CERN, DESY and NIKHEF documentation ;
- Preprints: the “SPIRES” preprint database (SLAC) ;
- News and annoucements (CERN and NIKHEF) ;
- Online help systems (CERN) ;
- The W^3 documentation in hypertext.

It is worth mentioning a few gatewayed servers. The “Archie” server in its WAIS version installed at Oregon State University, is accessible through the WAIS-W^3 gateway. The index contains records for almost every file available by anonymous FTP on the Internet. The W^3 browser can access the index and jump straight to the chosen file.

At the University of Graz in Austria, an existing “Hyper-G” hypertext system is gatewayed onto the web, the server performing on-the-fly conversions of hypertext formats.

A daemon is available to allow VMS/Help™ libraries to be served on the web, and such a server is running at the CERN computer centre. This program is available to anyone who has historically used VMS/Help as a format for online documentation.

```
#!/bin/sh
read get docid
echo "<TITLE>$docid</TITLE>"
echo Here is the data.
```

Fig. 2. A trivial HTTP server script. A more powerful server daemon written in C is distributed with W^3, allowing document name to filename mapping and screening rules.

The simplicity of the HTTP protocol makes it easy to set up a new server: a shell script will do in simple cases (Fig. 2). On the VM/CMS™ systems, a basic daemon program written in C invokes a REXX exec file for flexibility and easy access to system functions.

The server currently has the possibility of returning text in hypertext mark-up language (HTML) or in plain text. Plain text is sent as a hypertext file beginning with a special <PLAINTEXT> tag which introduces unlimited ASCII text.

Hypertext mark-up

The HTML format used to represent hypertext (HyperText Markup Language) is a simple tagging scheme based on SGML. It has a few simple formatting options (Fig. 3) which allow it to be effectively used for on-line documentation as well as menus and search results. Tagging with a high-level representation of the data allows different browsers to display the text optimally according to the functions available on particular systems.

The tag which makes the text hypertext is the “anchor” tag. An anchor is a section of text which may be the start- or end-point of a link. The start of an anchor is marked with an <A> tag (with attributes), and the end by a </A> tag.

To be the end-point of a link, the anchor must have a name. This name is used in the UDI for the anchor, after a hash sign.

```
        The <A NAME=PROOF>proof of the pudding</A> is in the eating.
```

In this example, the anchor is the text “proof of the pudding” and has the name PROOF. To be the start-point of a link, the anchor tag must have a reference. This is specified with the HREF attribute in the <A> tag.

`<TITLE>Alice in Wonderland</TITLE>`	Document title
`<H1>Main heading</H1>`	Top level heading
`<H2>Section heading</H2>`	Next level heading (up to 6 levels)
`<UL>..</UL>`	List of items
`<MENU>...</MENU>`	Menu of (smaller) items
`<DIR>..</DIR>`	Directory list of (even smaller) items
`<LI>`	Start item within UL, MENU or DIR
`<XMP> ... </XMP>`	80-column litteral text
`<LISTING> ... </LISTING>`	132-column (at least) litteral text
`<PLAINTEXT>`	Litteral text follows to end of file

Fig. 3. Some formatting tags of the HyperText Mark-up Language.

```
        The <A HREF=#PROOF>proof</A> has always eluded them.
```

In this second example, the anchor is the word “proof”. It has no name, but is linked to anchor PROOF in the same document. The value of the HREF attribute is a UDI. In this case the UDI has no document identifier, only the anchor following the hash sign ; the same document is assumed. When displayed, the anchor would be highlighted in some way: “The proof has always eluded them”. Clicking the mouse on “proof” would cause a jump to “the proof of the pudding”.

These examples are given to show how easy it is to generate a hypertext document from a database or any other data source.

Writing hypertext: Forming the web

All the facilities described so far allow very many readers to read data from a few “information providers”. The real power of hypertext becomes evident when readers have the ability to create links. We shall not here discuss the many advantages of writing hypertext over ordinary writing, except to mention that

- One never needs to write something twice, or copy it, as a reference will do just as well;
- Because data is not copied, it is less likely to be out of date;
- One can represent one’s own view of the world, and connect it to other people’s data.

The first immediate boon of editable hypertext is that one customises one’s “home” page to make it a personal hyper-notebook, with links to all the things one finds useful. As a

UDI may represent a document, an index, or the result of querying an index, links may be made to such things as “all the articles in the computer center about printers”. Each time the link is followed anew (subject to a certain amount of caching done by the browsers) the query will be reevaluated to catch any new articles.

One can distinguish modes in which the web is used by readers, from quick reference (phone numbers, weather check) through reading technical manuals to resource discovery (searching for a program to coordinate a car pool). After an exhaustive resource dicovery phase leading to some remote gem of information, the exhausted user can now link that gem to his home page, and it becomes a rapid reference item.

When a group of people all have hypertext authoring tools, they collaborate on a web which represents their joint shared knowledge. This computer-supported collaborative work (CSCW) phase will be a great boon to the HEP community.

Current status

A prototype WYSIWYG hypertext editor on the NeXT platform was developed to demonstrate the feasability of the project. Its development was made easy by the NeXTStep™ development environment (it took about a month). The editor is used for all the W^3 internal documentation, and the maintenance of a web of pointers to useful information. The prototype has not currently been extended to the full range of access protocols of the line mode browser.