Dressing up SGML for the Web

A look at UNL's project to create electronic books

by DeeAnn Allison and Jon Keene

Technology has made it easier for libraries to provide access to information. Online catalogs and electronic reference databases are commonplace in modern libraries. The latest technology makes it easier and more cost effective for libraries to move into the field of digitization. Many libraries have electronic text conversion projects in various stages of implementation.

The project at the University of Nebraska-Lincoln (UNL) is a cooperative venture that includes the university libraries, the university press, and faculty representing several humanities areas.[1] A major goal of the project is to produce electronic versions of books that can be used for classroom and research applications. The end product must be a faithful reproduction of the original text that is easy to read. It is important that the electronic version not lose the familiar organization of the book (including page numbers and a table of contents), and allows the reader to move through the text in a natural fashion.

Determining formats for electronic text conversion

Since it is important to preserve as much of the original print presentation as possible, the selection of electronic format is very important. For example, will documents be digitized as images or as text? Images have the advantage of appearing exactly the same as they did in the book, while text can be manipulated in a variety of ways to add value to the end product. If documents are to be digitized as text, the selection of a markup language is crucial. Markup language can determine not only how a document displays but has profound implications for preservation. The UNL Libraries decided to use Standard Generalized Markup Language (SGML). SGML is an international standard (ISO 8879) that is used when converting books into a format that can be stored on a computer. It is a recognized standard that is supported by many applications and is widely used by publishers.

What is SGML?

SGML provides a standard format for managing a document's format and structure. It defines the markup language to use for the document, the character sets used in the document, defines the document structure, and defines the elements of the document. From a technical standpoint, this has the advantage of handling groups of documents the same way rather than treating each document as a unique item. By defining document types, SGML provides a blueprint for how each item is to be presented to readers.

About the authors

DeeAnn Allison is coordinator for automated systems at the University of Nebraska-Lincoln, e-mail: deeanna@unlib.unl.edu; Jon Keene is network specialist at the University of Nebraska-Lincoln, e-mail: jonk@unlib.unl.edu

The major difference between SGML and HyperText Markup Language (html) is that it describes the document, not the formatting. It uses descriptive markup for the document structure like "author," "title," "language," "chapter," and "paragraph," which frees authors from the tedious activity of coding. Like html, it is hardware independent, with the software package doing the interpretation. Any software package that understands SGML can read and manipulate the data, which makes it useful for a variety of applications and for maintaining an archival copy.

The problem of providing Web access for SGML coded documents

One of the limitations of SGML is delivering it over the Web. Because it is not using the standard html coding, Web browsers cannot display it without installing a viewer. Commercial viewers, which a user can purchase and install on his or her computer to read SGML through a Web browser, are available from vendors like Interleaf.

However, one of the goals of the UNL project is to make the digitized texts available to a wide audience. As a consequence, any solution that requires end users to purchase software to use the digitized texts is not acceptable.

Likewise Internet users have a variety of computers and operating systems, so it is also important for the solution to be platform independent. Finally, it is important to control the staffing and disk space costs by eliminating the tedious task of creating and maintaining multiple copies of documents in two formats that both used disk space.

For these reasons, UNL is experimenting with an alternative where computer programs in the Perl language are written to be executed by the Web server when a link for an SGML document is clicked. The computer program converts the SGML to html "on-the-fly" so the document is displayed in html format for the browser. This leaves the document in SGML format for archival purposes. The conversion process is transparent to the user who views the document through the Web browser on his or her local computer.

Since the html file is dynamically produced, there is no second, permanent file that uses disk space or requires maintenance when editing changes are made to the SGML file. Also, because SGML uses standard document types, a single Perl script can be used for more than one document.

Preserving the book's look and feel

Many digitized books lose much of the original design of the book. Books are converted into very long files that are difficult to navigate in any other way than by sequentially moving from beginning to end. UNL is experimenting with a new way to display converted SGML coding.

The approach is to convert all SGML into html and display the document in one frame with the navigation bar in a second frame. This preserves the look of the book, which is displayed as text with graphics in one frame, while adding a navigation bar built from the elements provided by the SGML coding. The element coding of SGML creates a table of contents to add value to an otherwise long and difficult-to-navigate html file.

Two programs were developed: *tei2html*, which converts the SGML document into the html frame, and *navbar*, which creates the navigational frame from coding in the SGML document.

The *tei2html* program

The *tei2html* program converts TEILITE Document Type Definition (DTD) coding into html. The program starts by preprocessing the SGML file using a program called NSGMLS, a widely used program written by James Clark, which parses and validates an SGML document based on its DTD.

The output of the NSGMLS program is the SGML file content with its structure information in a format that greatly simplifies the next step in the conversion process. The *tei2html* program takes the output of the NSGMLS program as it is processed by the server. As each SGML identifier is encountered, it is converted to the appropriate html code.

Since there is no one-to-one correspondence between SGML and html, the program must keep track of the tag's location, which determines its context and the corresponding html code. For example, the html encoding of the SGML tag <TITLE> will be different depending on whether the tag occurs in the title statement or the bibliographic section or elsewhere.

(continued on page 304)

Continuing relationships

Outreach to your teaching colleagues need not stop here. The success of our article in *The Teaching Professor* led to the authorship of a second piece, this one on choosing appropriate search tools on the Web. We have been invited to address other topics for this newsletter. It has become clear to us that we have technology-related knowledge that is not common among teaching faculty overall, and that this knowledge is eagerly sought.

We urge you to share what you know with those who teach in the classroom. Librarians have done an excellent job of sharing our ideas with each other. We have been on the cutting edge in devising ways to make intelligent use of the Web. It is time to take what we know and share it.

Notes

1. There is a wide array of criteria checklists available. A handy compendium can be found at Susan Beck's Web site: http://lib.nmsu.edu/staff/susabeck/checs98.html#method. Esther Grassian's checklist, Thinking Critically about World Wide Web Resources, is one of the earliest (http://www.library.ucla.edu/libraries/college/instruct/web/critical.htm).

2. Marsha Tate and Jan Alexander, "Teaching Critical Evaluation Skills for World Wide Web Resources," *Computers in Libraries* 16, no.10 (1996): 49–54.

3. Trudi E. Jacobson and Laura B. Cohen, "Teaching Students to Evaluate Internet Sites." *The Teaching Professor*, 11, no.7 (August/September 1997): 4. This article is also available at: http://www.albany.edu/library/internet/teaching.html.

4. Keith Gresham, "Surfing with a Purpose: Process and Strategy Put to the Test on the Internet," *EDUCOM Review* 33, no.5 (September/October 1998): 22–29.

5. Susan A. Gardner, Hiltraut H. Benham and Bridget M. Newell, "Oh, What a Tangled Web We've Woven! Helping Students Evaluate Sources," *English Journal* 84, no.1 (1999): 39–44 and Karen Hartman and Ernest Ackermann, "Finding Quality Information on the Internet: Tips and Guidelines," *Syllabus* 13, no.1 (August 1999): 52–54. ■

("Dressing up SGML . . ." continued from page 294)

Once the processing is complete, the SGML file is displayed as a temporary html file that the computer automatically removes when the user disconnects. Since the html version is created dynamically when the user selects a hypertext link, and is deleted when the user navigates to another Web page, there is only one permanent file for any document. This means that only one file must be edited whenever corrections or revisions are needed.

An additional feature of the program is the creation of navigational footnotes from the SGML. All the notes are collected at the end of the html document with hypertext links so that the reader may jump to a note and back to the text.

Creating navigational links for the converted document

An auxiliary program called *navbar* was written to produce the html code for the hypertext links that display in a frame to the left of the html document. These links function as a hypertext table of contents, giving the user the ability to jump from one section to another. This imitates the printed version of the text by creating a method for jumping to specific sections and scanning the e-text for specific parts. This was critical for the poetry books completed early in the project. The navigation bar enables users to scroll through the volume's contents and jump to specific poems.

Although transferring printed material to the Web poses many challenges, it also provides this generation of librarians with the opportunity to improve on the design of the book. Innovative projects like this one undertaken at UNL unite the process of information preservation with information redesign, giving us the opportunity to enhance the end product.

Note

1. More information on the UNL e-text project and examples of digitized texts can be found at http://libr.unl.edu:2000. ■