GENVL and WWWW: Tools for Taming the Web*†

Oliver A. McBryan
Dept. of Computer Science
University of Colorado
Boulder, CO 80309.

Email: mcbryan@cs.colorado.edu
WWW: http://www.cs.colorado.edu/home/mcbryan/Home.html

Abstract

A fundamental problem with the World Wide Web is the enormous number of resources available and the difficulty of locating and tracking everything. In this paper we will discuss two tools, GENVL and WWWW, designed to deal in different ways with resource location on the WWW.

GENVL is an interactive user-driven hierarchical virtual library system for cataloguing Web resources. The real power of GENVL comes from the built-in recursion which is the key to extendibility and to avoiding the generation of massive linear lists. GENVL has been accessed 112,000 times in 4 months.

WWWW - the WWW Worm - is a resource location tool. It is intended to locate almost all of the WWW-addressable resources on the Internet, and provide a powerful search interface to those resources. Searches can be performed on document titles, reference hypertext, or within the components of the URL name strings of documents - for example to locate all mpeg movies in Finland. WWWW has been accessed 60,000 times in 45 days.

In the paper we discuss the design of GENVL and WWWW, the tools needed to make them work, and difficulties encountered with using underlying WWW facilities.

Keywords: WWW, virtual library, internet, resource location, bulletin board, search service.

---

* Research supported in part by NSF Grand Challenges Applications Group grant ASC-9217394 and by NASA HPCC Group Grant NAG5-2218.
† To appear in Proceedings of the First International World Wide Web Conference, ed. O. Nierstrasz, CERN, Geneva, May 1994.

TABLE OF CONTENTS

GENVL and WWWW: Tools for Taming the Web*†

Oliver A. McBryan
Dept. of Computer Science
University of Colorado
Boulder, CO 80309.

Email: mcbryan@cs.colorado.edu
WWW: http://www.cs.colorado.edu/home/mcbryan/Home.html

1. Introduction

A fundamental problem with the World Wide Web (WWW) is the enormous number of resources available and the difficulty of locating and tracking everything. Manually generated archives are extremely laborious to develop, and are intrinsically non-scalable. In this paper we will describe two tools, GENVL and WWWW, designed to deal with resource location on the WWW.

GENVL (GENerate Virtual Library) is an interactive hierarchical virtual library for cataloguing Web resources by subject area. GENVL has been in operation for some months and outside users have by now added hundreds of interesting entries and created dozens of new sub virtual libraries.

GENVL derives its power from the built-in recursion which is the key to scalability, extendibility and to avoiding the generation of massive linear lists. Operations are also provided for editing and moving entries thereby allowing users and moderators to reorganize data as appropriate.

WWWW - the WWW Worm - is a resource location tool. It is intended to locate all of the WWW-addressable resources on the Internet, and provide a powerful search interface to those resources. The system consists of two parts: one that locates resources, and the other which provides the search interface.

The resource locator is run periodically to create a reference database. The search client provides fast access to the database. Searches can be performed on titles, reference hypertext, or within the components of URL name strings.

Both GENVL and WWWW are experiencing very heavy demand. Usage is typically in the range of 1,500 acceses per day for each program, and it appears from user's comments that both services are quite appreciated.

Section 2 discusses GENVL in detail, while section 3 focuses on WWWW. While these tools are unrelated we have found useful ways to couple them. The GENVL virtual library pages include a pointer to WWWW so that a user who cannot locate a resource in the virtual library can try a keyword search for it. Also a user finding that WWWW is not able to locate a page known to exist (perhaps the user's home page) may add that page to the WWWW knowledge base by posting it to a specific GENVL virtual library.

---

* Research supported in part by NSF Grand Challenges Applications Group grant ASC-9217394 and by NASA HPCC Group Grant NAG5-2218.
† To appear in Proceedings of the First International World Wide Web Conference, ed. O. Nierstrasz, CERN, Geneva, May 1994.

2. GENVL

2.1. Overview

GENVL is an interactive hierarchical system for cataloguing Web resources. A "Virtual Library" (VL below) is a catalogue of Web resources in an area of interest. Each resource in the catalogue appears as a hypertext item in a summary list of that VL. Resources listed can be one of:

o Further Virtual Libraries

o Titles, usually containing URL pointers

o Reports inputted by the user, usually including URL pointers.

This tool was previously called GENBBB (**GEN**eric **B**ulletin **B**oard **B**uilder). The term "Bulletin Board" had been chosen here to provide analogy to conventional bulletin boards. However modern parlance would suggest the term "Virtual Library" instead. We will use the latter term throughout this paper.

Each VL is visible on the WWW as an HTML hypertext document. Commands are provided to add and delete each of the three basic types of resource. In addition newly created VL's allow a user-supplied inlined image and introductory description of purpose, which default to those of its parent. GENVL has been in operation for several months and outside users have by now added hundreds of interesting entries and created dozens of sub bulletin boards. Most of the current GENVL tree structure has evolved completely naturally through user additions. The power of GENVL comes from the built-in recursion which is the key to scalability, extendibility and to avoiding the generation of massive linear lists.

GENVL supports WAIS search on its hierarchy. It also provides a global resources list at every level, recording all resources at and below that level in a time-ordered list, useful for quickly spotting new additions to the system. We will describe GENVL, the tools needed to make it work, and difficulties encountered with using underlying WWW facilities.

GENVL is located at the URL location:

http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/summary.html

which is the summary page of the root GENVL virtual library. This page has been described elsewhere as *"The Mother-of-all Bulletin Boards"*. GENVL requires Forms support for adding new postings, and therefore users with MacMosaic can only read entries, but not post new ones. Mosaic 2.0 and similar clients can both read and post.

GENVL has received 112,003 accesses as of April 22, covering about a 4 month period. The vast majority of these accesses have been to read rather than add materials to GENVL.

2.1.1. Related Work

There are a large number of manually maintained virtual libraries available on the WWW. Some of these are subject-oriented such as [1] while others are service oriented, for example [2]. Many VL are specific to a single subject, for example a list of Computer Science Departments [3]. See [4] for a collection of pointers to virtual libraries.

Most of the well-known WWW VL are maintained and updated by one person. Because information is not inserted directly by information providers these systems do not scale well to an Internet with millions of computers. Furthermore, most of these systems are not hierarchical, which also severely impacts scalability. One approach to user-supplied information is represented by Martijn Koster's ALIWEB [5], which relies on each WWW server to provide information on resources it maintains.

GENVL is oriented towards allowing others to update VL information, and is explicitly highly hierarchical, thereby improving scalability. Some maintenance is still required, for example to correct submission syntax errors and occasionally to relocate a submission to a more appropriate place. GENVL takes inspiration from Eric Ebina's early demonstration, Free for All [6], of Forms capabilities in NCSA Mosaic.

2.2. GENVL Virtual Libraries

GENVL represents a general tree of VL's. A VL contains a set of resource postings, where a posting may be either an atomic entry (called an *entry* below) or a pointer to another VL created by GENVL (called a *sub-VL* below). Every posting is referenced by a posting title which describes the posting. To be interesting, a title should include a reference to a URL, thereby pointing to information somewhere on the WWW. In the case that GENVL creates a sub VL, it automatically associates the sub VL title with a URL that points to the new VL.

A VL is displayed to an HTML viewer such as Mosaic as an alphabetically sorted list of posting titles called the VL Summary Page. Clicking hypertext in any posting title leads to the corresponding information. Titles of sub-VL are preceded by an asterisk and are displayed in bold type to distinguish them from ordinary postings. Each posting title is preceded by an integer sequence number, which is used later in requests to edit, move or delete that posting. This last point is an awkward feature, but the user interface provided by Mosaic provides little by way of point and click capabilities for such operations. One could conceivably use the ISMAP feature, although it would be very inefficient given that pages are continually changing.

Each VL has several associated characteristics which are shown in Table 1. When a new VL is created, GENVL supplies a Form (requires Mosaic 2.0 or an equivalent WWW client) to fill out. The user can supply a pointer to a GIF file to be used as the image for that VL, and a statement of purpose to be placed in the header of the VL, in addition to the VL title. The statement of purpose might summarize the goals for that VL, any ground rules and the name of a moderator. If no image is supplied the VL image defaults to the image of its parent VL, while the statement of purpose defaults to nothing. The creator may also specify that a VL is only to allow creation of sub-VL - i.e. no entries may be posted. This is done by toggling the Create-Only bit. As an example, a VL entitled *"University Departments Worldwide"* might allow only creation of sub-VL with Department titles such as *"Computer Science"*. Each such sub-VL would then allow atomic entries to be posted, such as *"University of Colorado Computer Science"*.

Each VL records the name and email address of the creator and a user-supplied password. Only the title field is required, with all other characteristics being optional. However if a VL is created without a password then anyone can delete it. If a password has been specified, it can be deleted only by a user who specifies that password. When a VL is deleted, all postings to the VL - including both entries and sub-VL - are deleted, even though the creator of the VL may not have the passwords of the posted entries. Therefore it is important that VL be assigned a password to protect from abuse.

Table 1: Characteristics associated with a VL

Title	Used on the Summary Page and in all Reports
Image	Used on the Summary Page and in all Reports
Purpose	A description included in the Summary Page.
Create-Only Bit	If set, allows only addition of sub VL's to this VL
Author	Name of the person adding the entry
Email	Email address of the person adding the entry
Password	Password supplied by the person adding the entry

2.3. GENVL Entries

An entry in a VL is defined as a VL posting that is not a sub-VL. Entries are of two types:

1. Pointers: A title is supplied which contains a URL reference.
2. Reports: A title and a short report are supplied.

In the case of reports, the user inputs the report on the supplied Form and GENVL arranges to store the report and create a link from the title to the report. If the title already contains a URL pointer, then GENVL detects this and adds an additional pointer to the report. Thus such entries will typically have at least two pointers. Furthermore a report will often include further URL's within its text. When adding a report it is possible to specify a GIF image to be included in the report heading in place of the default image which is the VL image of the current VL. Finally, entries are also associated with the users name, email address and a supplied password, with the same implications as in the case of a VL - except that if a password is omitted, only that single entry could be deleted by others.

Thus an entry in a VL has associated characteristics as described in Table 2:

Table 2: Characteristics associated with a VL Entry	
Title	Used to provide a hypertext reference to the entry
Image	A user supplied inlined image used in a report
Type	Specifies whether a Pointer or a Report
Report	Contains the report in the case Type=Report
Author	Name of the person adding the entry
Email	Email address of the person adding an entry
Password	Password supplied by the person adding an entry

2.4. GENVL Operations

Each VL supports a set of modification operations that are accessed through the Add/Change/Delete key at the top of the VL Summary page. It is possible to add a pointer (Point command), add a list of pointers (Pointers command), input a report (Input command), create a sub-VL (Create command) or delete one or more previous postings. In addition one can edit a previously created entry, and copy or move a posting to a different location on the global VL tree. The Pointers command allows lists already generated or available elsewhere to be incorporated into GENVL.

Table 3: VL Operations	
Point	Add an entry by providing a one-line hypertext description
Pointers	Add entries by as a list of one-line hypertext descriptions
Input	Add an entry by inputting text which will be pointed to
Create	Create a sub VL
Delete	Delete one or more VL entries or sub VL
Edit	Change the contents of a VL entry
Copy	Copy a VL entry to a different VL location
Move	Move a VL entry to a different VL location

All of these operations require WWW Forms support and do not currently work as a result on MacMosaic 1.x although they do work with Mosaic 2.0 and with many other clients such as Lynx.

2.5. Other Features

GENVL supports several methods for accessing information other than through the VL interface. GENVL supports a WAIS server, which indexes all information in the global VL tree. Thus keyword searches on topics of interest are easily performed.

Each VL in GENVL also supports a global list of all VL postings within that VL. This is a recursive listing which includes all sub-VL of the VL. The list is presented in time order with most recent additions first, providing a simple way to monitor additions to any topic without having to traverse parts of a tree. In particular, the global list at the top level lists all entries anywhere on the tree.

Each VL summary also provides pointers that move to the parent VL and to the root VL of the tree, providing further useful ways to move around in the tree.

2.6. GENVL Implementation

GENVL uses a file system directory structure to implement the tree structure of the VL hierarchy. Each VL in GENVL is stored in a separate directory, a subdirectory of the VL that created it. Each such directory contains a file *summary.html* which describes all postings in that VL, and which is the main access point from WWW to that VL. The root directory therefore contains the root summary page, which is at address:

http://www.cs.colorado.edu/homes/mcbryan/public_html/bb/summary.html

and in turn provides access to all other VL. Each VL directory also contains a file called *makevl* which will be used later to create sub directories and install starting summaries and other configuration data for any new sub-VL that are created.

Each VL posting is represented by an access file *num.pass* where *num* is the numeric sequence number of the posting in the VL. File *num.pass* records the title, author, author's email and password of the posting. Each posting that includes a report will also have a file called *num.html* containing the user-supplied text of the report. Finally if posting *num* is a sub-VL, then there will be a corresponding subdirectory called *num* which will contain that sub-VL.

The sequence number num is also used to delete postings by simply removing the corresponding *num.pass*, *num.html* files or *num* directory. Following every posting or deletion, the summary file of the VL is rewritten and a copy of it is returned to the WWW client. Each VL uses a locking system to ensure that two or more users cannot post to the same VL simultaneously. Locking is tied to the assignment of sequence numbers. As soon as a new sequence number is assigned the VL is unlocked, allowing others to begin postings even while the first person is still composing a posting.

The global hierarchical history for a VL is stored in a separate file *SUMMARY.html* in each VL directory. Because a change to a VL affects the global summary of all parent VL directories, it is not practical to update all of these for each posting. A separate process scans the VL tree and updates all *SUMMARY.html* files every 10 minutes. The WAIS search service works similarly, with the WAIS indexer run from a separate process.

3. WWWW

3.1. Overview

WWWW - the WWW Worm - is a resource location tool. It is intended to locate all of the WWW-addressable resources on the Internet, and provide a powerful user interface to those resources. The system consists of two parts: one that locates resources, and the other which provides the user interface.

A program, wwww, scours the Internet locating all web resources - HTML files and more general URL's. It builds a database of these, which is currently over 110,000 entries (early March '94). Each HTML file found is indexed with the title string used in there. Each URL referenced in an HTML file is indexed by a) the clickable hypertext associated with the URL b) the name of the HTML file referencing the URL and c) the title of the latter HTML file. Inlined images are a special case as they often appear standalone in text; in that case they are associated with the containing HTML and its title. Inlined images that are contained within clickable hypertext references are indexed also by that reference.

WWWW.html is a WWW accessible page ("client" might be a better description) which provides access to the WWWW database. It provides direct access to the lists of URL's, where each is displayed with its title or reference text. More importantly it provides access to a URL-wise search server which can perform keyword searches for URL's or classes of URL's. Searches can be performed on titles, reference hypertext, or within the URL name strings of documents - for example to locate all MPEG movies in Finland. Sample searches are provided, ready to be submitted with a mouse click.

In this paper we discuss the design of WWWW, procedures used to ensure it is a well-behaved Internet process, and optimization issues. We will describe the search engine (which is still evolving) and why available search engines such as WAIS appeared less than ideal. We also discuss some additions to the HTTP protocols which would allow much more powerful use of tools such as WWWW. WWWW can be found at:

http://www.cs.colorado.edu/home/mcbryan/WWWW.html.

WWWW requires Forms support for running the search engine, and therefore will not currently work on MacMosaic 1.x.

WWWW has received 61,488 accesses as of April 22, 1994 covering a period of 46 days, for an average of about 1,500 accesses per day.

3.1.1. Related Work

There are currently a considerable number of WWW robots and search engines. For pointers to these we refer to M. Kostner [7] and to the CUI Search Engines catalogue [8]. Some examples of other robots are represented by [9-11]. New robots are appearing at an astonishing rate. Internet search services has been an active research area for years and we do not attempt to survey all of that work here. A good reference is [12] which has a comprehensive survey of resource discovery methods. Examples of such services include WHOIS [13], X.500 [14], archie [15], WAIS [16] and Netfind [17-18].

3.2. WWWW Resource Location

The resource location part of WWWW is a program wwww which searches the WWW for URL's and records them in an archive. From the set of all URL's we select a distinguished sub-category: an HTML URL (abbreviated to HTML below) is a URL which represents an HTML file, i.e. has the extension *.html*. The wwww program is run as:

wwww archive levels HTML1 HTML2 .... > new-archive .

Here *archive* denotes an archive generated in a previous run and *levels* is a recursion depth. WWWW reads each of the files represented by the URL's *HTM1*, *HTM2*, ... and applies itself recursively to any HTML references found within these files. The recursion continues to a depth of *levels*.

When wwww opens an HTML, it reads and records the HTML title field required at the head of every such document. The title text is then associated with that HTML. Whenever wwww encounters a URL within an opened HTML file, it reads and records the accompanying hypertext anchor. The hypertext and the containing HTML are then associated with that URL.

On completely processing an HTML document, wwww marks it as read so as to avoid visiting it again from some other HTML reference. This greatly improves efficiency while decreasing the impact wwww might otherwise place on the network.

The program wwww uses UNIX sockets to open HTML documents and includes several attempts to read an HTML if the first one fails. It also detects and records apparent syntax errors in HTML documents, such as missing end quotes and right angle brackets, or absence of a title. Due to a variety of commonly occurring errors, there are literally thousands of bad HTML files. Making wwww robust against such files was a major challenge.

3.2.1. Archive format:

The WWWW archive file records all of the resources located in a previous search. The WWWW archive format is a list of URL references which can be in one of the following forms:

Table 4: WWWW Archive Format

T	HTML	Title	
R	URL	HTML	Reference
I	URL	HTML	Reference
C	HTML		

The T lines denote an HTML file encountered and opened for reading during a search and the second argument is then the URL of the HTML. The remainder of the line is the Title as taken from the <*title*> line required at the top of every HTML. Some files omit a title and in these cases the Title *"Zero Length Title"* is substituted.

The R line is used to denote a URL referenced in an HTML file which was opened for reading. In this case the second argument is the URL, and the third argument is the URL of the HTML file that referenced the URL. The rest of the line is then the hypertext anchored by the URL reference. As an example, suppose that file *http://mach/home.html* contains the line:

Click here for McBryan's Photo

Then the archive will record the entry:

R http://mach/photo.gif http://mach/home.html McBryan's Photo

This allows the URL to be associated with both the associated hypertext and with the title text of the containing HTML.

The I line is used to represent an inlined image. Inlined images are similar to the previous example, except that such an image has no associated hypertext. in these cases we associate only the title string of the containing HTML. However it is common to embed an inlined image in an anchor, such as:

<img src="photo.html">McBryan's photo.

In such a case we associate the adjoining hypertext *"McBryan's Photo"* with the image URL, as well as with the anchor URL *home.html*. The point here is to maximize the available contextual information.

The C line is used when an HTML that was previously opened is completed. This indicates that all HTML references within the HTML have been completely processed to all recursion levels.

3.2.2. Ensuring that WWWW is Well-Behaved

When wwww is run, it first reads the specified archive file, and records all HTML files associated with C lines as completed. Such files will not be read again. Even if an HTML has been only partially processed on a previous pass, some of the HTML references within the file may well have been fully processed and will therefore appear in C lines. In this way, wwww ensures that when run again it will access *only* files that it had not accessed on a previous pass. Of course since HTML files frequently change, it may be appropriate at times to start wwww from an empty archive in order to reread all HTML. Alternatively selected C lines may be deleted from the archive before running wwww, causing just these HTML to be reread and recursively processed.

Even a well-behaved robot should attempt to cooperate as much as possible with sites that it visits. WWWW provides the following information when it accesses a URL called *url* at a remote site:

```
GET url HTTP/1.0
Accept: text/html
User-Agent: WWWW - The WWW Worm version 1.1
From: mcbryan@cs.colorado.edu
```

this information allows the receiver to recognize the identity of the requester, determine that it is likely a robot, serve only text and HTML documents to it, and also provides an email address to correspond with if problems develop. M. Koster[7] has formulated a useful set of "Guidelines for Robot Writers".

3.3. WWWW Search Engine

The WWWW search engine is executed when a user accesses the WWWW home page and fills out a search Form. The engine supports two kinds of search: on HTML and on general URL. In the HTML case, only those HTML files that were located by wwww are searched. In the URL case, all URL references are searched. Keyword search functions are provided that operate on one or more of:

Table 5: WWWW Search Types
HTML Title Strings
URL Hypertext References
HTML Title string of HTML containing a URL
Any component of the URL name of a URL
Unions of components of the URL name.

The current search engine is based on the UNIX egrep program and simply creates an appropriate egrep search string based on the user's specified type of search and on the search keywords provided. Standard egrep wildcards and regular expressions are allowed.

In the case of HTML searches, the title strings of documents are returned as hypertext and clicking on these leads to the relevant document.

In the case of URL searches, the hypertext associated with the URL reference is returned, and again clicking on the hypertext leads to the URL. In this case the URL of the citing HTML file is also returned and can be clicked to reach the citing HTML file.

This simple search engine proves quite useful in practice. Search times are usually shorter than the time taken to return and display the results to the user. However because all searching occurs on one server, it places a significant computational and networking load on that one machine.

There is an interesting limitation of WWWW in that wwww can only find URL that are cited at some recursive level from the starting HTML arguments and archive file. A document or tree that is not referenced anywhere externally will certainly not be found by wwww. GENVL has proved useful here. We have created a specific VL within GENVL to record home pages that user's own which they find are not known to wwww. The WWWW home page carries a hypertext pointer to that GENVL page. This feature has proved very successful and as a result many whole trees have been added to WWWW's knowledge base that might otherwise have gone undetected.

4. Conclusions

GENVL and WWWW deal with different aspects of simplifying and organizing access to the enormous information resources of the World Wide Web. Both programs have proved immensely popular, resulting in thousands of accesses per day on a long-term basis. It is clear that more powerful cataloguing and resource discovery tools are needed however. Ideally these tools should interface also to other search services such as Archie [15] or Netfind[17].

References

[1] T. Berners-Lee, *"The WWW Virtual Library"*

[2] T. Berners-Lee *"Resources Classified by Service"*

[3] A. Mohammed, *"Computer Science Departments across the Web"*

[4] CERN,

"Virtual Libraries"

[5] M. Koster, *"ALIWEB"*

[6] E. Ebina, *"Free For All"*

[7] M. Koster, *"World Wide Web Wanderers, Spiders and Robots"*

[8] CUI, *"W3 Search Engines"*

[9] J. Fletcher, *"The Jumpstation"*

[10] R. Fielding, *"MOMspider -- Distribution Information"*

[11] C. Tronche, *"The WWWMM Robot"*

[12] M.F. Schwartz, A. Emtage, B. Kahle, B.C. Neuman,

 "A Comparison of Internet Resource Discovery Approaches" , Computing Systems 5(4), 1992.

[13] K. Harrenstein, M. Stahl and E. Feinler, *"NICName/Whois"*, RFC 954, SRI International, Oct. 1985

[14] CCITT/ISO, *"The Directory, Part 1: Overview of Concepts, Models and Services"*, CCITT/ISO, Gloucester, England, Dec. 1988. CCITT Reccomendations X.500/ISO DIS 9594-1.

[15] A. Emtage and P. Deutsch, *"Archie - An Electronic Directory Service for the Internet"*, Proc. Usenix Winter Conf., pp. 93-110, Jan. 1992.

[16] B. Kahle and A. Medlar, *"An Information System for Corporate Users: Wide Area Information Servers"*, ConneXions - The Interoperability Report, 5(11), pp. 2-9, Interop. Inc., Nov. 1991.

[17] M.F. Schwartz and P.G. Tsirigotis,

"Experience with a semantically Cognizant Internet White Pages Directory Tool" , J. Internetworking: Research and Experience", 2(1), pp. 23-50, Mar. 1991.

[18] M.F. Schwartz and C. Pu,

"Applying an Information Gathering Architecture to Netfind: A White Pages Tool for a Changing and Growing Internet" , University of Colorado CS Dept. Technical Report CU-CS-656-93, December 1993.