

## Internet Publishing Feature

# Publishing Textual Databases on the Web

## Part 2 examines HTML options for moving from the desktop to the Web

by Péter Jacsó

**P**ublishing a database using some of the advanced features of HTML 3.2 is the least expensive (although not the simplest) approach to make an existing collection of bibliographic, full-text, or directory data accessible on the Web. Often, it is also the only possibility for those whose Internet service provider does not allow the installation of Common Gateway Interface (CGI) scripts on the server. The reason for the prohibition is that CGI scripts can do a lot of harm on a server by deleting files and crashing the system. The pure HTML options have certain limitations, but they provide inexpensive tools for individuals and organizations with tight software budgets, such as libraries and information centers. The choices among the pure HTML, Java, and CGI or DLL (Dynamic Link Library) options are not exclusive. Neither are they exhaustive.

For a masterpiece example of highly functional and intuitive design that will please all the users all the time, see the Scholarly Electronic Publishing Bibliography (<http://info.lib.uk.edu/sepb/sepb.html>) of Charles Bailey, who pioneered the use of the Web in libraries and by librarians sooner than many of us could spell HTML correctly. This excellent bibliography is matched with equally excellent options that allow users to print in Microsoft Word or Adobe PDF format, search via Boolean operators, or navigate through well-structured HTML pages composed of and linked by major topical categories (Economic Issues, Legal Issues, Case Studies, etc.). I will illustrate the HTML option here using a simple Web prototype of an abstracting-and-indexing database that was created originally for local use and CD-ROM distribution.

### Repurposing the Original Database

The abstracting/indexing database in its desktop version filled a niche by providing access to the most recent 12 volumes of the journal *Social Process in Hawaii (SPIH)* by authors, author affiliation, descriptors, title and abstract words, and a combination of these. Such access to this valuable local source was not available at the conception of the project in any form. The database was created by students in my Abstracting/Indexing, Database Design, and Database Publishing courses. It was enhanced by adding searchable full-text records and page images for some of the articles. Currently, I am working on a simplified HTML-only version of it to demonstrate that a well-organized bibliography in pure HTML (although using frames) can be a useful and valid alternative to the fully-searchable desktop and CD-ROM database versions.

If you have created a bibliography with any bibliography formatting or database software, you are able to print the bibliography and the indexes to a file. This file in turn can then be fed to a decent word

processor that is endowed with HTML output capabilities. Up to this point the process is a no-brainer. However, a linear bibliography of several hundred screens would not be efficient to consult. It must be organized and broken into smaller units, such as subject bibliography (subj.htm) and author bibliography (auth.htm), and possibly into further subsets by the letter of the alphabet (subjA.htm, subjB.htm.). These subsets also can be easily created using the method described above.

### Choosing Primary Access Points

The HTML documents can be linked through a main page to allow users to consult first the HTML list of subject terms (descriptors) or authors. These in turn would display an index of the articles that

Elite, Economic Factors, Economic Planning or Ethnic Groups, Ethnic Identity, Ethnic Minorities, Ethnic Neighborhoods, Ethnic Relations, etc. These were assigned from the thesaurus of *Sociological Abstracts* and enhanced by additional terms of local importance, such as Native Hawaiians or Plantation Workers.

### Linking

The lists of author names and subject terms were linked to the author and subject indexes, respectively. As shown in Figure 1, clicking on the term Ethnic Neighborhoods in the left pane displays the title and the publication year of three articles in the right top pane, and also shows the first entry under the next subject term, Ethnic Relations. In turn, clicking on the title of the

Figure 1: The Social Process in Hawaii (SPIH) database

have the selected subject term or author, respectively. The index entries consisted of the title and the publication year. Of course, there could be other primary access points and indexes generated to allow a variety of access paths.

The prototype was kept simple, catering to the most common needs. Journal titles are certainly good candidates for a browsable list, in case someone wants to look up articles by journal source. In our case there was no need for this because all articles are from the same journal. A title list was not deemed important, as users rarely know the exact title of the articles. Keyword lists and sublists could be produced easily from the title and abstract fields, but due to the nature of subject indexing (discussed below), this was not essential. An index by publication year was irrelevant because it is less likely that users want to find articles by publication year alone. This also emphasizes the obvious limitation of the pure HTML approach. It can only offer a search (actually a link to the short index entries) by a single term at a time.

This compromise solution still can offer quite a powerful access mechanism especially because of the dominance of specific, often compound subject terms, such as Economic Change, Economic Conditions, Economic Development, Economic

article about Race and Ethnic Relations (1982) displays the full record for Andrew Lind's article in the bottom right pane. Creating the links was easy because in Inmagic DB/Textworks (the software that was used for the original database) the designer has many options to add suffixes and/or prefixes to variables, thus it was simple to add the `<HREF A=.....>` prefix and the `</A>` suffix to the bibliographic data elements for the links automatically.

Even if this possibility does not exist in the software used to create a database, such prefixes and suffixes can be added through smart search and replace operations in the word processor used to convert the text file into an HTML file. It is to be noted that although DB/Textworks does offer printing output directly in HTML format, this option in itself does not create the links between these files!

### Anchoring

Instead of requiring the user to wade through hundreds of entries in the index page or in the full record page, the links were generated to point to a named section in the target page. For example, the anchor entry for the above example looks like this: `<A HREF="subjectindex.htm#Ethnic Neighborhoods">Ethnic Neighborhoods </A>`

The section had to be marked in the target files, but this could also be automated: `<A NAME="Ethnic Neighborhoods">Ethnic Neighborhoods </A>`

These anchored references make the browser jump and show the exact section of the document where the author name or the subject heading appears, instead of displaying the entire file from its very first line.

Even in the case of relatively short indexes of a few hundred entries, this would be very inconvenient when searching for articles by the author Yamamoto, or on the subject of Young Adults that appears at the very end of the documents. This type of anchoring could have been implemented also for the author list and subject term lists to act like a thumbnail index. Because they are relatively short and they have single-line entries, the users can as quickly spot the sought-after term by scrolling down a few screens as by clicking first on the letter of the alphabet to get in the alphabetic neighborhood. It is a judgment call when to add yet another tool (and yet another click) to cut to the chase. Had a keyword index been generated from the abstract and title fields, such an alphabetic pointer certainly would have been needed.

### Framing

Originally, the component files completely overlapped the previous screen, i.e., after clicking on an entry in the author name list, the author index screen replaced the author name list. Two buttons (BACK to NAMES and BACK to SUBJECT TERMS) took the user back to the list of primary access points. (This is the same function as the browser's BACK button but offers a little extra help for novice users.) The full record screen overlapped the author and subject index screens.

It obviously makes the navigation more intuitive to keep all three components on the screen in three panes, i.e., the name or subject list, the author or subject index, and the full record. This alternative required the use of frames. Using frames has two disadvantages. One is that getting the frame references correct is a somewhat involved procedure, although it does not require a Ph.D. The other is that users with old browsers cannot display frames. Good Samaritans are usually willing to develop a no-frames version, but in this example it would not have been worth the compromises. Hopefully, users will be willing to install a frames-capable browser to get access to this database.

In spite of the limitations of the pure HTML version (no Boolean searches, no searching by words in the title or abstract), it offers a convenient way to port mainframe and desktop databases to the Web with an intuitive interface. In the next article in this series, I will discuss the Java and CGI options for porting a database from the desktop to the Web.

*Péter Jacsó is associate professor of library and information science at the department of information and computer sciences at the University of Hawaii. He won the 1998 Louis Shores—Oryx Press Award from ALA's Reference and User Services Association for his discerning database reviews. His e-mail address is [jacs@hawaii.edu](mailto:jacs@hawaii.edu).*