Integrating Local and CRI Online Documentation Using SGML and DynaWeb

T. R. Girill
Lawrence Livermore National Laboratory
University of California
P. O. Box 808, L-72, Livermore, CA 94551
trg@llnl.gov
Virginia (Jean) Shuler
Lawrence Livermore National Laboratory
University of California
P. O. Box 808, L-67, Livermore, CA 94551
jshuler@llnl.gov

ABSTRACT:

This paper tells how Lawrence Livermore National Laboratory enriched CRI's online documentation set by publishing local manuals using the same SGML DTD used by CRI and delivered using (a more sophisticated version of) the same World Wide Web server (DynaWeb 3.0). This approach supports flexible local content and styles, yet integrates local and CRI manuals through one access mechanism and user interface. We explain the basic strategy involved, compare the benefits of this approach with three alternatives, and discuss the problems to which it gives rise.

KEYWORDS:

client-server publishing, CrayDoc, DynaWeb, DynaText, HTML, information retrieval, online documentation, SGML, Standard Generalized Markup Language, text encoding, usability, user services

Introduction

This paper tells how the Livermore Computing [Center] at Lawrence Livermore National Laboratory (LLNL) enriched CRI's online documentation set by publishing additional, local manuals encoded using the same SGML document type definition (DTD) used by CRI and delivered using (a more sophisticated version of) the same World Wide Web server (DynaWeb 3.0). This approach allows completely flexible local content and updates, and independent local styles, yet it integrates local and CRI manuals through one server address and one user interface for greater reader convenience in answering questions. After explaining the basic strategy, we compare the benefits of this approach with three alternatives (including native HTML), and we end by discussing the problems inherent in such an on-the-fly SGML-to-HTML publishing process.

The SGML Strategy

Our approach to publishing local documentation relies on the distinction between managing information and displaying information that underlies the design of the Standard Generalized Markup Language (SGML) itself. We prepare our user manuals (so far, about 3300 pages in 13 books) with the rich encoding that true SGML makes possible, to capture the nested structure of the document sections and to reflect and preserve important content distinctions. In fact, we use the same document type definition (DTD), or inventory of SGML elements and their relations, that Cray Research Inc. uses to manage the source files for their own online reference manuals. Also, publishing technical material in SGML is a goal of the U.S. Department of Energy, for which LLNL is a contractor [1].

We then treat HTML, with its sparse content and structural distinctions, as a formatting or display language. Because it is widely used, HTML serves to present our SGML-encoded material to familiar Web clients with visual convenience, but without limiting the basic representation of the technical content. Using HTML merely to deliver SGML is not a new strategy (e.g., see [2]), but, when well executed, it proves a very practical one.

EBT Software Tools

To execute this strategy well we took advantage of the clever software engineering built into DynaWeb 3.0, a commercial product developed by Electronic Book Technologies (now INSO Providence Corporation). DynaWeb has all the usual features of a standard HTTP server, but it is also able to deliver native SGML-encoded documents directly to HTML clients (such as Netscape, Mosaic, or Internet Explorer) by using an elaborate, carefully designed framework of publisher-controlled, on-the-fly conversions. The DynaWeb server thus enables us to prepare material in content-aware SGML and yet have passages from it delivered smoothly and flexibly in HTML just when inquiring clients request them.

DynaWeb is not the only product to perform run-time conversions from SGML to HTML. But its design is perhaps the most sophisticated, driven by a thoughtful awareness of both the general value of SGML and of the specific needs of serious technical publishers. These engineering benefits no doubt explain Cray's choice of (an earlier version of) DynaWeb to support their own documentation delivery on the World Wide Web, and indeed their choice of client software (DynaText) from the same company as the basis for their even earlier CrayDoc tool. (Subsequent sections will compare these three alternatives in more detail.) So acquiring a DynaWeb (and support tool) license was the first step in carrying out our plan for local SGML-based documentation.

Document Preparation

One benefit of DynaWeb is that it accepts as input SGML files encoded in accordance with *any* DTD. Mechanisms exist to let many documents share the same DTD easily (as we do), or instead to encode each of many documents using quite different DTDs to support divergent structures or topics. Thus besides its immediate practicality, its future generalizability made this approach appealing.

This server, like most Web servers, is also indifferent to the tool used to construct its input files. We can therefore prepare our SGML documents using any text editor or word processor, or dedicated (but more expensive) SGML authoring software. We check all our SGML source files with an SGML parser to detect faulty nesting or mismatched tags, or other tagging errors.

To accelerate its run-time conversions from SGML to HTML, DynaWeb relies on lookup tables and index files prepared in advance, when each document is "released" for display. We purchased a license for the EBT software (e.g., MKBOOK) to build, place, manage, and update these helper files along with our DynaWeb license, and access to these tools is essential for publishing local documents and maintaining an often-changing documentation database. This software processes our SGML files after we have edited and parsed them, to enable their subsequent rapid delivery to users through the DynaWeb server.

Server Configuration

Virtually every aspect of the DynaWeb server (even the icons on the buttons it offers) can be locally configured to adapt to the material being published and to the users who are trying to read that material [3]. Many appropriate default choices are available, but three features must be specified by each SGML publisher:

Collection titles
: Served documents are grouped into (one or more) collections, and those managing the server specify the descriptive (public) name and the (private) pathname for each collection. The server software and the collection(s) it delivers need not reside in the same directories or file systems, a convenience for managing disk space.

Book titles
: Served documents are delivered using a book metaphor, and the publisher specifies the descriptive (public) name and the (private) pathname for each book in each collection. Collection and book titles are easy to specify and change, but the DynaWeb server must be restarted for users to see these changes, sometimes an inconvenience.

SGML-to-HTML mapping
: Each DTD used requires a corresponding translation file to specify how the publisher wants the DTD's SGML

elements presented in HTML by the server. (This translation file is itself coded in SGML.) DynaWeb's translation machinery is constrained by the limits of HTML, of course, but it is interpretively elaborate. For example, it allows an element to be presented differently in different contexts (so paragraphs inside lists could look different than those outside lists). Conditional tests are available, along with the ability to add or hide content (such as labels) at the time of presentation. And while many documents sharing a DTD can easily share a mapping file as well (our normal practice), each document can have a distinct mapping file if this is desired. We chose to borrow heavily from the mapping file by which EBT presents its own SGML documentation, but one could instead borrow from the Cray DynaWeb mapping file for a somewhat different output style.

Integration with CRI Documents

One side benefit of this approach to delivering SGML-encoded manuals is that pointers (hypertext links) to any DynaWeb document use the same URL syntax familiar from other links on the World Wide Web. Hence, locally developed manuals can point or link to SGML-encoded manuals published by Cray Research and delivered on Cray's own DynaWeb server simply by citing the target document's URL, as reported by any WWW client while displaying that target.

For instance, a selective summary of locally relevant UNICOS 9.0 differences developed at LLNL refers readers to CRI's *UNICOS 9.0 Release Overview RO-5000* for additional details simply by containing the SGML element

```
<ULINK URL="http://www-lc.llnl.gov:8080/library/all/RO-5000">string</ULINK>
```

where

8080
: is the port for CRI's server on LLNL's host www-lc.llnl.gov, and

RO-5000
: is the (private) file name used in CRI's DynaWeb directory structure for the document they assigned the public name *UNICOS 9.0 Release Overview* , and

string
: is any part of the text of the citing local document.

Our SGML-to-HTML mapping file only needs to convert this SGML element into the usual anchor (A) tag in HTML,

```
<A HREF="http://www-lc.llnl.gov:8080/library/all/RO-5000">string</A>
```

for any HTML client to handle it correctly and retrieve the cited manual from CRI's LLNL DynaWeb server.

A natural extension of such individual links to DynaWeb-served CRI manuals would be a descriptive directory of every available CRI manual, with each entry linked to the corresponding Cray document for immediate retrieval. LLNL users can thus answer questions by consulting locally developed and CRI online manuals together (and easily pursue cross references among all of them) by starting with the single collection offered by LLNL's own DynaWeb server.

Comparative Benefits

At least three alternatives to the foregoing approach were considered. This section explains their relative merits and tells why we favored the chosen approach over all the others.

Comparison with CrayDoc (CDOC)

CrayDoc (CDOC) is Cray Research's licensed and customized version of DynaText, a graphical user interface (client) developed by Electronic Book Technologies to display and navigate native SGML publications without recourse to

HTML [4]. Although the DynaText, and hence the CrayDoc, client includes many features well suited to delivering complex, SGML-encoded technical material online, perhaps its two most useful visual characteristics are:

(1) offering an interactive, scrollable table of contents (of section headings) in a separate pane along side each publication's text, so that the hierarchical structure is revealed at a glance and can be used to guide the display of text sections in the other pane, and

(2) optionally spawning a new window for each selected hypertext link, so that readers can see and easily compare the text on both ends of the link at the same time.

Despite its engineering strengths, three practical drawbacks led us somewhat reluctantly to decide against relying on the DynaText-CrayDoc client software as the way to deliver documentation to LLNL users. First, users would have to learn, remember, and execute a special documentation-only client distinct from the WWW client (such as Netscape) that they were already familiar with and probably already running to retrieve other information every day. This seemed a counterproductive competition. Second, DynaText-CrayDoc requires X Windows support (and correspondingly elaborate set up for each user). X terminals are not always available to all our users, however, while many work on personal computers without invoking X Windows. And third, CrayDoc is no longer supported by Cray Research, forestalling any bug fixes or refinements.

Fortunately, version 3.0 of the SGML-to-HTML DynaWeb server is able to use frames and other recent HTML techniques to cause most ordinary WWW clients to replicate the two key DynaText-CrayDoc features mentioned above (shared-view, interactive tables of contents and dual-end link display). So we were able to offer users approximately the same benefits as DynaText-CrayDoc, but without the three drawbacks cited here.

Comparison with Raw HTML

The most obvious alternative that we rejected was to simply publish local user manuals in raw HTML. Technical documentation routinely presents three demands that HTML meets poorly but that full SGML meets very well.

VOCABULARY.
HTML offers only a limited vocabulary of elements (and hence tags), most of which merely control font size and style. SGML document type definitions (DTDs) for computer documentation, such as the DOCBOOK DTD used by LLNL and CRI, offer a much larger and more content-oriented element vocabulary. Identifying examples and their parts, for instance, or identifying commands and their components, typify the content distinctions this richer SGML vocabulary supports. It also enables adding metadata, such as index or keyword entries, that can make the text easier to manage and retrieve. And with the DynaWeb server's ability to automatically (after configuration) use HTML to display the many additional roles, distinctions, and combinations encoded in SGML in documentation source files, taking advantage of that richer vocabulary was straightforward.

HIERARCHY.
One of the ways that editors and information developers add value to the computer documentation they draft is by carefully constructing its hierarchy of sections and subsections to adequately reflect the structure of the technical topics discussed, their priority or parallelism, and the inclusion of some topics within others. But with the exception of lists and tables, raw HTML is incapable of encoding this text hierarchy. Most HTML elements support no parent-child relations. In HTML, "sections" are not nested containers but just sequential section headings (<h1>, <h2>, etc.) for which no correct sequence is enforced, much less any true hierarchy. Even "paragraphs" are just locations (of <p> tags), not envelopes with content inside.

Documentation DTDs, on the other hand, deploy SGML's full ability to accurately represent text hierarchy, priority, and parallelism many layers deep, as often occurs in typical software, library, and system-administration manuals. Preserving a text's extensive parent-child relations with nested SGML elements enables readers to recognize and exploit these relations if they later view the text using SGML-aware delivery tools. The DynaWeb server is such a tool. In response to each client request for documentation, the server automatically generates a complete, hierarchical table of contents (TOC), with each heading a hypertext link to its corresponding text section, and

- shows readers this TOC guide to the text's nested structure simultaneously with the text itself,
- lets readers navigate to any section at any level (repeatedly, if they wish) by selecting a desired TOC heading,
- offers previous-section and next-section buttons for every section (automatically generated from its knowledge of text organization), and

- lets publishers control the *default* grain size of each TOC (optionally suppressing some levels of heading), yet lets each reader expand or contract that grain size at run time to suit their personal needs.

None of this is possible with text encoded in raw HTML because the underlying, multilevel, parent-child hierarchy simply cannot be adequately represented.

SCOPE.
Very large documents encoded in raw HTML always present a dilemma for their publishers. If kept in a single file, they are easy to edit, manage, and search, but then the whole large file must be transmitted to every requesting client, including much that is irrelevant, before the client can display any small relevant section within the document. Dividing the large document into many small pieces allows the fast transmission of each piece alone, but then the resulting multitude of little files is hard to edit, manage, and use in any comprehensive way.

Encoding large documents with elements from an appropriate SGML DTD and then delivering them with SGML-aware software creates a way to go between the horns of this dilemma. We make, manage, and update a single SGML source file for each of our locally published manuals, with the DynaWeb server accepts as input. The server then uses the structural information conveyed by the SGML tags to respond to each client request for a specific section by locating that section, generating on the fly an HTML-displayable version of only that section, and sending just that part to the client without transmitting HTML for the other, much larger, irrelevant parts of the file too. For example, readers can quickly see in HTML the 130-line section describing the DCHEX library routine, which the DynaWeb server extracts for them from the midst of one of LLNL's 750-page *SLATEC Mathematics Library Reference Manuals*, while the SGML-encoded manual itself remains whole for other purposes.

Comparison with DynaWeb Version 2.0

Because Cray's licensed and adapted server, DynaWeb version 2.0, was already in service at LLNL delivering Cray manuals, we considered simply using it to deliver our locally developed manuals as well. Version 2.0 is strategically the same as version 3.0 (both are SGML-aware standard HTTP servers with strong usability engineering). But version 2.0 has a few tactical weaknesses that led us to prefer using the newly available version 3.0 of DynaWeb instead.

First, the later version is more easily configurable and has several defaults (such as button icons) that we judged to be more helpful. Second, version 3.0 automatically splits the client's display window into side-by-side panes, so that the table of contents for a document appears next to the text whose use it cues. This mimics DynaText behavior and is a significant human-factors improvement over version 2.0, in whose output the table of contents appears above the text in a single window pane, which often prevents a reader from seeing both at once.

DynaWeb version 3.0 clearly takes advantage of HTML "frames" to achieve its improved output, and its third advantage is its ability to revert to the simplier (but psychologically weaker) version-2.0 behavior for any client that cannot support frames. This allows a greater variety of clients to be served successfully, and indeed the opportunity for readers to toggle between frames and no-frames output is offered from a button presented on every screen.

Side Effects and Problems

Generating HTML at run time to display requested parts of underlying documents encoded in SGML has some negative side effects. Fortunately, reasonable technical solutions exist to overcome all of them.

Links to Specific Sections

As the Integration section above mentioned, a hypertext link to a whole document served by DynaWeb is achieved just by constructing a URL with the usual format, starting with the server's domain name and port, for example:

```
http://www-lc.llnl.gov:6336/dynaweb/LCdocs/ezfiles
```

To link one passage to another within the same document also begins normally, by citing (in SGML) the unique identifying string assigned to the target, for example:

```
<LINK LINKEND="global-quota">See GLOBAL_QUOTA</LINK>
```

Because the server generates HTML on the fly, only in response to reader requests, however, DynaWeb cannot simply translate this intended link as the normal HTML

```
<A HREF="#global-quota">See GLOBAL_QUOTA</A>.
```

No HTML target section, including the one that would have the identifier *global-quota* , exists in advance of its being requested by some client. To avoid this circularity in identifying link targets, a DynaWeb support tool (MKBOOK) creates an index of all the SGML elements in a book. This enables the SGML-to-HTML translator to automatically generate a URL requesting a search that retrieves the intended target for every (internal) link by using its indexed address. In this case, the URL for the above link automatically becomes:

```
http://www-lc.llnl.gov:6336/dynaweb/LCdocs/ezfiles/@ebt-link;
uf=0?window=next;target=%25N%14_1287_START_RESTART_N%25
```

The same problem arises when trying to link from one document to a *specific section* in a second document served by DynaWeb. Again, the target HTML section does not exist until it is created when someone tries to follow the link. And again, the solution relies on the index of SGML elements in the target document to provide a nontransitory identifier for the intended target section. For a separate document this cannot be discovered automatically, however, so the publisher must view the target passage and manually insert its browser-reported URL in the external link that should lead to it. Thus LLNL's introductory output guide links from a summary of the (local) NFT utility to the command dictionary inside the NFT reference manual by using this technique and the URL

```
http://www-lc.llnl.gov:6336/dynaweb/LCdocs/nft/@Generic__BookView/1281
```

The same approach applies with external links into specific sections of the Cray SGML documents served by Cray's DynaWeb version 2.0 server, except that the indexing is slightly different. For example, to arrive specifically at Appendix E (Disk Capacities) of the *UNICOS Basic Administration Guide* calls for using the URL

```
http://www-lc.llnl.gov:8080/library/all/SG-2416/32275#X
```

Printing Documents

Browser-supported local printing of sections served in HTML by DynaWeb 3.0 from SGML source documents occurs in the usual way, complicated by the (usual) care needed to pick the right frame before making a print request. But no comprehensive HTML file exists to meet the needs of readers who want to print an entire document. Browser printing of large manuals is often crude in any case, usually lacking the page numbers and running headings helpful for managing many pages of output.

A plausible alternative is to separately generate for each SGML manual a file with appropriate fonts and page numbers, intended just for printing. These output-only files could then be cataloged and offered from a globally-readable storage directory or (perhaps with the aid of an interactive form) delivered by FTP. The special EBT DynaText client, mentioned above, makes such suitable paginated output from the same SGML source and index files that feed DynaWeb, suggesting this as a convenient tool to mediate such whole-book printing.

Testing and Verification

Privately testing draft versions of a new manual or elaborate updates of an old one before their public release online is prudent practice. Because DynaWeb translates SGML to HTML on the fly, however, the usual technique of asking an HTML client to display a private (local) HTML file for previewing will not work. One easy alternative is to create a separate "collection" (partitioned, labeled set) of dummy SGML documents into which various test versions are swapped whenever verification is needed. After a modest one-time preparation of this framework, using it to test SGML drafts is just as straightforward, and yields just the same benefits, as viewing local HTML files with a browser.

Extended Access Aids

By default, DynaWeb offers generous support for full text searches. Users can limit their searches by scope (one book or a complete collection) as well as by kind (exact word searches, wildcards, proximity control, or overt Boolean combinations). But as a local documentation collection grows to dozens of manuals and thousands of pages, other proven access aids become increasingly important for isolating relevant answers amid much material irrelevant to any single query:

- the ability to search across multiple collections (perhaps even those delivered by multiple servers),
- vocabulary enhancement or synonym assignment to enable successful searching on words not actually in the text [5],
- approximate syntactic matching of search terms to help handle unusual terms or spelling errors [6], and
- SGML-aware searching limited by element (for instance, headings or examples only).

Past experience suggests that all of these extended access aids could be added to LLNL's basic SGML-based documentation service, probably gradually, using external lookup tables and software to exploit such tables. DynaWeb's customizability will simplify integrating such software extensions into the current documentation interface as they are needed.

Conclusion

The model of SGML-aware publishing of online documentation demonstrated by Cray Research, and implemented first with the CrayDoc (DynaText) client and then with the DynaWeb version 2.0 server, embodies many astute engineering decisions. We have extended that model to include LLNL local user documentation by means of DynaWeb version 3.0 and its related software tools. The result is an effective, integrated documentation service with clear advantages over its alternatives.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract W-7405-Eng-48. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

References

[1] Office of Scientific and Technical Information, **Electronic Exchange of Scientific and Technical Information Strategic Plan**. U.S. Department of Energy, Oak Ridge, TN, January 1993.

[2] John Price-Wilkin, "Using the World-Wide Web to Deliver Complex Electronic Documents: Implications for Libraries," **Public Access Computer Systems Review**, 5 (no. 3, 1994), 5-21.

[3] **DynaWeb Publisher's Guide (Release 3.0)**, EBT INSO Providence Corporation, Providence, RI, 1997.

[4] **DynaText System Publisher Guide (Release 2.3)**, vol. 1, Electronic Book Technologies Inc., Providence, RI, 1994.

[5] T.R. Girill, Thomas Griffin, and Robert B. Jones, "Extended Subject Access to Hypertext Online Documentation. Parts I and II: The Search-Support and Maintenance Problems," **Journal of the American Society for Information Science**, 42 (July 1991), 414-426.

[6] T.R. Girill and Clement H. Luk, "Fuzzy Matching as a Retrieval Enabling Technique for Digital Libraries," **The Digital Revolution: Proceedings of the American Society for Information Science Mid-Year Meeting**, (San Diego, CA, May 1996), pp. 139-145. Information Today, Inc., Medford, NJ, 1996.

Author Biographies

Virginia (Jean) Shuler is the Group Leader for the Customer Services and Support Group in the Scientific Computing and Communications Division at Lawrence Livermore National Laboratory. She was the Group Leader of the User Services Group at the National Energy Research Supercomputer Center for 8 years while NERSC was located at LLNL. She has been very active with the Cray User Group for several years. She was the CUG Newsletter Editor, Chair for the User Services Special Interest Committee, Vice President, and is currently the Chair for the J90 Mutual Interest Group.

T.R. Girill publishes computer documentation at Livermore Computing [Center], Lawrence Livermore National Laboratory. He led the documentation project at DOE's National Energy Research Supercomputer Center from 1982 through 1996, served as associate editor of *Technical Communication* from 1983 through 1990, and chaired the Encoding Committee of the DOE SGML Technical Working Group from 1992 through 1994. He has taught computer documentation classes in the extension programs at UCLA and UC Santa Cruz, and he now serves as editor in chief of the Association for Computing Machinery's *Journal of Computer Documentation*. He is also an Associate Fellow of the Society for Technical Communication.

UCRL-JC-127306

TRG (22Apr97) Contact: trg@llnl.gov
