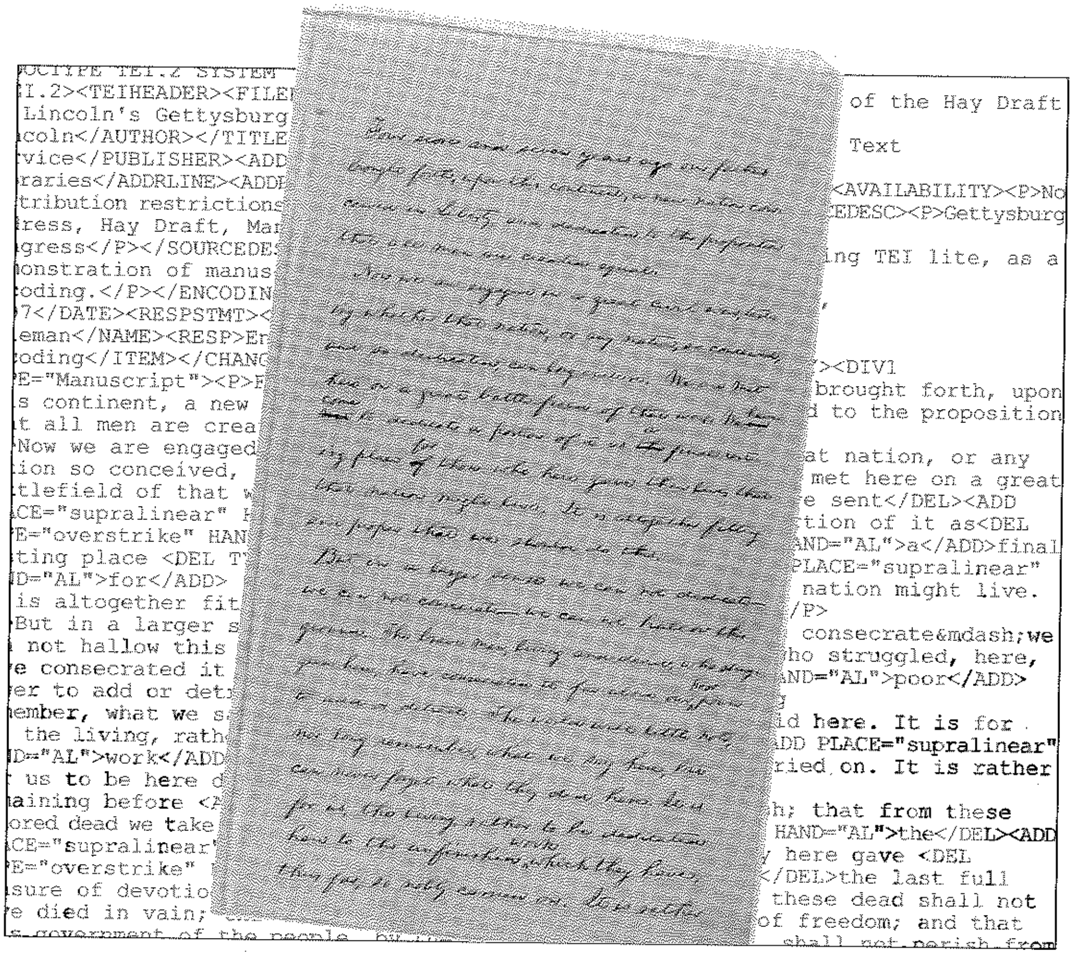
The Commission on Preservation & Access

SGML as a Framework for Digital Preservation and Access



```
OCTYPE TEI.2 SYSTEM
I.2><TEIHEADER><FILE
Lincoln's Gettysburg
coln</AUTHOR></TITLE
vice</PUBLISHER><ADD
raries</ADDRLINE><ADDI
tribution restrictions
ress, Hay Draft, Mar
gress</P></SOURCEDES
onstration of manus
oding.</P></ENCODIN
7</DATE><RESPSTMT><
eman</NAME><RESP>En
oding</ITEM></CHANG
E="Manuscript"><P>F
s continent, a new
t all men are crea
Now we are engaged
ion so conceived,
tlefield of that w
CE="supralinear" H
E="overstrike" HAN
ting place <DEL T
D="AL">for</ADD>
is altogether fit
But in a larger s
 not hallow this
e consecrated it
er to add or detr
ember, what we s
 the living, rath
D="AL">work</ADD
 us to be here d
aining before <A
ored dead we take
CE="supralinear"
E="overstrike"
sure of devotio
e died in vain;
s government of the people, by
```

```
of the Hay Draft

Text

<AVAILABILITY><P>No
EDESC><P>Gettysburg

ing TEI lite, as a

/><DIV1
brought forth, upon
d to the proposition

at nation, or any
 met here on a great
e sent</DEL><ADD
tion of it as<DEL
AND="AL">a</ADD>final
PLACE="supralinear"
 nation might live.
/P>
 consecrate&mdash;we
who struggled, here,
AND="AL">poor</ADD>

id here. It is for
ADD PLACE="supralinear"
ried on. It is rather

h; that from these
 HAND="AL">the</DEL><ADD
/ here gave <DEL
</DEL>the last full
 these dead shall not
of freedom; and that
 shall not perish from
```

July 1997

The Commission on Preservation and Access is a program
of the Council on Library and Information Resources.

Published by
The Commission on Preservation and Access
A program of the Council on Library and Information Resources
1400 16th Street, NW, Suite 740
Washington, DC 20036-2117
E-mail: info@CLIR.org
WWW: http://www-cpa.stanford.edu/cpa.html

Reports from the Commission on Preservation and Access are intended to stimulate thought and discussion. They do not necessarily reflect the views of the Commission or the Council on Library and Information Resources.

The image of the Gettysburg Address (known as The Hay Draft or Second Draft) is used with permission. It can be viewed on the Library of Congress' Exhibitions site.
Exhibitions Home Page: http://lcweb.loc.gov/exhibits/archives/intro.html
Library of Congress Home Page: http://www.loc.gov/

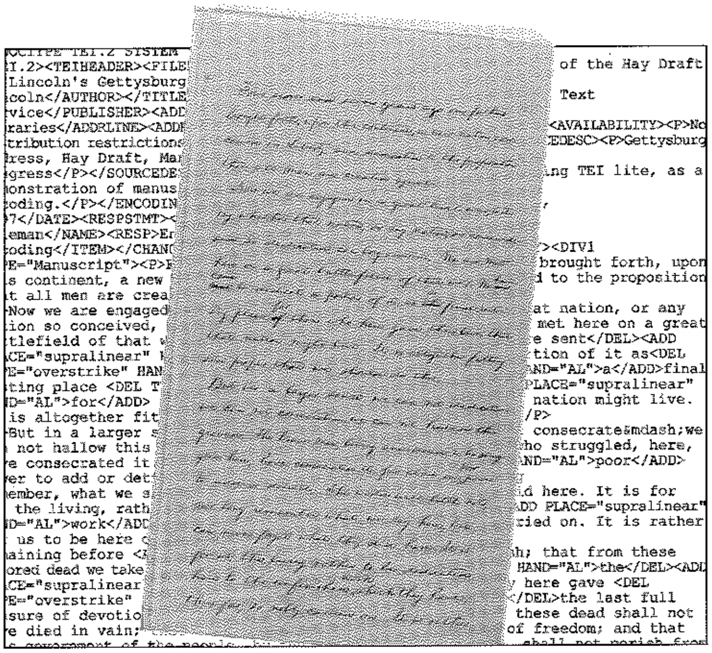
Additional copies are available from the above address for $20.00. Orders must be prepaid, with checks (in U.S. funds) made payable to "The Commission on Preservation and Access."

This report has been submitted to the ERIC Clearinghouse on Information Resources.

∞ The paper in this publication meets the minimum requirements of the American National Standard for Information Sciences-Permanence of Paper for Printed Library Materials ANSI Z39.48-1992.

ISBN 1-887334-51-3

SGML as a Framework for Digital Preservation and Access



by

James Coleman
Head, Academic Computing for the Humanities and Area Studies
Stanford University

and

Don Willis
President, Connectex

July 1997

Commission Preface

In June 1996, the Commission on Preservation and Access began an investigation into the methods and costs involved in using SGML (Standard Generalized Markup Language) for the creation and use of digital documents, with particular attention to preservation and access issues. Technical consultant Don Willis was engaged to conduct research structured to obtain practical results. In addition to reviewing printed and electronic information sources, Willis interviewed SGML users from industry and the library and preservation communities throughout the summer and fall of 1996.

The technical report was then used as the basis for this final report, which is directed primarily to library and archives communities. Among other reviewers, Barclay Ogden, Head, Conservation Department and Digital Library R&D Department, University of California at Berkeley, made substantial contributions to its content and approach. Information in the report has been updated throughout its preparation. However, it necessarily reflects the situation as of the date of publication.

This report is one in a series of Commission publications intended to spark additional discussion and contribute to a collective understanding of how preservation and access needs can be addressed within an evolving technological environment.

About the Authors

James Coleman is the Head of Academic Computing for the Humanities at Stanford University, and runs an SGML-based electronic text center there. He has been involved in developing information retrieval and access systems for research and scholarly use at Stanford University and the Research Libraries Group, Inc. for more than 10 years.

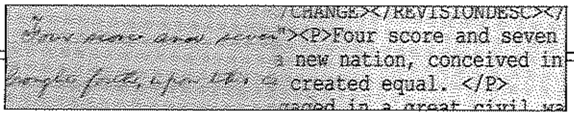
Don Willis is one of the founders of Connectex (www.connectex.com). He has developed information systems for The Library of Congress, Bell & Howell, and Sony. Willis, formerly Vice President, Electronic Product Development, University Microfilm International, is the author of the CPA report *A Hybrid Systems Approach to Preservation of Printed Materials* (November 1992), one of a series of technical reports from the Commission's Technology Assessment Advisory Committee.

Authors' Acknowledgements

The authors would like to acknowledge Rex Lau, Walid Tout, Phil Storey, Marcia Militello, and Mary Laplante for their research and contributions to this report, as well as Maxine Sitts for managing its editing and production.

Table of Contents

APPENDICES


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal. </P>
gaged in a great civil wa

As the nation's cultural resources become increasingly represented in digital form, the need grows to collect, preserve, and provide broad access to those resources in ways that are both efficient and affordable.

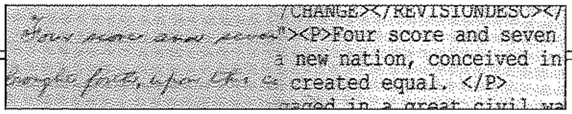
NDLF Planning Task Force, Final Report-June 1996,
http://lcweb.loc.gov/loc/ndlf/plntfrep.html

About This Report

This report explores the suitability of Standard Generalized Markup Language (SGML) as a framework for building, managing, and providing access to digital libraries, with special emphasis on preservation and access issues. SGML is an international standard (ISO 8879) designed to promote text interchange. It is used to define markup languages, which can then encode the logical structure and content of any so-defined document. The connection between SGML and the traditional concerns of preservation and access may not be immediately apparent, but the use of descriptive markup tools such as SGML is crucial to the quality and long-term accessibility of digitized materials.

Beginning with a general exploration of digital formats for preservation and access, the report provides a staged technical tutorial on the features and uses of SGML. The tutorial covers SGML and related standards, SGML Document Type Definitions in current use, and related projects now under development. Looking ahead, the authors describe a tiered metadata model that could incorporate SGML along with other standards to facilitate discovery and retrieval of digital documents. Endnotes and a bibliography provide further resources. Appendices include: a discussion of practical concerns related to the uses of SGML in conversion and authoring projects, descriptions of markup formats and SGML tools, and a vendor's look at cost metrics.

The authors conclude that SGML meets current preservation and access requirements for digital libraries and, furthermore, that SGML solves the widest range of archival database problems today: SGML is a standard; it is non-proprietary and platform-independent; and it allows rich, full access to content-based digital documents. This is not to say that SGML is the perfect solution for all preservation and access needs, nor that it should always be the first choice for all preservation and access decisions. Nevertheless, the authors conclude that the SGML framework should be seriously considered when planning digital library projects.


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal. </P>
gaged in a great civil wa

Background

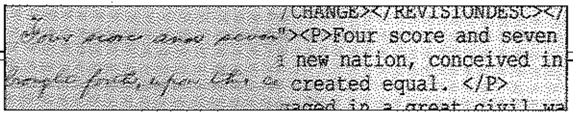
A change in the way that paper was manufactured in the mid-19th century presented preservation challenges for librarians and archivists. A switch from rag content to the more economical and widely-available wood pulp revolutionized book production, creating a medium whose self-destruction due to the acidic content of the "improved" paper became widely known only in the early 20th century. A few decades later, to this challenge was added the preservation needs of film and micrographic materials—materials used for capturing photographic images and for the reformatting of information from embrittled books, but which themselves are unstable and subject to chemical and physical decay.

Just as the preservation and access profession worked with bookbinders, environmental engineers, materials scientists, and micrographics technicians to understand and address these problems, they now must work within the digital environment to assure that digitizing and reformatting projects achieve the same goal that the community has always sought: enhancing the long-term preservation of and access to information of enduring value for as long into the future as possible.

Now, as the 21st century nears, increasingly sophisticated preservation challenges come into focus—challenges that are in many ways markedly different from the past. On one front, preservation professionals are incorporating the use of scanning systems to capture information from previous media (paper and film) in order to create a digital surrogate of the original. The application of these electronic imaging systems for preservation purposes has a number of parallels to and relationships with preservation .microfilming systems.[1]

On yet another front, those concerned with preservation need assurance that the systems and formats they choose can truly capture the required intellectual content of original material and provide useful access to that content now and into the future. The new technologies, exemplified by the SGML framework discussed in this report, not only allow for the digital preservation of a representation of a page, but also include the means and methods to locate and easily retrieve the components that make up that page, even across systems.

Consideration of an SGML framework for digital preservation and access requires a new level of technical understanding. However, here again, the past experiences of the preservation community are proving helpful for today's planning and decision-making.[2]


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal. </P>
gaged in a great civil wa

Exploring Digital Formats for Preservation and Access

Preconditions for Digital Preservation and Access

If the emerging digital library is considered as a virtual collection of distributed resources accessible and available to any authorized user from any authorized location, it becomes clear that those building this collection must employ at least a minimal set of standard protocols to enable the sharing of resources. To be most effective, each system must be able to interact effectively with all other systems, while at the same time meeting the needs of the individual institution. As the final National Digital Library Federation (NDLF) Planning Task Force Report puts it:

> *Digital library collections will comprise components from a variety of heterogeneous systems operated by a variety of independent institutions, including publishers, value-added distributors, scholarly organizations, academic departments and study centers on our campuses, but also (most critically from the standpoint of the Federation) other research libraries. Each of these various institutions will build their digital services based on local priorities and capabilities.*[3]

Successful interactions between producers and consumers of digital resources will depend on standard network protocols, common exchange formats, and common data formats. These protocols and formats are likely to evolve and change altogether over time as current networking, transmission, storage, and application technologies are supplanted by new ones.

Within such a changing environment, librarians and archivists need to be able to clearly delineate the content and nature of exactly what they are attempting to preserve. For example, are they interested in preserving the digital file/system/application itself, that is, a living morgue of dead applications such as WordStar, MultiMate, or Lotus 1-2-3? If they are interested in the "content" of a digital document, what is meant by "content," and how is that content related to use?[4]

In the digital environment, these are crucial questions for helping assure that the formats and systems librarians and archivists choose for preservation and access can mine the intellectual content of the digital object being preserved and, at the same time, permit its delivery, use, and re-use in a variety of current, and future, environments.

For this reason, any system or digital data format that librarians and archivists wish to employ as either users or managers of digital documents will need to meet the following requirements:

- Adequate to the task
- Non-proprietary
- Widespread
- Portable
- Persistent
- Data-rich

Task Adequacy is the most obvious of the criteria, and yet elusive. For example, if a scholar wants to retrieve a copy of the Gettysburg Address using phrases within it, a page image would not be an adequate digital object: a full text in ASCII or UNICODE format would be required. Alternately, if a scholar is interested in comparing samples of Lincoln's hand under various conditions, an ASCII version will not be adequate: here a page image is needed. The issue can become more complicated if the scholar is interested in doing research using images[5]. He or she may wish to focus strictly on reviewing a large number of images: in this case, low-resolution thumbnails may be adequate. But this review may then lead to a need for some higher resolution "service image", to be delivered over, say, a Web page. Or the scholar may want a full copy of the archival resolution[6] to include in a publication (assuming, for the moment, that issues of copyright do not come into play). In the first (thumbnails) instance, the image would be delivered in a GIF format, in the second (Web page) as a JPEG image, and in the last (publication) as a 24 bit uncompressed color TIFF image.[7]

From the point of view of the delivery system, the general requirements of *task adequacy* will help determine the requisite fidelity and format. From the point of view of the digital storage/preservation system, however, task adequacy will most likely require one *data-rich* format that can then be converted to other delivery formats.

Wherever possible, *non-proprietary, widespread* formats should be employed. By now, this should be obvious to most users of computers. Proprietary formats and systems are invariably hardware- and/or device-dependent, difficult to translate, and dependent on the vendor for support. For example, the TIFF format developed by Aldus and the Postscript format developed by Adobe are both 'standards'; while both are widely employed in the delivery of digital documents, only the non-proprietary TIFF format seems to be a good candidate for *de facto* standardization. The danger here is that a digital object, represented in a non-standard format, can lose the ability to be accessed or transmitted from one system to the next. Standards-based formats at least afford the protection of knowing that a digital object conforms in a very specific way to a public description of how the object is supposed to be constructed.

Portability and persistence[8] similarly ensure that the given formats interoperate over a wide range of systems and are at least somewhat immune to continual requirements for change and updating. Although they may seem to stand in opposition to each other, portability and persistence are in fact opposite sides of the same coin: portability—that is a design that enables storage and transfer with no loss of information— is one of the hallmarks of persistence.

Lastly, as mentioned above, librarians and archivists should preserve resources in formats that are *data-rich*. This means that the stored object should contain as much digital information as possible. For example, it is far preferable to capture and store color images as 24-bit uncompressed TIFF files than, say, in Photo CD format. The Photo CD format is not only proprietary, it also changes and removes valuable color information from the original image. Similarly, for page images, an archival resolution uncompressed image file would be preferable to Postscript, since the former retains digital data about the object, whereas Postscript only contains the programming or processing instructions necessary to render the object on a Postscript-compliant device. Those familiar with the differences of rendering the same Word document, for example, as a Postscript file on a Macintosh or on a PC running DOS or Windows will readily understand the problems inherent to files that merely provide machine instructions.

Digital Formats

In most current digital library projects, digitization refers to the production of page images. In some of these projects, the text component of these images may be subsequently rendered into full-text by using OCR programs, rekeying, or a combination of the two. The resulting files can be image files, ASCII text files, or a combination of text and images. For information currently in digital form, the usual instantiation is some form of word-processing file. Occasionally, these word-processing files are likewise distributed in a presentation format such as PDF or Postscript that presents the content in a more published form.

Text Format

ASCII

ASCII—American Standard Code for Information Interchange—is an international standard for transmitting text in a digital form. Defined by the American National Standards Institute (ANSI) in 1968, ASCII has become a *de facto* standard for encoding text data. ASCII assigns numbers, from 0 through 127 (so-called seven-bit, or lower ASCII)[9], to numerals, letters, punctuation marks, symbols, and control codes. ASCII can be easily transferred over networks and can only be displayed as plain, unadorned text.

ASCII is simple, easy to handle, and can be read on nearly every computer using any operating system: qualities that were important when computers had more severe limitations on processing power and that are still important today[10]. The primary disadvantages of ASCII are well known. It cannot render languages other than Western European (and even those not well), and it cannot deal with alternate character sets or writing systems (i.e., right–to–left or bottom–to–top). This makes plain, unadorned ASCII an unacceptable choice to the scholarly and preservation community, which needs to be able to represent and use all of the world's printed languages, both past and present.

UNICODE

One promising replacement for ASCII is UNICODE. The current UNICODE specification (version 2.0) as defined can represent 38,885 characters and includes character sets for most languages, including Chinese, Japanese, Korean, Arabic, and Hebrew. It does this by expanding the length of the character representation from one byte (or eight bits) to two bytes (or sixteen bits), which exponentially increases the number of characters that can be encoded.

UNICODE holds out the promise of a becoming a widespread, standards-based universal character set, and as such is gaining acceptance. It is supported by a consortium of computer makers, software vendors, and national user groups. It is implemented in Windows NT, is the character set for Java, and is available in the new Netscape Communicator browser. It will be supported in the new Macintosh operating system. However, to date, few applications have implemented UNICODE, and it is likely to be a few more years before many software tools will conform to the UNICODE standard.

However, even UNICODE alone is not sufficient for rendering complex textual information pertinent to document structure. This requires some form of descriptive markup, discussed below.

Image Format

In addition to textual, content-based information, images are also a part of the digital repertoire. Indeed, since the earliest days of the digital library, the notion of digitization has meant the creation of page images. In a certain sense, imaging—both monochrome and color—is a least-common-denominator archival format. Imaging can render graphics, photos, special characters, foreign languages, symbols, and formulas more effectively than any other format. Image formats provide a convenient, effective and (using compression techniques) relatively efficient way to store digital surrogates. If managed with sufficient forethought, page images are relatively easy to deliver and use. Page images can provide high-quality copies and can be retrieved successfully with automated aids.

However, projects that rely solely on the digitization of page images have three significant drawbacks: 1) the information represented on these pages is stored as a kind of digital photograph and is therefore unsearchable; 2) the digital photograph contains no machine-intelligible information, requiring conversion to text to expand the usability and value of the content; and 3) the images are almost completely dependent on external systems to manage their relationship to one another and to the paper object of which they are the surrogate.

In its operation, a simple digital image capture (scanning) system is similar to a photographic or micrographic system. The micrographic system has been a mainstay for libraries for preserving intellectual content when the physical object begins to disintegrate. However, as a digital surrogate of the physical object, a page image is not the equal of its micrographic counterpart in terms of fidelity, simplicity of use, and resistance to obsolescence. Page images do not endure beyond the machines that can read them. The advantage of the digital form comes from increased accessibility, retrievability, transmissibility, and greater capacity for automated management.

Both micrographic and digital image formats have their strengths and weaknesses and can be used intelligently to create cost-effective preservation solutions.[11] The questions surrounding the digitization and delivery of micrographic images *qua* images in a digital format are beyond the scope of this paper. Suffice it to say a microform generated within a preservation program should be able to be located and digitized, so that it can be presented to the user electronically as a page image or processed into a more richly encoded version of the original object.

Indeed, the data-rich digital library environment of the future will require electronic access to much of its information, which will most likely be stored as compound documents, that is, documents consisting of ASCII or UNICODE text components, graphical components, monochrome or color images, and perhaps sound or motion picture files, together with a methodology for describing, rendering, and accessing these documents. Librarians and archivists should be interested in creating compound documents, to the extent that projects and budgets can afford it.

Compound Document Format

As noted above, a compound digital document can include a variety of components and features: metadata, text, graphics, special characters, foreign languages, audio, video, and photos. These components can be separate files or formats, they may reside within the same file, or they may be separate components in separate places (for example, as files in a UNIX file system or as records in a database). The composition of the document and its instantiation is a matter of implementation. For example, it can be accomplished by a separate document that records the components, their sequence, their location, and instructions for reconstituting these components.

Thinking back to the original notion of the emerging digital library as a virtual collection of resources from many different institutions, it becomes evident that there may be no best way to store or manage the components of a compound document. What becomes important is having a syntax to describe these documents that allows them to be located across systems and instantiated when needed. The system or server that owns the document (or components of it, since they may not all reside on the same system or server) must be able to render it when presented with a syntactically legitimate request. The digital library arena needs protocols for requesting and delivering compound documents across systems. SGML provides those protocols: it offers the possibility of rendering textual information richly, and it provides syntactic containers for a variety of digital documents.

Compound documents must be structured according to a standard syntax (markup) if they are to be universally accessible. The two most common methods of markup are procedural and descriptive.

Procedural Markup

Most users are familiar with a word processor's ability to embed layout or formatting instructions to determine a document's physical appearance on screen or in print. Although these instructions differ from word processor to word processor, it is possible in almost all full-featured systems today to control such design elements as justification, centering, or font style by using commands or keystrokes to turn these features on or off. This is known as procedural markup. It is usually invisible to the user, and the commands or instructions are embedded in the file as part of the proprietary file structure. Most publishing and word-processing software use this type of markup.

Procedural markup formats (e.g., word-processing formats such as WordPerfect, Microsoft Word, or Framemaker) meet a variety of output processing needs, but they cannot be recommended for archival data storage. Not only are they for the most part proprietary, they also employ file formats that are unable to record the intellectual richness of the digital object. Procedural markup describes the look of a document, but says nothing about its content or structure. Although many document retrieval systems are able to index and read documents that contain procedural markup, the information they contain is almost always ambiguous or erroneous to some degree. The format was simply not designed for this purpose.

For example, if the creator of a digital document has been consistent in the use of document formatting (or styles), then main section headings in the document might be all 14–point Helvetica bold, centered, with 12 points of spacing before and 3 points spacing after. Secondary and tertiary section levels might be distinguished by additional, or different, instructions, and index entries might use encoding that is hidden from the user altogether. Unless an indexing system is able to read and interpret these procedural markup clues, access to these documents would not be possible by main section heading. Neither could this markup be used to generate automatically a table of contents or index.

What is more likely is that the author has not been careful or consistent in using styles and formatting. Certainly, most word-processing systems do not help authors be consistent or prevent them from being inconsistent. This means that even a system that could read this form of markup might be misled through user error. The complexity and structural ambiguity of format-embedded files make automated, intelligent conversion and retrieval difficult, if not impossible. To know more about the document *qua* document, another device is needed, one that separates the presentation of the digital document from the structure of the document. That device is descriptive markup.

Descriptive Markup

Descriptive markup identifies the semantics of a document and its component parts. Descriptive markup can be thought of as information about the document that can be included and embedded within the document itself. Descriptive markup can delineate structure and content. The elements of a document may be easily declared, identified, found, re-used, and presented in many alternate output styles. The approach of descriptive markup when properly executed is independent of any particular vendor product, so that data conversion is no longer required when presentation tools change.

The advantages of descriptive markup are manifold. It permits the creation of electronic documents that can be:

- modularized for reuse
- structured for maximum subject-specific retrieval
- moved and handled in like ways from one operating system to another
- composed of many separate digital components (i.e., text files, image files, external or internal pointers/references)

All of the intelligence that is embedded within the document — document structure, segmentation, separation of document elements by element type (e.g., paragraphs, lists, running heads, and so on) — can be identified and used to great advantage in descriptive markup systems.

However, descriptive markup does not easily carry information about the formatted output. Indeed, under the descriptive scheme, formatting is conceptually separate from the syntax and semantics of the document structure. In descriptive systems, formatting is left to another set of document specifications and to different software tools altogether.

The insight here is that formatting may take a variety of output paths—paper, computer file, CD-ROM, or Web page, for example—and that formatting instructions and filters applicable to the appropriate format path should be used to mold the information content to those separate delivery pathways. For this purpose, procedural markup packages—or better yet, packages that understand and filter descriptive markup—have critical roles in the world of information management.

Presentation Format

PDF - Portable Document Format

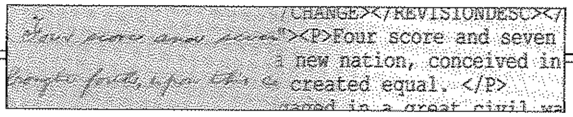
(See also, "PDF" under "Procedural Markup" in Appendix 2)

The Acrobat Portable Document Format (PDF), first introduced by Adobe in 1992, was developed to allow electronically published documents to retain their look and feel independent of computer platform.[12] One of PDF's aims is to create a new document standard to share document information electronically. For example, PDF files created from documents authored in a Windows environment look the same to users on Windows, DOS, Macintosh, or UNIX computers. Specialized software from Adobe takes an electronic file and then "publishes" it as a PDF file, from which it must then be read by using a specialized viewer or Web browser plug-in. This means that both the creation and viewing of PDF files rely on software from a single vendor — Adobe — and on the existence of an electronic file from which to create that PDF file. Until late in 1996, with the advent of another Adobe product, Capture[13], there was no good way to create PDF files from documents that were not already digital, absent rekeying them as word-processing files.

Because PDF is based on Postscript, PDF files give content providers the ability to retain the "look and feel" of a printed document for their data. In a Web environment, this can be important. By their very nature, Web browsers are built to allow a substantial amount of control by the user, such a significant amount of control, in fact, that the overall presentation of the publication can be affected by the choices the user makes. Since a PDF file requires a viewer or a "plug-in" to be read and interpreted, the publisher is guaranteed (at least to the extent that the PDF document creation process is successful) that the document appears very nearly exactly as it has been created. This viewer also can be used to create printed pages that duplicate the original electronic file. In fact, as a device for assuring publishers that "what they publish is what you'll get" and as an aid to facilitate remote printing of an electronic document, PDF is currently without peer.[14]

If PDF succeeds as a way of transmitting a document's look and feel and as a method of enabling remote printing of exact copies of published electronic documents, it fails to meet even minimal standards of portability and data richness. PDF retains no information about document structure, and in this regard is even weaker than HTML. Text within a document cannot easily be searched from remote systems, although there are rudimentary searching capabilities with PDF viewers. PDF files do not contain editable data, nor can a PDF document easily be converted to another format. Although some programs permit libraries of PDF documents to be searched, this technology too is limited and dependent on Adobe.

It does not seem that PDF could be a viable archival format for the digital library. Even for documents originally created electronically, the original document file is almost assuredly more task-adequate, persistent, portable, and data-rich than a PDF file is likely to be, even if this document is descriptively, rather than procedurally, marked up. PDF documents that have been created by scanning using the Capture product with its OCR engine would fare even worse in comparison. This is not to say that PDF does not have its uses as a presentation format. In fact, PDF could easily be one of a set of delivery and presentation tools, even in an archival setting.


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal. </P>

SGML and Related Standards

SGML

Standard Generalized Markup Language (SGML) is an international standard (ISO 8879), first published in 1986, designed to promote text interchange. SGML is used to define markup languages, which can then encode the logical structure and content of any so-defined document. SGML is concerned only with the formal properties and interrelationships of a document, not with the semantics of markup itself or formatting. According to Severson, "By focusing on the structure and intent of the document, rather than proprietary formatting codes, it [SGML] maximizes accessibility and reusability of the underlying information. SGML lets us find information more effectively because we can use internal document structure to help guide our search. It lets us reuse the information more effectively because SGML documents are not tied to any particular format, application, or publishing platform" (Severson 1995).

SGML accomplishes these goals through document type definitions (DTDs). A DTD is a markup language composed in SGML syntax as a collection of rules that name the objects or components of a document and then describe their interrelationships. An SGML DTD specifies what components are allowed, what components are required, how the markup itself is to be distinguished from text, and what the markup means.

Three concepts are fundamental to an understanding of SGML languages: the notion of an "entity," the notion of an "element" (or tag) with its attributes, and the notion of a document type. As Lou Bernard puts it:

> *At the most primitive level, texts are composed simply of streams of symbols (characters or bytes of data, marks on a page, graphics, etc.): these are known as "entities" in SGML. At a higher level of abstraction, a text is composed of representations of objects of various kinds, linguistically or functionally defined. Such objects do not appear randomly within a text...they may be included within each other, linked to each other by reference or simply presented sequentially...This level of description sees texts as composed of structurally defined objects, known as "elements" in SGML. The grammar defining how elements may be legally combined in a particular class of texts is known as a "document type."*[15]

These notions — and particularly that of "entities" (defined in the following section on Features of SGML/Reusability) — form the building blocks for thinking more deeply about compound documents in digital libraries.

The DTD is thus actually a set of structural rules that can be parsed (or interpreted) by SGML-compliant computer applications. Electronic documents that use a given DTD can likewise be said to follow or fail to follow the rules of any given DTD. This process of testing whether a document conforms to a DTD is called validating, and the results of the process are a valid instance (or document) of a particular DTD. In marked distinction to the software used in the creation of procedurally marked up documents, SGML-aware software can assist the document producer in creating a document that is consistent and unambiguous in its markup.

The value of an SGML DTD as an interchange format becomes clear: it provides a definition of document structure that can be known by any SGML-compliant remote system. A document that is valid according to that DTD then can be processed by any SGML-compliant remote system when presented with the document and the DTD. Because SGML documents are not tied to any particular word-processing format, application, or publishing platform, flexibility and mobility are preserved from the point of view of the document. Because SGML is machine-processible, the same flexibility and mobility are preserved from the point of view of the processing application.

A number of SGML DTDs have emerged as the descriptive markup languages best suited to handle encoding requirements for hierarchically structured digital collections. It also has been proposed that SGML could be used to encode bibliographic data as well and provide thereby a single access and exchange platform not only for compound digital documents, but for the bibliographic information associated with them. An initiative to map the current MARC bibliographic formatting to an SGML model is discussed later.

Special Characters

In general, and without particular specification, an SGML document is rendered in ASCII.[16] Support for the addition of extra characters comes through the use of character entities. Any character not in the ASCII character set can be defined in a string that begins with a "&" and ends with a ";". As one example, the markup string for the Greek alpha character (α) is &alpha;. The SGML-aware output processor converts the strings to display the proper symbols on the output device (e.g., a screen, paper, CD-ROM). Many standard sets of characters, including UNICODE, are already defined and registered, and users can define their own. This markup scheme solves the problem of supporting additional characters, but it adds a level of indirection to an SGML conversion effort. SGML is not tied to ASCII, however. An SGML DTD can be defined to work with any character set, including UNICODE.

Since SGML separates content and structure from presentation, the same information can be reused for any delivery mechanism, be it print-based or electronic. How SGML data are presented is largely dependent on the SGML-aware application that processes the data. This approach has distinct advantages and one large disadvantage. On the one hand, a user can develop separate style sheets of formatting characteristics for paper presentation or for electronic presentation and thereby use one data steam for multiple output formats or products. These products then can be molded or directed toward a variety of users for a variety of uses. On the other hand, this approach requires that a style sheet (or some other form of formatting instructions) be developed for every output format or purpose, and usually also for every SGML-aware processing program, since there is no common method or syntax for creating these style sheets.

For many users and applications, the need to filter SGML for every use can entail significant costs in addition to the resources expended for markup. This is a particularly significant problem for those who are creating new, ongoing digital publications — a problem addressed by DSSSL.

Document Style Semantics and Specification Language (DSSSL)

About the same time that SGML received approval as a standard, the working group that created the SGML standard began work on the issues of document style sheets and formatting. Out of the work of this group came Document Style Semantics and Specification Language (DSSSL), which became an international standard in April 1996. What does DSSSL do?

According to Dianne Kennedy, founder of the SGML Resource Center,

> *DSSSL provides standardized syntax and standardized semantics to specify style and layout for SGML documents or SGML document fragments. Using DSSSL we can specify the style for an element or attribute, interchange that style specification, and reproduce that style within certain limitations.*
>
> *DSSSL, like SGML, is declarative in nature, and has no biases toward national language, writing direction, or character set. All style specifications in DSSSL are made by describing final formatting results, not by describing the algorithms to be used to create the formatting result. In other words, DSSSL enables us to specify how data appear in output (print or electronic) but not how to create that format.*[17]

Although DSSSL is quite new, there are DSSSL processing engines, DSSSL style sheets for the "lite" version of the TEI DTD and HTML 3.2, and scheme interpreters written in Java. All in all, DSSSL offers significant promise for defining presentation in a way that is as application-neutral and platform-independent as SGML.

Features of SGML

SGML As an Exchange and Storage Format

The main components of a digital library will come from a variety of largely heterogeneous systems operated by a variety of independent institutions. In such an environment, it is necessary to have consistent methods of transmitting data from one institution or system to another and to transmit data consistently in formats that are understood in the same way by all systems. What is needed are common exchange/interchange formats and common network protocols that can be used and understood by all participants. For bibliographic data, the USMARC format fills this role for libraries. The nation's emerging digital libraries, however, will need not only to manage and share the bibliographic information across common exchange/interchange paths, but also to manage a wider matrix of metadata than is captured by the USMARC record and devise access protocols to deliver the heterogeneous digital objects to which the metadata refers.

For example, for any compound digital document, the digital library of the future might need to manage and deliver metadata about:

- the compound digital document itself (e.g. creator, date/time/place of creation, issuing agency, type of document, subject access points)
- the digital objects associated with that compound document
- the location of those digital objects
- the methods of navigating and/or retrieving components of that compound digital document
- the rights and use management information associated with that compound digital document

Intersystem access protocols might need to:

- query this metadata, using some standard protocol
- manage user authorization
- deliver on request from an authorized user additional information about the compound document and/or related documents
- deliver the document in a particular way/format/segment

How can SGML help address these issues?

Three general SGML features — modularization, extended document reference, and encapsulation — used in combination and with community- and standards-based DTDs offer an integrative path for the digital library.

Within the SGML framework, the creation of a compound document is relatively simple. The SGML document type definition can simply state that a document consists of other files (or documents), and then declare and name these files (or documents).

The same principle can be used to construct whole databases out of SGML documents with more complicated DTDs by making the DTD itself an element or, in the following example, by exchanging the DTD of choice with the reference to TEXTFILE. In this way the inherent modularity of SGML can be used to build out and transmit databases or data documents as needed.

Example

To build a document that consists of a concatenation of files, one could create the following DTD:

```
1.  <!ELEMENT METADOC - - (TEXTFILE+)>
2.  <!ELEMENT TEXTFILE - - #PCDATA>
```

Where line 1 defines the document type, **METADOC**, as consisting of one or more elements **TEXTFILE**; line 2 states that element **TEXTFILE**, consists of character data, i.e., an ASCII file.

From this DTD, one could then create an SGML document that concatenates any number of files, in any chosen order, by declaring entities. For example:

```
<!DOCTYPE METADOC SYSTEM "METADOC.DTD">
<!ENTITY DATA1 SYSTEM "<OSFILE>/DOC1">
<!ENTITY DATA2 SYSTEM "<OSFILE>/DOC2">
<TEXTFILE>
&DATA1;
</TEXTFILE>
<TEXTFILE>
&DATA2;
</TEXTFILE>
```

The `<!DOCTYPE` statement names the DTD that the current document conforms to and describes its location. The `<!ENTITY>` statement declares each file to be referenced as being a file on a system following a specific path, and the **&...;** construct completes and instantiates the actual reference to these files. When executed by an SGML-aware system, this "document" would result in the concatenated output of DOC1 and DOC2.

Similarly, pointer facilities exist within SGML to reference other documents, elements within documents, documents on remote systems, and elements within documents on remote systems. This referencing ability is far more widely developed within SGML than is exercised by the HTML DTD, whose links are all of the one-to-one variety (or many-to-one). SGML generally provides support for one-to-many links, and this facility is widely used within the DTDs discussed in the next section.

Thus, SGML offers a generalized set of facilities to meet the challenges posed earlier.[18] By using DTDs and valid SGML documents that follow them, it is possible to exchange digital information between different systems so that they can use the data and understand its syntax. As explored later in this report, by using a small set of standard, community-based DTDs, the digital library community begins to build an infrastructure that makes possible interoperability between these heterogeneous systems on the basis of DTDs, MARC bibliographic records, the Dublin Core elements, and a standard communication protocol.

Reusability

The creation of knowledge, as embodied in text, is difficult, time-consuming, and expensive. Too often, though, knowledge has been created once and then discarded — because of a new incompatible word processor, because there was no way to re-use a text module in more than one place, or even because there was no notion of "text modules" or resuability. These are the problems that SGML was created to solve. SGML's design supports re-usable modules, called "entities," built in at its core.

If entities are defined as data streams, one can then logically assume that these data streams can essentially be of any length or type; they can represent a single character (for example, the German ü), a phrase, a paragraph, an entire text file, a graphic, or even another SGML file or set of files. This means that SGML entity references can be used to create the logical components of a compound document, where the main SGML file acts as a container (or driver file) for these digital components. When used for special characters (such as that ü), they may then also be translated automatically into the character/font for the system delivering and rendering that character.[19]

Also, because SGML is vendor-neutral, compliant SGML products should be able to process any SGML document.[20] SGML as a meta-language makes it possible to develop your own generalized markup, but with a standard form and in standard ways. SGML makes it relatively easy to handle large and complex documents and to manage large information repositories.[21]

Retrievability

The advantages that SGML offers for intelligent documents also are extremely valuable. For example, SGML-aware document indexing and delivery systems can use the intelligence of a DTD to:

- search the document by SGML elements (e.g., title/author fields, first lines of poems, text contained in footnotes)
- generate tables of contents automatically using SGML document components
- create indexes/glossaries of specialized terms (e.g., foreign languages, name authority lists, figures/graphs in a document)
- allow the navigation of large document databases in meaningful chunks (e.g., by speeches/scenes/acts in play, by line/stanza in poems, by sentence/paragraph in works of prose, or by other, domain-specific methods)
- allow the logical presentation and navigation of documents with greater intelligence and at a finer level of detail or granularity

In current SGML indexing and retrieval systems, such as those offered by OpenText or INSO Corporation's DynaText, searches can be restricted to a glossary, a title, or a footnote—essentially any structural element in the document. Moreover, since SGML embeds hierarchical structure, a search may be conducted for elements only when these elements are contained by other elements or have particular qualities (for example, a term in a list, but only when that list is of a particular type). The potential for thereby dramatically increasing the accuracy and efficiency of the search is greatly enhanced. This structure also can be used in conjunction with database management systems for document storage and management.

The retrieval, analysis, and logical presentation capabilities of documents encoded in SGML are essentially limited only by three factors: the capabilities of the current generation of tools, the intelligence and richness of the DTD, and the care with which it has been used by the author or encoder.

HTML - HyperText Markup Language

The most widely used DTD in the world is HyperText Markup Language (HTML), which is used for the markup of documents on the World Wide Web. HTML is an application of SGML. In distinction to other DTDs of importance for digital libraries, HTML originally was created for a relatively limited purpose: to communicate information an share documents among a small group of research physicists. This meant that the element (or tag) set would be small, containing only those elements that scientists needed: headings, lists, links, and some rudimentary formatting. Since the explosion of interest in the Web, the development and extension of the HTML has focused almost exclusively on on-screen presentation rather than document content. As noted before, the HTML focus on presentation is rather, if not completely, at odds with the general objectives of SGML. For HTML, these goals were necessary so that tools for displaying and manipulating HTML would be easy to implement. The HTML 3.0 standard has fewer than 50 elements, while a typical DTD can have hundreds, and most HTML tags, as noted, are used specifically for formatting.

Why HTML came to be constructed as it is now is easy to understand. Its implementers were eager to use the features of SGML as building blocks for HTML, but they did not want to repeat the mistakes of the military CALS[22] community or other implementers of classic "technical documentation" applications. HTML had to be kept simple; the historical evolution of CALS was a model to be avoided at all costs.

That is not to say that HTML has remained static. It has changed over time to reflect the needs of the Web community, as is evidenced in this list of the major versions of HTML:

HTML 1.0 DTD	very basic, no forms support
HTML 2.0 DTD	forms support
HTML 3.0 DTD	expired draft, not widely implemented
HTML 3.2 DTD	latest HTML version, tables, backgrounds, colors
Netscape DTD	additional extensions (color, frames)
Microsoft DTD	different extensions from Netscape

Most other vendors also have implemented a few special tags or attributes, and makers of each browser naturally maximize the browser for their own DTD and ignore tags or attributes associated with the other DTDs. Although ignoring tags not understood is correct behavior — this is the only way to ensure some level of compatibility between browsers — it is also easy to imagine the day when HTML is not so much a standard as the proprietary "flavor" of the browser of the month

Despite these changes to HTML, current versions are nowhere close to being robust enough for an archival format. Its simple markup is its major weakness. HTML does not allow for complex document structuring. Most documents in HTML contain little markup beyond designating paragraphs, titles, graphics, and HyperText links. Therefore, documents created on the Web using HTML are devoid of much of the structure that supplies intelligence to today's document collections. Worse yet, the HTML files that exist on the Web today are not likely to conform to any HTML version at all. After all, most casual — and even not so casual — Web page creators are more interested in achieving a certain look by the creative use of tags than in communicating content or structure. This practice renders almost any attempt at a structured encoding of content within HTML useless, and it is one of the more unfortunate facts confronting those who are interested in preserving Web content.

Under the current circumstances, the usefulness of HTML as anything but a delivery/presentation mechanism for the emerging digital library is seriously questionable. Eric Severson frames the fundamental question in this way: "Should HTML be viewed as a general interchange format, or is it just a lowest common denominator useful primarily for casual applications? Could it be used as the archival format, or should HTML always be produced from some other source, such as an SGML-based information repository?" (Severson 1995). The latter seems more appropriate.

Severson (1995) points out that HTML is not rich enough to be a mainstream repository format. He further posits that "SGML in general (i.e., one's own DTD) is the right way to build a corporate information repository; HTML is a specific vehicle to put that information on the Web. Yet, if HTML is not meant to be reasonably rich, then a significant amount of its appeal may be lost. That is, rather than being a universal solvent for information being passed across the Web, it becomes just another intermediate format driving a particular set of browsing software."

Severson is certainly right to voice these concerns. From a digital preservation perspective, HTML cannot be seen as a preservation or archive format. However, HTML will continue to have a presence as a presentation format. The creators of digital libraries will want to use the richness afforded by standards- and community-based DTDs to capture, record, and preserve information, and then to deliver it to their users in the most convenient and appropriate way. For the next few years, HTML over Web browsers is likely to be the delivery path of choice. Beyond that period, work is in progress on a new standard, Extensible Markup Language (XML), which offers the promise of bridging the "intelligence gap" between mature DTDs and HTML.

XML - Extensible Markup Language

This investigation has presented the advantages of SGML when used in conjunction with SGML-aware software and applications. Full-blown support of SGML, however, is not a trivial matter from the point of view of the software provider. An SGML application must be able, among other things, to:

- parse a DTD
- understand SGML declarations
- handle tag minimization
- handle concrete content models
- handle all facets of SGML modularization, entities, public identifiers, and so on

The full list of features for SGML compliance is quite long, and the number of products on the marketplace that support these, while growing, is smaller than the number of word processors, by at least an order of magnitude.

As might be clear from the above explanations, the amount of processing and filtering necessary for presenting SGML documents can likewise be considerable. For example, Panorama, an SGML viewer, when presented with an SGML document, requires at least three additional files to parse, understand, and present the document: the DTD, a catalog file to map entities, and a style sheet for presentation. Users of Panorama also need experience in configuring the behavior of the viewer to use its capabilities successfully.

An ideal environment would enable librarians and archivists to more fully exploit the capabilities of rich DTDs with software that is as easy and convenient from the user's point of view as a Web browser. This is the bridge between the Web and full-blown SGML that Extensible Markup Language (XML) hopes to become.

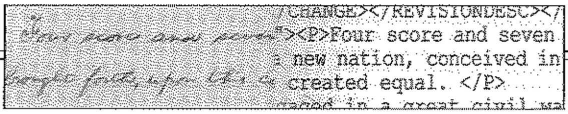
Extensible Markup Language (XML) is a much simpler dialect of SGML. XML is itself a meta-language standard on par with SGML. According to the current working draft, the goal of XML is. . .

> *. . .to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML has been designed for ease of implementation and for interoperability with both SGML and HTML.*[23]

As stated in that draft, the primary design goals are:

1. XML shall be straightforwardly usable over the Internet.
2. XML shall support a wide variety of applications.
3. XML shall be compatible with SGML.
4. It shall be easy to write programs which process XML documents.
5. The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
6. XML documents should be human-legible and reasonably clear.
7. The XML design should be prepared quickly.
8. The design of XML shall be formal and concise.
9. XML documents shall be easy to create.
10. Terseness in XML markup is of minimal importance.

Currently the work on XML is being championed by the World Wide Web Consortium (W3C). Other supporters of XML are Digital, Hewlett-Packard, IBM, JavaSoft, Microsoft, Novell, Spyglass, and Sun. It is also notable that Netscape recently joined the committee that is drafting the specification for XML[24]. Although it is too early at this date to be sure of a positive outcome, the digital library community should follow the development of XML with interest. XML is a standard that moves users one step closer to a "scalable HTML," and at the very least will allow significantly increased formatting flexibility for documents delivered across the Web from an SGML database.


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal.. </P>

SGML Document Type Definitions (DTDs)

The use of SGML as a basis for digital libraries and databases is purely theoretical without standard, community-based document type definitions (DTDs) for the creation and transmission of compound digital documents. Given the level of complexity in creating digital documents, the varying requirements for access and retrieval placed on digital libraries by domain-specific user communities, the need to create a critical mass of digital materials within any specific domain, and the likelihood that members of that domain will be creators and not just consumers of these digital materials, achieving some form of consensus, or at least general acceptance, among the members of user communities is critical. What the library and archival communities have gained in terms of information description and object retrieval through the introduction and development of the USMARC standard for bibliographic records still remains to be accomplished for digital documents.

Most Commonly Used DTDs

This section examines three important DTDs currently in use—the Text Encoding Initiative (TEI) DTD, the Encoded Archival Description (EAD) DTD, and the Consortium for the Computer Interchange of Museum Information (CIMI) DTD. Each has been developed with the creation of digital libraries in mind, and each can carry not only compound digital documents but also metadata relating to those documents.

Text Encoding Initiative (TEI)

The Text Encoding Initiative (TEI) grew out of the recognition that there was, in the words of the participants in the Poughkeepsie Planning Conference, "a pressing need for a common text encoding scheme researchers could use in creating electronic texts, to replace the existing system in which every text provider and every software developer had to invent and support their own scheme. . . ."[25] The sponsoring organizations for the project were the Association for Computers and the Humanities (ACH), the Association for Computational Linguistics (ACL), and the Association for Literary and Linguistic Computing (ALLC). The initiative received support from the National Endowment for the Humanities, and later from the Commission of the European Communities and the Andrew W. Mellon Foundation. Following a six-year international effort that drew in more than 100 scholars and specialists in more than 15 different working groups, the TEI Guidelines were published in 1994 (Sperberg-McQueen, 1994).

The TEI is managed by a Steering Committee consisting of two representatives from each of the sponsoring organizations. Fifteen scholarly organizations are represented on the project's Advisory Board, which approved the plan of work and endorsed the published Guidelines. Two editors, one European and one North American, coordinate the work and are responsible for overseeing the TEI Guidelines.

The TEI specification includes a document header—the TEI Header—that contains a description of the electronic file, a standard set of bibliographic elements for the source document, and additional non-bibliographic information, such as a description of markup practice and revision descriptions. The primary object is to describe an encoded work so that the text itself, its source, its encoding, and its revisions are all thoroughly documented. A MARC record may be linked to a TEI document to augment data already in the TEI header, but it is not intended to substitute for the TEI header. The header can either be created as part of a document's original encoding, or it can be created to describe existing documents and thereby used for bibliographic control. The TEI Guidelines provide a framework for storing headers independent of the text they describe, and they include suggestions for mapping specific TEI elements to MARC tags.

As a DTD, TEI has a daunting richness: the specification runs to two very large volumes, with more than 300 individual tags. But the DTD has been designed to be modular and extensible, so that the TEI is very much a kind of 'DTD Erector Set' for encoding texts in the humanities. This is accomplished by layering the tag sets. The DTD defines a base tag set that is present for all 'flavors' of the DTD. To this, one can add tag sets that are specific to any particular project. There are tag sets for prose, verse, drama, dictionaries, critical apparatus, and linguistic corpora. In addition, it is possible to enable or disable tags that will not be used within each tag set declaration.

TEI developers have planned for extensions: the editors devote an entire chapter to the operative rules for modifying the DTD, and another to rules for interchange. Probably the full TEI DTD is rarely used in projects. The demands of the user community for a more compact subset of the TEI DTD, one that did not require the rather specialized knowledge of SGML needed to modify the DTD, led the authors to create a 'TEI Lite' version of the DTD. This version contains the more useful features from a number of the tag sets, and retains compatibility with the full version. This version is widely used in electronic text archives, including those at the University of Virginia, the University of Michigan, and Stanford University.

Overall, the TEI has found a good deal of resonance in the humanities computing community. The TEI applications page (http://www-tei.uic.edu/orgs/tei/app/index.html) lists more than 50 large-scale projects that use the TEI in their applications, and the likelihood is that many more are using the TEI informally or in smaller projects.

This is not to say that the TEI has found complete acceptance or agreement. As with all living and vital documents, it generates heated discussion. Some people believe it is too complicated and expensive, some that its flexibility makes it interoperable only at the highest levels, and some that it is insufficiently detailed to meet the demands of their particular disciplines. The organizing committee has established working groups to deal with some of these criticisms. Nevertheless, it is safe to say that the TEI DTD remains the standard against which humanities-based encoding projects must measure their work, and that active consideration of the suitability of the TEI is a prerequisite of any such project.

Encoded Archival Description (EAD)

Over a three-year period ending in 1995, the Library at the University of California at Berkeley developed an encoding standard for electronic versions of archival finding aids. The project was inspired by a recognition that archival repositories wished to expand and enhance network access to information about their holdings beyond that available in MARC catalog records. The requirements for the DTD included the ability to describe hierarchical relationships between source materials and to navigate, index, and retrieve information from within this layered architecture. Investigators, led by Daniel Pitti, evaluated tagging in HTML, SGML, and MARC formats. SGML was chosen as the standard because it met all the functional requirements of finding aids and was supported by a large number of platform-independent software products.

The Berkeley group undertook the development of a document type definition that resulted in the original Berkeley Finding Aid Project (BFAP) DTD. Subsequent work by Pitti and others led to increasing refinements of the model, which resulted in the creation of the first version of the Encoded Archival Description (EAD) in 1995. At the Society of American Archivists (SAA) annual meeting in Washington, DC, in early September 1995, the Committee on Archival Information Exchange (CAIE) was asked to become formally involved in the ongoing development of EAD. By July 1996, the EAD had been released in a beta version, and the Library of Congress Network Development/MARC Standards Office had agreed to serve the maintenance agency for the EAD[26].

The EAD data model includes a header record fashioned after the TEI header. Many TEI naming conventions and tagging structures are utilized. The EAD standard also provides for the use of a MARC equivalency attribute for finding-aid elements matching USMARC fields. The EAD is fairly tightly bound to encoding findings aids, which are themselves used to describe, control, and organize access to other information. The finding aid is not an end in itself or an object of study, but rather a tool for locating the objects of study. The EAD has been designed to support existing finding aids and inventories and to permit an encoding scheme that lets the user specify increasing levels of depth or complexity of encoding, depending on the collection itself and the available resources.

The archival community is particularly interested in the EAD. Organizations including SAA and the Research Libraries Group have been active in its promotion and in providing training for its use. Within the digital library, the EAD can find a place as an aid to locating print and traditional archival collections, and it also shows promise in modeling digital collections themselves — that is, collections of digital materials with no other "archival" existence that are created specifically for Web or networked delivery. These new digital archives are likely to be created by librarians, scholars, or consortia and represent for the first time distributed archival digital resources. The EAD can provide the enabling metadata structures for the organization, discovery, and retrieval of these collections.

Consortium for the Computer Interchange of Museum Information (CIMI)

The Consortium for the Computer Interchange of Museum Information (CIMI) is a consortium of museums, museum bodies, academic institutions, and bibliographic utilities that have come together to create and exchange information about museum collections and objects within these collections.

CIMI has seen its work as the creation of standards to advance the technical framework within which such an exchange can take place and the practical application of these standards through demonstration projects. An initial technical document, *Standards Framework for the Computer Interchange of Museum Information,*[27] resulted from the work of internationally based committees between 1990 and 1993. In that document, CIMI adopted SGML as its interchange format. Since then, CIMI has used SGML in an experimental project, named CHIO (Cultural Heritage Information Online), to test the viability of the *Standards Framework.* Project CHIO combines heterogeneous information (exhibition catalogs, object records, bibliographic references, authority files, and so on) about folk art.

In Project CHIO, CIMI developed a domain-specific DTD following the generic TEI framework, specifically using the approach developed in the creation of the TEI Lite DTD. According to Richard Light, the author of the CIMI DTD,

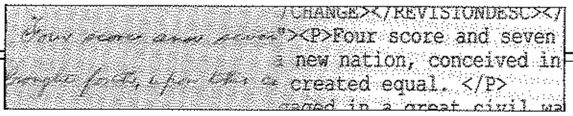
> *We agreed to start, not from the TEI Lite DTD itself (which is fixed), but from the modifications file used to generate TEI Lite. We then modified the modifications (!) to suit our own needs, removing tags that are not required and adding back in 'standard' TEI tags that we wanted. Finally, we added a set of additional tags to express the access points required for Project CHIO.*[28]

Since the CIMI DTD follows the TEI, it uses the standard TEI header to include metadata about each document. Particular emphasis is placed on bibliographic information and on access information and conditions (copyright statements, credits, and so forth). Other features of the TEI, such as its ability to point into other documents and manage links and sets of links within the same document, are likewise retained.

A principal aim of project CHIO is online access to relevant parts of a document in response to user queries. This "access centeredness" leads to some interesting design decisions related to the scope of the query and the DTD itself. In any SGML retrieval system, the "what" of retrieval is important. What constitutes a successful hit to a search? What syntactic unit should be fetched in response to such a hit? A distinction needs to be made between an item whose primary focus is on a subject (like African masks), and a passing reference to African masks. Systems that rely solely on string searches for "African masks" will always confound the mere mention of a term with its development. In the CHIO DTD this distinction is carried out and refined through a separation of primary and secondary access points.

Once the access points have been determined, it is then necessary to determine what chunk of text is related to these access points. The CIMI DTD provides a mechanism for marking specific text chunks with the relevant access-point values. This permits the targeted retrieval of the correct text chunk. This mechanism can be further extended within a museum setting by embedding these chunks within a particular context. That is, the access point for a text chunk might be "African masks" embedded within the context of "museum labels," thus permitting primary access concepts to be qualified more exactly.[29]

According to the Web page describing Project CHIO, "CIMI expects to see a community-endorsed system of encoding museum information using SGML; a methodology for searching texts and collections data using Z39.50; and a demonstration system that will show the power of a standards-based approach to electronic interchange."[30] Currently, the use of the CIMI DTD is limited to CIMI participants. However, the importance of the CIMI work for information that is from finding aids and bibliographic records cannot be overstated. It is likely that the work done by CIMI will significantly improve understanding of difficulties in achieving online access to cultural heritage information held in multiple databases at heterogeneous sites.


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal. </P>

Extending the Use of SGML and Its DTDs

This section first considers the efforts to reconceive the MARC record as an SGML document, and then describes the work to develop metadata standards for Web resources—the Dublin Core and Warwick Framework.

Developing a USMARC DTD

The Library of Congress, recognizing the need for conversion between MARC and SGML data structures, established a 10-member committee in November 1995 to develop an SGML DTD for USMARC. The project framework, as described in the comments section of the DTD, was motivated by a need for non-proprietary, user-friendly utilities to convert from MARC to SGML and back (Library of Congress Network Development and MARC Standards Office, 1994, 1995, 1996).

An alpha version released in July 1996 is available for anonymous FTP at ftp://ftp.loc.gov/pub/marcdtd. The DTDs map SGML data structures to MARC data structures. Two DTD fragments support five USMARC formats. The DTD for Bibliographic Data contains element definitions for the Holdings Data and the Community Info formats. A second DTD for the USMARC format for Authority Data contains mapping to Classifications elements.

The mapping is reversible without loss of intellectual content or bibliographic information. The DTDs are termed "enabling" rather than enforcing, that is, they are not designed to enforce specific subsets of the MARC record, nor do they ensure strict adherence to MARC rules. Just as the enforcement of syntactic MARC validity is left to cataloging application, similar rules would need to be controlled by site-specific authoring and conversion applications, not through the SGML parser and DTD. Only the numerical order of the MARC fields and sub-elements is strictly enforced in the DTD. All fields except the Leader are optional.

According to Randall Barry of the Library of Congress Network Development and MARC Standards Office, the project is slightly behind schedule due to funding delays. The next phase of funding was expected to be approved in 1997, at which time utilities will be developed to automate the conversion process. At present, the library is not tracking the use of the alpha USMARC DTD, and the project is seen as a research-and-development effort.

The library expects that the growth of the Internet, the identification and use of additional metadata required by digital libraries, and attempts to integrate metadata elements are likely to spur this effort.

Bringing Together Digital Libraries and the Networked Environment

The two largest and most important access and discovery tools for information about the contents of digital libraries are 1) the cataloging records found in national utilities and in local systems; and 2) search engines capable of locating shared or "visible" data objects, principally Web-based search engines like Alta Vista and Excite. The former are marked by rigorous attention to the structure, format, and content designation of the bibliographic information. The latter are characterized by very loose use and understanding of the meta-information and an abundance — essentially a one-to-one correlation — of records referring to the shared or visible data objects.

The recognition by the digital library community that these two important discovery realms were moving in unrelated, often opposite, directions led to an invitational workshop in March 1995 organized by OCLC and the National Center for Supercomputing Applications (NCSA) in Dublin, OH, to "address and advance the state of the art in the development and extension of methods, standards, and protocols to facilitate the description, organization, discovery, and access of network information resources."[31] The goal of this metadata workshop was the development of consensus concerning network resource description across a broad spectrum of stakeholders from the worlds of computer science, text markup, and libraries.

Dublin Core

Dublin Core refers to a set of metadata elements used to describe networked resources that were identified at this workshop. The objective was to build consensus for a simple record format that would satisfy two requirements. The first was to define a set of metadata elements that could support precise descriptions for specialized information contained in document-like objects,[32] as USMARC does, and yet be understood easily and generated by the creators of digital objects without specialized and costly training. The second was to provide an extensible framework to describe items other than document-like objects. The result is a model comprising 13 core elements.

The thirteen elements are grouped into three general categories: Bibliographic Access Points (Title, Subject, Author, OtherAgent, Identifier); Identification Information (Publisher, Date, ObjectType, Language, Form, Coverage); and Relationship to Other Objects (Relation, Source). The intent is that the core set can be mapped to any syntax, including USMARC and SGML.

Workshop participants also enumerated a number of principles that govern the application of core: intrinsicality, extensibility, syntax-independence, optionality, repeatability, and modifiability. Intrinsicality refers to the intrinsic qualities of the object; extensibility refers to the element-set itself and provides for its development and extension; syntax-independence acknowledges that at this early state no syntactic bindings should be presumed; repeatability ensures sufficient flexibility to describe the object at hand; and modifiability recognizes that different communities may wish to modify the element name to more adequately describe the object.

A subsequent workshop was convened in September 1996 to test the suitability of the Dublin Core for the description of visual resources such as photographs, slides, and image files. This workshop found that the Dublin Core, with the addition of two additional elements—Description, for information about the visual content of an image, and Rights, for information about rights-management—could be used for the discovery of images and their subsequent retrieval, depending on whether rights management information permitted such retrieval.

The Library of Congress has proposed an SGML syntax for Dublin Core records, to be used as an alternative to the MARC catalog records, for submission of source materials to the National Digital Library Project. An SGML Document Type Definition (DTD) template is under development for the Dublin Core elements, and SoftQuad, Inc., has announced plans to support the DTD in a future software release.

Warwick Framework

The Warwick Framework arose from a second metadata workshop, held in 1996 in Warwick, England. The goals of this workshop were to broaden the international scope of the Dublin Core initiative and to refine the original metadata structure. The Warwick conference resulted in a proposed syntax for the Dublin Core model, the development of application guidelines, and a framework for a "container architecture." This container architecture framework is of critical importance.

Essentially, container architecture provides a method for aggregating logically distinct packages of metadata. According to Lagoze, Lynch, and Daniel, this framework has the following characteristics:[33]

- It allows the designers of individual metadata sets to focus on their specific requirements and to work within their specific areas of expertise, without concerns for generalization to ultimately unbounded scope.
- It allows the syntax of metadata sets to vary in conformance with semantic requirements, community practices, and functional (processing) requirements for the kind of metadata in question.
- It distributes management of and responsibility for specific metadata sets among their respective "communities of expertise"
- It promotes interoperability and extensibility by allowing tools and agents to access and manipulate individual packages selectively and to ignore others.
- It permits access to different metadata sets related to the same object to be controlled separately.
- It flexibly accommodates future metadata sets by not requiring changes to existing sets or the programs that make use of them.

This architecture makes it possible for users in various knowledge domains to transmit and exchange metadata packages that may be of use or interest only to them. It also describes an architecture in which the Dublin Core itself is but one of potentially many packages of information. For example, these metadata packages might contain information about the terms and conditions associated with the use of a digital object; they might contain content ratings information for filtering programs; they might contain provenance information associated with the digital object; they might contain structural information about the document (such as a DTD). The issues surrounding this framework and its possible implementation go beyond the scope of this report. However, it is clear that the Warwick framework architecture offers a basis for metadata extension, interchange, and communication that will be of increasing importance.

Considering a Tiered Metadata Model for Retrieval

The discovery and retrieval of digital documents or objects from digital libraries present a number of complex issues and challenges. In the past, discovery of print material was facilitated largely by local, union, and national bibliographic catalogs. By the early 1990s, these catalogs did a fairly good job of recording the majority of bibliographic holdings in the major libraries of the United States. Most would recognize, however, that for materials in archives, the coverage was spotty at best, and the level of access not very specific. For other collections, particularly of photographs, slides, computer files, and data files, the level of cataloging and retrieval offered by most online bibliographic catalogs was negligible, and the level of information relating to the specific data object was even worse.

The electronic resources available on the Internet and the search engines specifically designed to locate and exploit these resources—coupled with the lack of interoperability between bibliographic systems and Internet systems—have done nothing but confound information retrieval and hasten information overload. What is needed is a model of digital retrieval that takes advantage of the metadata (descriptive information) inherent in digital documents and integrates them within the bibliographic and Internet search systems. Such a model would feature a scheme that 'tiers' its approach to metadata structuring for more precise object retrieval.

What would metadata tiering accomplish?

A primary goal of navigation is to locate and retrieve items of interest by presenting sufficient descriptive information to identify items or collections of note. In traditional library cataloging schemes, the creation of descriptive information is left largely to cataloging experts, and the cataloging itself is logically and physically separate from the item being cataloged. As discussed in this report, in digital formats, a significant amount of descriptive information is created as part of the document and is carried along as a structural part of the document. Moreover, while the part/whole relationship between various items of a digital collection (or within the components of a single digital document) is problematic to record and model using a traditional MARC record scheme, these relationships lie at the heart of compound digital documents and digital document repositories. By layering or tiering descriptive information across delivery platforms — local catalog, national catalog, Web site, other electronic finding devices — librarians and archivists can use all of the resources appropriate to each layer, so that retrieval and navigation of digital objects across these information spaces is more transparent to the user and, at the same time, less confusing.

What might such a tiered approach look like in today's environment?

Assume that a university decides to create a digital collection of Dime Novels for scholarship and teaching. (See http://www-sul.stanford.edu/depts/dp/pennies/.) The digital collection would make these resources widely available and would at the same time protect the artifactual (paper) collection exhibiting serious signs of fragility. The intents might be:

- to capture cover images from the Dime Novels;
- to assemble additional information about the collection for its local management (a finding aid);
- to deliver a searchable and browseable database of a representative sampling of the texts;

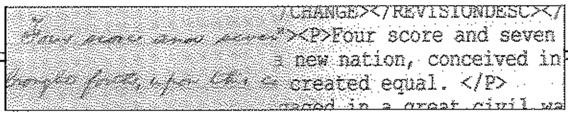
- to enhance access to the images by describing features of the images using a controlled vocabulary;
- to present the elements of the collection in a hierarchy to provide structural and sequential information and to facilitate navigation, and
- to enhance overall access by tiering or layering the metadata across multiple information-retrieval domains.

Since Dime Novels were published largely in series, the project might begin by creating standard MARC cataloging records for each series, including a few subject headings that allow the collection to be identified and aggregated. Linkages could then be created via 856 fields in the MARC record to the individual sections within the Web pages pertaining to these individual series. These Web pages would include additional information about the collection such as a detailed holdings statement, location, box number, and so on, and also act as the aggregator for the items held within that title, such as the images and full texts. The images themselves would have additional 'feature' information that would be used to create sub-collection-level records to be distributed back into the bibliographic database. The included texts would be encoded using the TEI, linking in images where necessary. These texts then could form part of a searchable database, be used to render HTML (or, in the future, XML) representations of the texts, or both. Lastly, each of the individual Web pages that presented the collection could include Dublin Core information for capture by Internet search engines,[34] and the local bibliographic records could be distributed to national utilities like RLIN or OCLC for inclusion in their databases.

The metadata for the creation of such a tiered, interrelated system could be represented in SGML, using the EAD to organize the interrelationships, the TEI to manage full-text information and for bibliographic control of the individual document-like elements within the site, and the links to the associated images. Metadata for the images could be similarly contained in EAD/TEI documents. All of this information content would be stored in a database, from which the Web pages containing the requisite Dublin Core tags, and perhaps even the MARC cataloging records for local systems/national utilities, could be generated dynamically through SGML processing systems (or delivered to SGML processing systems from other applications).

This system would permit the location and discovery of information about the site, items within the site, and related items from wherever one entered the environment: local catalog, national catalog, Web, or other networked retrieval system. The tiering of the data would not permit a user to know immediately all of the relevant information about the site from the highest level (MARC record or Dublin Core retrieval), but it would be sufficient for quick and easy navigation of the digital collection.

In the longer run, the current work underway to define a Z39.50 access protocol for digital collections[35] offers a navigation methodology that would even more tightly bind the metadata components together and facilitate the navigation and layering of metadata. In addition to defining record access protocols to digital collections and helping clarify the confusion between digital collections/objects and collection/object descriptive records, the protocol also presupposes *companion profiles* — compatible extensions to the main access protocols. These companion profiles can be aimed at more refined and domain-specific applications[36] or sets of data. This approach should integrate well with current Z39.50 applications and Z39.50-aware software and server solutions. This would provide powerful tools for an integrated discovery, delivery, and use environment.


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal. </P>

Conclusion

When viewed from the perspectives of the preservation community and the digital librarian, SGML appears to be the best choice for archival document storage. SGML is standardized, portable, flexible, modular, and supported by the market. SGML's support of re-usable modules and its vendor neutrality, extensibility, standardization, and ability to manage large information repositories make it an attractive choice for both archival and retrieval purposes.

The use of SGML, particularly in conjunction with the DTDs and related metadata standards discussed here, represents a firm data/metadata foundation on which to build the digital libraries of the future. In the current environment, the integrated, tiered retrieval of digital documents in large digital collections is neither easy nor inexpensive. However, the models and tools for creating a better environment do exist.

The development of additional standards along the lines of the Z39.50 profile for digital collections, the application and domain-specific extensions offered by organizations such as CIMI, and the container framework architecture posited by the Warwick framework point toward a future where all components will be more easily managed by the creators and disseminators of information; more easily located by users; and more easily delivered, maintained, and preserved by librarians and archivists.

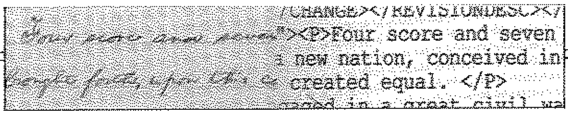
TECHNICAL SUMMARY

General

- The juxtaposition of preservation and access—which in paper-based collections can become very much an either/or—becomes completely complementary in the digital arena.
- Librarians and archivists need to be sure that they are developing formats and systems that allow users and researchers to create, share, access, and maintain digital resources across institutions and over networks.
- Those building the digital library as a virtual system of distributed resources accessible and available to any authorized user from any authorized location must employ at least a minimal set of standard protocols to manage system interactions and data interchange.
- Any system or digital data format that librarians employ either as users or producers of digital documents in the emerging digital library infrastructure will need to meet the following requirements:
  - Adequate to the task
  - Non-proprietary
  - Widespread
  - Portable
  - Persistent
  - Data-rich

SGML

- The SGML framework offers a generalized set of facilities that meets the requirements of the preservation community for archival document storage. It is standardized, portable, flexible, modular, and supported by the market.
- Three general SGML features—modularization, extended document reference, and encapsulation—used in combination and with community- and standards-based DTDs offer an integrative path for the digital library to accomplish its objectives of access and sharing.
- By using DTDs and valid SGML documents that follow them, it is possible to exchange digital information between different systems so that they can use the data and understand its syntax.
- SGML is able to reference other documents, elements within documents, documents on remote systems, and elements within documents on remote systems. This referencing ability is more robust within SGML than is currently implemented within the HTML DTD (basic for the Web).


/CHANGE></REVISIONDESC></
"><P>Four score and seven
a new nation, conceived in
created equal. </P>

Endnotes

[1] See *A Hybrid Systems Approach to Preservation of Printed Materials*, by Don Willis, 1992. Washington, DC: Commission on Preservation and Access.

[2] See *Preservation in the Digital World*, by Paul Conway, 1996. Washington, DC: Commission on Preservation and Access.

[3] NDLF Planning Task Force, Final Report-June 1996, http://lcweb.loc.gov/loc/ndlf/plntfrep.html.

[4] The term "digital document" refers to a digital object that may comprise one or more digital components. The document itself might only be an SGML file that associates, connects, and establishes the relationship between the various components.

[5] Note that a similar set of questions could be posed of textual, content-based objects. Images are used here because they are a more familiar, and thereby more obvious, example.

[6] Archival resolution can be defined as the highest resolution that meets the objectives of the collection, yet remains within the boundaries of economic feasibility. It could include a combination of resolution and gray scale or color depth (represented in bits/pixel).

[7] See, *Digital Image Collections: Images and Practice,* by Michael Ester, 1996. Washington, DC: Commission on Preservation and Access.

[8] Persistence here is closely associated with integrity and fidelity. Regardless of the instantiation of the digital object (text, monochrome image, color image, sound, or video), it must be possible to convert from one format to another and recreate a high-quality representation of the original.

[9] A computer sees all data as a string of ones and zeroes. One computer "bit" contains either a 0 or a 1. Two bits can represent four different patterns: 00, 01, 10, and 11. Each additional bit doubles the number of patterns of ones and zeroes that can be represented. It takes eight bits to represent an ASCII character. In computer parlance an eight-bit representation of data is called a "byte" or a "word." (Mathematically speaking, eight bits can represent 256 patterns of ones and zeroes, but one of the bits in an ASCII representation is usually reserved as a "stop bit." Therefore, only 128 characters can be represented in ASCII.)

[10] As a tribute to the persistence of ASCII, and despite the phenomenal growth in processing power and applications since the invention of the personal computer, most computer users still use e-mail systems that are limited to lower ASCII.

[11] See the discussion of the advantages and disadvantages of micrographics and digital imaging in the paper by Willis cited earlier.

[12] Two other companies market PDFs: Hummingbird, with Digital Papers, and Tumbleweed Software, with Envoy.

[13] Adobe Capture creates PDF document files out of scanned images. One of the features permits the creation of a full-text document via OCR that stands behind the PDF bit image, and it is this document that can be searched using Adobe or other PDF indexing products. Note, however, that this feature is optional, and that the OCR in Capture is no more accurate than in other OCR programs.

[14] In the Internet publishing wars, however, PDF appears to find more use for delivering electronic documentation/electronic versions of already extant print products than for providing a stand-alone publication environment, and it is likely that this separation will grow even wider as print and electronic products diverge.

[15] Lou Bernard, *TEI EDW25: What is SGML and How Does It Help?*, 3 October 1991 revision.

[16] The SGML Declaration, a separate part of an SGML document, can in fact name any character set as its base set. Typically, this is a standard, registered character set, like Latin 1, or ISO 646:1983.

[17] Dianne Kennedy, *DSSSL; An Introduction,* http://www.mcs.net/~dken/dslintro.htm.

[18] It should be noted that the ability to use SGML in this capacity does not require that one actually store or build document databases and libraries in SGML, merely that one is able to transmit information in this format and permit access to it through the agreed-upon facilities.

[19] See the section on "special characters" for a longer discussion of how SGML handles platform-dependent artifacts..

[20] Practically speaking, a small, fairly advanced, and specialized set of features from the SGML standard, such as CONCUR, is not supported by any of the tools listed in the appendices.

[21] For a discussion of using and re-using large SGML-based document repositories, see John Price-Wilkin, "Just-in-time Conversion, Just-in-case Collections, Effectively leveraging rich document formats for the WWW," *D-Lib magazine*, May 1997, http://www.dlib.org/dlib/may97/michigan/05pricewilkin.html.

[22] CALS (Continuous Acquisition and Lifecycle Support; originally Computer-aided Acquisition and Logistics Support) is one of the first large-scale SGML applications for the military environment.

[23] *Extensible Markup Language (XML): Part I. Syntax, W3C Working Draft 31-Mar-97*, http://www.textuality.com/sgml-erb/WD-xml-lang.html.

[24] Web Week, April 14, 1997, page 6.

[25] *A Thumbnail History of the TEI,* http://www-tei.uic.edu/orgs/tei/info/hist.html.

[26] http://lcweb.loc.gov/ead/eadback.html.

[27] http://www.cni.org/pub/CIMI/www/framework.html.

[28] Richard Light, *Getting a handle on exhibition catalogues: the Project CHIO DTD,* http://www.cimi.org/Project_CHIO_DTD.html.

[29] Lou Bernard and Richard Light, *Three SGML metadata formats: the TEI, EAD, and CIMI A Study for BIB-LINK Work Package 1.1,* sections 2.3 and following.

[30] http://www.cimi.org/CHIO.html.

[31] OCLC/NCSA Metadata Workshop: The Essential Elements of Network Object Description March 1-3, 1995, http://www.oclc.org:5046/oclc/research/conferences/metadata/.

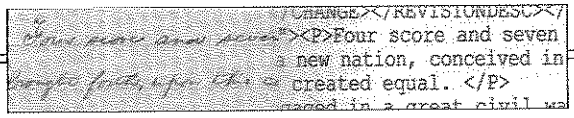
[32] The term "document-like objects" was specified in the course of the workshop both to narrow the range of discussion and at the same time to name what was believed to be the most common type of resource sought in the Internet.

[33] Carl Lagoze, Clifford Lynch, Ron Daniel, *The Warwick Framework A Container Architecture for Aggregating Sets of Metadata,* TR96-1593, 1996.

[34] This assumes that such search engines begin to support the use of Dublin Core elements more adequately than they have to date.

[35] http://lcweb.loc.gov/z3950/agency/profiles/collections.html.

[36] The work of CIMI to develop an application-specific profile within this context should be noted.



"><P>Four score and seven
a new nation, conceived in

gaged in a great civil wa

Bibliography and Resources

Alexander, George, and Alschuler, Liora. "SGML Europe '96: What's the Next Step for SGML?" *The Seybold Report on Publishing Systems*, June 30, 1996.

Arms, Caroline R. "Historical Collections for the National Digital Library, Lessons and Challenges at the Library of Congress." *D-Lib Magazine*, April, 1996. (http://ukoln.bath.ac.uk/dlib/dlib/april96/loc/04c-arms.html)

Aschuler, Liora. *ABCD...SGML*. London: International Thomson Computer Press, 1995.

Bartlett, P.G. Personal correspondence. November 1996.

Boeri, Robert J., and Hensel, Martin. "Corporate Online/CD-Rom Publishing: The Design and Tactical Issues," *CD-Rom Professional*, pp. 77-9. February, 1996.

Bray, Tim. "Measuring the Web." Fifth International World Wide Web Conference, May 6-10, 1996, Paris, France. (http://www5conf.inria.fr/fich_html/papers/P9/Overview.html)

Bray, Tim. (http://www.textuality.com — home of Bray's consulting service)

Colby, Martin and Jackson, David. *Using SGML*. Indianapolis: Que Corporation, 1996.

Cover, Robin. "SGML Bibliography: News and Popular Sources." 1996. (http://www.sil.org/sgml/bib-misc.html)

Cover, Robin. "SGML: What's New, Relatively New, or New in the 'SGML Web Page'?" 1996. (http://www.sil.org/sgml/sgmlnew.html)

Dempsey, Lorcan, and Weibel, Stuart. "The Warwick Metadata Workshop: A Framework for the Deployment of Resource Description." *D-Lib Magazine*, July/August 1996. (http://www.dlig.org/ dlib/july96/07weibel.html)

DuCharme, Bob. "DBMS Support of SGML Files." 1996. (http://ww-hci.stanford.edu/sgml/ bdc_sgml/dbms.html)

"EAD Progress Reports." 1995. (http://sunsite.berkeley.edu/FindingAids/EAD/history.html)

Fleischhauer, Carl. "Organizing Digital Archival Collections: American Memory's Experience with Bibliographic Records and Other Finding Aids." 1995. (http://www.sil.org/sgml/amm-carl.html)

Gaynor, Edward. "From MARC to Markup: SGML and Online Library Systems." 1996. (http://www.lib.virginia.edu/speccol/scdc/articles/alcts_brief.html)

Heery, Rachel. "Review of Metadata Formats." Program Vol 30, Issue no.4, October 1996. (http://www.sil.org/sgml/heery-review.html)

Herwijnen, Eric van. *Practical SGML*. Massachusetts: Kluwer Academic Publishers, 1994.

Karney, James. "SGML: It's Still a la Carte." *PC Magazine*. February 7, 1995.

Kunze, John A. and Rodgers, RPC. "Z39.50 in a Nutshell (An Introduction to Z39.50)." *Lister Hill National Center for Biomedical Communications, National Library of Medicine*. July 1995. (http://www.nlm.nih.gov/publications/staff_publications/ rodgers/z39.50/z39.50.html)

Lapeyre, Deborah A. and Usdin, Tommie. "TEI and the American Memory Project at the Library of Congress." March 14, 1996. (http://www.cs.vassar.edu/~ide/DL96/Lapeyre)

Larson, Ray and McDonough, Jerome. "USMARC DTD and utilities to convert USMARC records to SGML formatted records." 1996. (http://www.ua.ac.be/WGLIB/MARCSGML/sgml.html)

Library of Congress Network Development and MARC Standards Office. "The USMARC Formats: Background and Principles." 1994. (http://lcweb/loc/gov/ loc/standards/...)

Library of Congress Network Development and MARC Standards Office. "Changes to FTP Label Specifications for Electronic Files of USMARC Records." Proposal 96-7. May, 1996. (http://lcweb/loc/gov/loc/standards/...)

Library of Congress Network Development and MARC Standards Office. "Changes to Field 856 (Electronic Location and Access) in the USMARC Formats." Proposal 96-1. December, 1995. (http://lcweb/loc/gov/loc/standards/...)

Library of Congress Network Development and MARC Standards Office. "Mapping the Dublin Core Metadata Elements to USMARC." Discussion Paper DP86. May, 1996. (http://lcweb/loc/ gov/ loc/standards...)

"NICE Technologies SGML Tools Listing." November 08, 1996. (http://www.nicetech.com/venprod.htm)

Noreault, Terry R., and Crook, Mark A., Office of Research, OCLC Online Computer Library Center, Inc. "Page Image and SGML: Alternatives for the Digital Library." 1996. (http://www.sil.org/sgml/noreault.html)

Pepper, Steve. "The Whirlwind Guide to SGML Tools and Vendors." 1996. (http://www.falch.no/ people/pepper.html)

Pitti, Daniel V. "The Berkeley Finding Aid Project, Standards in Navigation." 1995. (http://sunsite.berkeley.edu/FindingAids/EAD/arlpap.html)

Pitti, Daniel V. "Settling the Digital Frontier: The Future of Scholarly Communication in the Humanities." April 1995. (http://gopher.sil.org/sgml/pittiFrontier.html)

Pitti, Daniel V. "Standard Generalized Markup Language and the Transformation of Cataloging." 1994. (http://www.sil.org/sgml/berknasg.html)

Prescod, Paul, and Bunn, Bill. "Multiple Media Publishing in SGML." (http://www.incontext.ca/ techturf/sgml.html)

Severson, Eric. "How SGML and HTML Really Fit Together: The Case for Scalable HTML." January 1995. (http://www.sil.org/sgml/severson/html)

Southwick, Karen. "How Far Can Serendipity Carry Adobe?" September 1995. (http://upside.master.com/print/sept95/9509fl.html)

Sperberg-McQueen, CM and Burnard, Lou. "Guidelines for Electronic Text Encoding and Interchange." Chicago and Oxford, Text Encoding Initiative, 1994. (http://etext.lib.virginia.edu/TEI.html)

Summers, Kent. "How to Select an Electronic Publishing Solution That's Right for You: A 'Best Fit' Model." March 1996. (http://www.kents.com/docs/pub-mar96.htm)

Thompson, Marcy "An Element Is not a Tag: (and Why You Should Care)," paper delivered at SGML '96, Boston MA. November 1996.

Turner, Fay. "An Overview of the Z39.50 Information Retrieval Standard." UDT Occasional Paper #3. July, 1995. (http://www.nlc-bnc.ca/ifla/VI/5/op/udtop3.html)

Weibel, Stuart, et al. "OCLC/NCSA Metadata Workshop Report." 1995. (http://www.oclc.org/ oclc/research/conferences/metadata/dublin_core_report.html)

Weibel, Stuart. "Metadata, The Foundation of Resource Description." 1995. (http://www.oclc.org/ oclc/research/publications/review95/part2/weibel.html)

Willis, Don. "A Hybrid Systems Approach to Preservation of Printed Materials." November 1992. (http://www-cpa.stanford.edu/cpa/reports/willis/index.html)

Appendix 1: Practical Concerns: Using SGML in Conversion and Authoring Projects

This report has concentrated on describing how SGML-encoded documents, preferably using or compliant with the major, community-based DTDs, can be used in the creation of digital collections. This section explains the two distinct ways that SGML is used in digitization projects. Considerations for *conversion to* SGML are quite different from those of *authoring with* SGML.

Conversion

Conversion to SGML—whether from existing digital or printed form—is a costly proposition.

For materials already in digital form, a number of conversion tools are available, many of which are discussed in Appendix 3. These conversion tools use a variety of procedural markup information to help map electronic information into the conversion, or target, DTD. All have a programming interface to assist in the conversion; some employ a graphical user interface (GUI-front-end) to make the tools easier to use. But before any of them can be used, it is necessary to understand the nature of the documents to be converted and the target DTD, and to develop a conversion strategy that maps the legacy data into the target DTD. Coming to an acceptable conversion strategy can be a long and arduous process. However, once it is taken on, it usually results in a set of decisions centering on the question, *"How much is enough, given current constraints?"*

These constraints can be considerable, because in most instances, even the best tools will fail to convert significant portions of the legacy documents, and these portions must be cleaned up by hand. SGML authoring tools can facilitate and speed up this process, but manual clean-up still will be necessary. Since SGML concerns itself with the structure of a document, and not with the appearance, it should not be surprising that these conversion efforts can prove difficult. The points made earlier about the difficulties retrieval systems would face in 'understanding' document structure on the basis of its presentation hold equally well here. One must also consider the depth and complexity of the conversion target DTD. The more complex, the greater the likelihood for significant manual intervention. On the whole, projects that seek to convert a large number of documents that are similar in their use of structural clues — layout, styles, or the like — will be the most cost-effective overall. For these projects, a significant investment in conversion tools and custom programming can have an equally significant payoff. Projects, however, that need to convert a large number of relatively heterogeneous documents may find that the level and depth of the conversion efforts will be hampered by having to treat each document separately.

If conversion efforts begin with a printed product rather than electronic files, the costs of conversion from print to ASCII must be added in as well. This conversion can be handled in one of three ways: (1) by scanning/OCR assembly lines, with the addition of some rudimentary SGML tagging in SGML-aware programs; (2) by keyboarding, usually using a technique called 'double keying';[1] 3) by some combination of these two.[2]

Although scanning can prove effective for some, particularly more recent, materials, the best scanning/OCR programs are unable to read and convert documents well without additional editorial clean-up. Consider, for a moment, programs that claim and actually achieve a 99.9% accuracy rate. This would mean that one character in a thousand is wrong. For this report, that would represent approximately two errors per page, a rate that no publisher would permit. That 99.9% error rate is generally much better than even the best OCR programs are able to achieve. Depending on the age of the material, the editorial clean-up can be so extensive that scanning/OCR proves more expensive than keyboarding.

SGML Authoring

Projects that can use SGML authoring tools from the very start (e.g., with the creation of the document) will face none of the costs involved with conversion. This is not to say that the transition from current authoring practice to an SGML-authoring environment will be easy or straightforward, despite the tangible advantages. Why is this so?

Since SGML and its resulting DTDs define document structure and set forth a set of rules about what document elements are permitted in what order and in what situation, an SGML-aware editor will enforce the structures that are given by the DTD. This means that authors must understand the DTD and employ it successfully. It also means that they should not think about document style or layout as they compose, but instead focus on content and structure. Most authors are used to word processors and procedural markup, but most are probably unaware, or only fleetingly aware, of the style-sheet capabilities of their word processors. And to many authors, the notion of some kind of stylistic enforcement would be completely foreign, if not anathema.

Successfully creating an SGML-authoring environment may well require both a conceptual reorganization of the authoring process as well as the physical reengineering of the work environment. SGML authoring requires common understanding, common goals, some authorial discipline, and a good deal of 'friendliness' on the part of the authoring software. All of the SGML-aware authoring software shares with word processors the ability to tailor the individual editing environment, and the best software provides the ability to completely customize the environment from the points-of-view of both the user and the application (or DTD).[3] In addition to customizing the editing environment, the software is increasingly providing WYSIWYG (what you see is what you get) capability. This capability has no effect on the resulting SGML output, but it gives the documents a 'visual intelligence' that is appealing to the author.

Finally, the SGML environment requires a good deal of ancillary facilities: for the storage of documents, for document and document-component tracking, for style-sheet creation (DSSSL and others), and for creating useable products for whatever is required by the publishing agency. All of these factors combine to form a commitment to at least a partial, if not full-blown, 'SGML view' of the world that is not trivial, either in terms of cost or time. However, as SGML gains greater acceptance, costs can be expected to decrease, and the availability of tools and facilities to exploit the value of SGML will increase.

[1] Double keying involves two operators entering the same text. A comparison program is run on the resultant two files to look for discrepancies to be resolved. Many data-entry houses now specialize in adding the proper SGML markup to the text as part of the conversion process. Depending on how complicated the DTD is and on the agreed-upon 'acceptable error rate', a conversion effort like this might cost $1.20 to $1.50 per 1,000 characters, according to Boeri and Hensel, 1996, or $1.50 to $2.00, according to other sources.

[2] This process would benefit from a system designed to segment either manually or automatically the page image into text, graphical, or grayscale areas. Once the segmentation has been completed, then the digital objects (entities in SGML terminology) can be processed and stored separately so as to maximize fidelity and minimize required storage.

[3] Many of the word processors are offering SGML add-ons (Word and WordPerfect, specifically). WordPerfect appears to have done the better job of offering something close to SGML support. Both the Word and WordPerfect product add-ons take pains to hide the SGML from the user, and both fail to implement features that would enforce compliance with the DTD being used. This means that documents created by these systems can fail to pass validation tests.

Appendix 2: Markup Formats

The formats below often come up in discussions about data storage and presentation.

Procedural Markup

PDF

See also "Presentation Format" in the section, "Exploring Digital Formats for Preservation and Access"

Even though only a fraction of Internet documents are currently published in PDF (Portable Document Format), it does have its place in retaining complex document formats. Clement Mok, president and CEO of Clement Mok Designs Inc. in San Francisco, which helps publishers design Web pages, believes that Acrobat by Adobe will eventually be among a group of four or five formats that will be standard on the Internet. "We don't want to have 27 flavors, but neither do we want just one," he says. Still, most of his clients continue to publish in HTML.

Some proponents of both PDF and HTML argue that they are competing standards, but it is more useful to regard them as complementary standards. If a user needs to view or print an exact representation of the original page, then PDF is very effective. PDF also provides some rudimentary text search and hypertext-linking capabilities, which extends its usefulness beyond just viewing a document. If, on the other hand, a user has the need for multi-modal output (i.e. the ability to produce many different views of the same data), then HTML is the way to go.

The sticking point in the argument over whether PDF is a better format solution than HTML mainly concerns the best way to deliver information across the World Wide Web. PDF proponents argue that because it is based on a page description language, it gives Web content providers the ability to retain the "look and feel" of a printed document for their data. Its presentation of tables, charts, and graphics is also superior to what can currently be achieved in HTML. However, any information about the logical structure of the document is lost in PDF, as is the ability to embed searchable keywords. PDF is also not editable data. There are no PDF editors available that can open a PDF file and allow users to modify it. Given this, PDF must be regarded as strictly an output format.

The distinction between PDF and HTML blurs because HTML is also used for presentation of data. Most of the general public's exposure to HTML comes via the World Wide Web, where the vast majority of the information is presented in HTML format. This can give the impression that HTML is only an output format for file presentation. Unlike PDF, however, many tools have been written for the editing of HTML files. PDF files can be viewed by using a "plug in" program for the browser, that will be called in a separate viewing window to handle the PDF data stream.

The "plug in" technology allows all sorts of viewing programs to be attached to the main browser. OCLC is involved in an initiative to make 50 academic journals available in PDF format on the Web by 1997. OCLC is a prime example of a publishing enterprise making use of both HTML and PDF.

TeX and LaTeX

TeX and its descendent, LaTeX , are markup formats particularly well-suited for technical publications such as math and science journals. TeX was invented by programmer Donald Knuth, who was frustrated at the options available to him for the presentation of complex mathematical formulas. However, as Eric van Herwijnen points out in *Practical SGML*, "The language (TeX)... is very typesetting oriented and difficult to convert into a structured language that captures the mathematical intent of the formula."

DVI (Device Independent) Format

The DVI format, an intermediary format between the TeX format and an output format, contains formatting commands that can be further translated into commands for a variety of output devices, including PostScript devices.

Postscript

Adobe Corporation's Postscript was originally written as a printer description language. As such, it translates data from textual and graphics systems into a format that can be understood by print output devices, such as laser printers. Although Postscript is a proprietary language owned by Adobe, it has emerged as the *de facto* standard for controlling printer output. Postscript's limitations include its lack of support for hypertext linking and the amount of processing power required to convert the Postscript code into an output format. On the Web, Postscript has all but been eliminated by PDF. Furthermore, because Postscript is densely coded with formatting instructions, it is not easily converted to other formats.

Frame MIF (Maker Interchange Format)

MIF is the format the FrameMaker application (formerly marketed by Interleaf, now marketed by Adobe) generates as an intermediary conversion file between its binary storage format and a chosen output format. In this sense it is analogous to the DVI format. An MIF document can roughly be divided into four parts: specification of document structures (style sheets, etc.), definition of tables and frames, page layout information, and text.

RTF (Rich Text Format)

RTF was developed by Microsoft to allow the sharing of data between the various applications it markets. An appearance-based language, RTF contains no information about the structure of documents. RTF cannot describe equations or other types of specialized markup. Finally, RTF is a proprietary markup format, and the vendor is liable to change the markup specifications at any time (as was the case for many releases of RTF for the Microsoft Word product).

Binary word processing formats

Microsoft Word and WordPerfect are two popular word processing software packages that store and work with files in a binary format. (Binary files contain codes that are not part of the ASCII character set. Binary files contain information that can be represented by an 8-bit byte—a possible 256 values.) Because Word is the most popular word processor in the world, supported by one of the largest software companies in the world, its output format cannot be readily dismissed as a viable access format. This consideration has nothing to do with its usefulness as a storage format—for which it is not suited—but rather because many users consider it to be a storage format. WordPerfect is also a very popular program in wide use, and it has to be given the same consideration.

Descriptive Markup — Metadata

HyTime (ISO 10744)

HyTime, the Hypermedia/Time-based Structuring Language, is an ISO standard for the representation of open "hypermedia" documents and is an application of SGML. Hypermedia, also known as "interactive multimedia," allows random access to information in a program. Unlike normal video, for instance, which starts at an unchangeable beginning and proceeds through to a predetermined end, hypermedia allows the viewer to control a program's pace, explore sections within it, and determine where the beginning and the end are. An interactive multimedia document lets users explore at their own pace.

Like SGML, HyTime does not specify the document content, but provides a standard way in which different types of information can be combined. It is not a DTD, but provides guidelines for DTDs. For example, the Standard Music Description Language (SMDL) and the Standard Multimedia Scripting Language (SMSL) are applications of HyTime with their own DTDs that conform to the HyTime standard. HyTime allows the association of objects within documents using hyperlinks, and the interrelation of objects in time and space. An object in HyTime is unrestricted—it could be video, sound, text etc. A HyTime engine recognizes HyTime documents and processes them. Several commercial HyTime engines are being developed.

SMDL (Standard Music Description Language)

This International Standard defines an architecture for the representation of music information, either alone or in conjunction with text, graphics, or other information needed for publishing or business purposes. Multimedia time sequencing information is also supported. It is intended for the transmission and storage of music and hypermedia information, and is optimized for completeness, flexibility, and ease of use. SMDL is a HyTime application. Typical original sources of works that can be represented in SMDL are printed scores, performances recorded in a manner that permits machine transcription, pieces that are already represented in a (non-standard) music description language, and compositions generated by computer-based compositional and/or notational aids that have the capability of exporting the pieces as SMDL.

SMSL (Standard Multimedia Scripting Language)

The rationale behind developing the SMSL is to support the same kinds of capabilities in SGML authoring tools that are now available to users of multimedia authoring tools, and to do so within the SGML environment. Currently, SGML authoring tools support the creation of documents in SGML markup, including the addition of hyperlinks to entities containing graphics, audio, and video. However, unlike the user of multimedia authoring tools, the SGML tool user has no control over the placement of graphics and video or the duration of their display and the playing of any accompanying audio. SGML itself does not define the semantics for specifying the required parameters; HyTime, through its scheduling, measurement, and rendition modules, does. SMSL will provide the means for building applications that exploit these features of HyTime by providing an object oriented interface between any programming language that can support the services required by SMSL and SGML/HyTime documents. Through the use of these applications, authors will be able to specify how an "anchor" is rendered when a link is selected; for example, authors could specify when and where a graphic is displayed.

VRML (Virtual Reality Markup Language)

VRML represents an attempt to make three-dimensional graphics technology, which historically required expensive high-end workstations, available to the millions of desktops connected to the Web.

MARC (Machine-Readable Catalog) Format

The international MARC standard was developed in the late 1960s to facilitate the exchange of catalog records between libraries. Like SGML, the MARC standard defines a record structure but does not prescribe the record's content within the structure. It has long been the primary format used by libraries to electronically represent and communicate bibliographic information, and it is used today in thousands of libraries in the United States and Europe. Specific adaptations include USMARC (in the United States), CANMARC (in Canada),and UKMARC (in Great Britain). All MARC formats conform to ISO 2709, which defines the standard for bibliographic information interchange on magnetic tape. The record structure reflects the requirements of computers in the early 1970s, when systems were tape-based to keep data storage at a minimum. MARC records are discrete bibliographic entities, each consisting of a directory pointing to variable length fields with content designators. Because the format was designed for exchange on magnetic tape, record separators are defined for data streams. The USMARC format has undergone several revisions during its long life in an effort to keep it current with ever-changing technologies. Recent revisions include changes to the record label specification for support of File Transfer Protocol (FTP), and new "linking" fields that describe access and location of networked resources.

Related Formats and Standards

Many of the formats and standards listed below are only peripherally related to SGML, but they are treated here because they often come up in discussions of SGML data management.

OpenDoc

OpenDoc was developed by a consortium whose members include IBM, Apple, Novell, and Lotus. While SGML and MARC are data standards, OpenDoc is an application standard. Documents or other data "objects" may be marked up in proprietary formats, but OpenDoc sets standards for how these objects are accepted into or manipulated by various applications. A standard like OpenDoc is sometimes referred as a component integration technology for object-oriented development.

OLE (Object Linking and Embedding)

OLE, developed and supported by Microsoft, is a competing application standard to OpenDoc. OLE has been defined as follows: "OLE is a unified environment of object-based services with the capability of both customizing those services and arbitrarily extending the architecture through custom services, with the overall purpose of enabling rich integration between components." OLE allows one to essentially embed a part of one application in another.

CORBA (Common Object Request Broker Architecture)

CORBA is an open distributed computing infrastructure that automates many network programming tasks such as error-handling and the registering, locating, and activating of objects on a network. Essentially, CORBA allows application "objects" to communicate directly with other application objects wherever they may be located on a network. Under this scenario true distributed computing is possible across a network, where one computer on a network can access the power, memory, and storage of another computer on the network, even if the second computer is remote from the first.

ODA (Open Document Architecture)

ODA is an international standard (ISO 8613) aimed specifically at handling office documents. ODA does not provide a grammar for users to define their own document structures, rather ODA strictly defines a set of structures to which users have to conform. Although ODA is supported by the European Economic Community, commercial ODA-compliant applications are in short supply.

Z39.50

Z39.50 (formerly known as ANSI/NISO Z39.50-1995-Information Retrieval) is a network protocol or set of rules that govern database searching and information retrieval between communicating computers systems. The standard was invented to deal with the problem of searching multiple databases, each of which might have unique menus, command languages, and search procedures. Through the use of the Z39.50 protocol, the search process is simplified because the searcher can use a familiar user interface on his or her home system and the protocol can provide the right query to disparate databases on remote systems.

Java

Java is an general-purpose object-oriented programming language featuring multi-threading, solid interface design, and a simplified C++-like syntax. Implementations of the language exist for both the Window and UNIX platforms. Java is translated to byte-codes upon compiling, making it possible to run Java programs on an computer with a byte-code interpreter. Some World Wide Web browsers incorporate byte-code interpreters, which means that Java code can be added to any HTML page. Java was developed in 1990 at Sun Microsystems.

Appendix 3: SGML Tools

Data Conversion Tools

Choosing the proper tools to convert legacy documents into SGML-formatted data is an important consideration. The solutions to this problem fall into three basic categories: in-house programs and scripts written in c, perl, or other languages; commercial software packages created especially for conversion efforts; and service bureaus that specialize in conversion.

There is also often a need to convert from SGML markup to other forms of markup — including HTML. This is a very effective model for preserving a very rich document, and for translating it to a less rich display or print format. It allows the interchange and display systems to be significantly simplified. Tools that translate to and from SGML are considered below.

Balise

Balise, marketed by the French firm Advanced Information Systems/Berger-Levrault, is a high-level interpreted language similar to the UNIX utility, awk. Balise can be used to translate an SGML document to another format, another format to SGML, or from one form of SGML to another (in conversion terms: up-translation, down-translation, and cross-translation). The latest release of the product (Balise 3.1) also supports linking directly to a database. Balise does require some amount of programming skill to use.

CoST

CoST (Copenhagen SGML Tool) is an SGML structure-driven tool based on the Tcl (Tool Command Language, developed at Berkeley). The user needs to write CoST scripts for every source DTD. Since CoST is a low level programming tool, a user needs working knowledge of SGML and Tcl to use it effectively.

DynaTag

Electronic Book Technologies' DynaTag employs a "point-and-click" setup to allow its users to create style sheets for converting word-processing files to SGML-conforming files. The same technology can be used to convert a document that conforms to one DTD to a document that conforms to a different DTD. Many users of DynaTag store their data in a more complex SGML model and convert it to the simpler HTML model on the fly whenever the data is called for display.

FastTAG

Interleaf's FastTAG tool is another commercially available SGML-aware data conversion tool. Like OmniMark, FastTAG can convert data from virtually any document format into SGML-tagged data and vice-versa. FastTAG uses visual recognition technology (i.e., taking structural cues for the document from the document's layout) to convert inconsistently marked up text. Both OmniMark and FastTAG require some programming background in order to build the necessary conversion rules.

Fred

Fred, the SGML Grammar Builder Project, can automatically build DTDs from tagged text. It includes a translation language that allows direct mappings based on tag names, attributes, and structure, as well as movement of text within the tree and several other manipulation features. It is especially useful in conversion efforts where a user has SGML marked up data, but not a DTD to go along with it. Fred can build the requisite DTD by parsing through the tags in the supplied documents. Fred is an ongoing research project at OCLC (Online Computer Library Center).

marc2sgml and sgml2marc

The marc2sgml conversion tool translates files incorporating USMARC to files that parse against the USMARC.DTD (document type definition). The sgml2marc tool does the back translation of USMARC files into USMARC.DTD files. The marc2sgml converter has some translation problems with older USMARC records that were created before the USMARC standard was fully formalized. These tools are available through the University of California at Berkeley's School of Information Management and Systems.

OmniMark

OmniMark, a product of the OmniMark Corporation, is a commercially available conversion tool commonly used for translating legacy data. It differs from many other text conversion tools on the market because SGML-awareness is built into the basic product. Besides legacy conversion, OmniMark is being used by many organizations to manage their repositories of data for World Wide Web distribution. In this scenario, the users maintain their base data in the much more richly tagged SGML format and convert the data on the fly — using OmniMark — whenever they want to download data onto the Web.

perl (Practical Extraction and Report Language) and perlSGML

Perl, developed and maintained by programmer Larry Wall, is a computer language optimized for textual processing. It excels particularly in the areas of text recognition and pattern matching, making it particularly well-suited for the job of converting legacy documents into SGML-tagged data. Working with perl requires a higher degree of programming knowledge than working with conversion tools such as OmniMark and FastTAG, but there are repositories of perl utilities (pre-written code) available for free on the Internet. One of the better such collection of utilities is perlSGML, but many other collections are also available.

Rainbow

Electronic Book Technologies (EBT) is providing General Translator Archive (Rainbow), a set of converters to move from proprietary word processor formats into various SGML formats, including HTML. This is achieved through a series of two translations. First, the EBT converter for a particular format converts information in that format to legitimate SGML that matches a corresponding Rainbow DTD that EBT also provides. Some other conversion process must be chosen then to translate the Rainbow-formatted information to conform to the DTD of the user's choosing. Still, it is easier to translate from

Data Conversion Tools Table*

Product Name	Notes	Supported Platforms	Vendor Name and Web Site	Unit Price
Balise	Powerful tool with an SGML parser built in	DEC, DOS, UNIX, Windows, VMS	AIS Software www.ais.berger-levrault.fr/BlWeb/index.html	£1900 DOS / Windows £3900 UNIX/DEC/ VMS
CoST (Copenhagen SGML Tool)	Tool based on Tcl language	UNIX	uranus.linguistik .uni-erlangen.de	Public Domain
DynaTag	Add-on to generate DynaText style sheets for conversion of SGML and word-processing files	Windows	Electronic Book Technologies www.ebt.com	$5,500 (1995 price)
FastTag	Multiple format conversion to and from SGML; scripting required	DEC, DOS, OS/2, UNIX, Windows	Avalanche/Interleaf www.avalanche.com	$3500-4000 (1996)
Fred	Builds DTDs from tagged text	UNIX	OCLC www.oclc.org	Public Domain
marc2sgml sgml2marc	Developed at University of California - Berkeley	UNIX	www.ua.ac.be/WGLIB/ MARCSGML/sgml.html	Public Domain
OmniMark	Validating parser; multiple format conversion to and from SGML; scripting required	DEC, DOS, Mac, OS/2, UNIX, Windows	Omnimark Technologies Corp. www.omnimark.com	$10,000 for 2 licenses, $3,000 each additional (PC and UNIX) (1997)
PerlSGML	Conversion utility based on Perl scripting language	DOS, UNIX	Maintained by programmer Larry Wall	Public Domain
Rainbow	Conversion template DTD describes word processing formats	DOS, Mac, UNIX, Windows	Electronic Book Technologies www.ebt.com	Public Domain
SGML Hammer	Library of output filters include SGML and several proprietary formats	DEC, DOS, OS/2, UNIX, Windows	Avalanche/Interleaf www.interleaf.com	$1,950 (PC), $2,730 (UNIX) (1997)
TagWrite	Converts RTF, WordPerfect, ASCII to tagged output; also bundled with Corel's Ventura	DOS, Windows	Zandar Corp.	$1,200-$1,700 (PC only) (1997)
tei2marc	Conversion utility written in Perl	UNIX	Available through the University of Virginia Library	Public Domain

*Check Web sites for most up-to-date information

one DTD to another DTD than it is to convert from a proprietary—possibly binary—format to SGML markup. The Rainbow converters handle the hard part.

SGML Hammer

Originally developed by Avalanche, which was bought by Interleaf, SGML Hammer can convert SGML documents into many electronic formats: Interleaf (text and binary markup), Interleaf Frame MIF, Microsoft Word RTF, WordPerfect, HTML, and SGML documents based on different DTDs. Because SGML Hammer is marketed by the same company that markets FrameMaker+SGML, its "filter libraries" for converting from Interleaf's proprietary MIF format to SGML are especially robust. SGML Hammer can be directly integrated into an SGML application through its built-in API functions.

TagWrite

Zander Corp.'s TagWrite is a tool for converting RTF, WordPerfect, or ASCII files into any tagging scheme, including SGML. TagWrite employs user-configurable templates to accomplish the translations. The RTF filter is the most developed feature in TagWrite and, as such, TagWrite is often licensed technology for applications that have to either read in or output the text in the RTF format.

tei2marc

The tei2marc conversion tool, written in perl, translates the headers of files marked up in the TEI.DTD (document type definition) to USMARC-formatted ASCII files. In essence, it extracts the bibliographic data from the top of a TEI file and translates that header information into the USMARC. It also creates a log file of its conversion activities. The program is available from the University of Virginia Library.

Database and Document Management Tools

A variety of schemes have been devised to manage SGML documents in a systematic way. Some vendors use and extend traditional SQL (standard query language) database technology to store and retrieve information about SGML objects (structured parts of SGML documents). Other vendors rely on the newly emergent object-oriented databases that can be structured in the same way that SGML documents are structured — as inter-related, inter-linked, reusable text objects. On top of this, other integrators also add a workflow management layer, so that organizations can control the circumstances under which users can retrieve and edit document objects. Described below are some of the approaches these vendors have used.

Astoria

Astoria, originally developed by Xerox, is an object-oriented document component management system that enables users to find and share SGML documents and their components. Like OpenText, Astoria also can handle unstructured documents. Astoria incorporates a complete document management system and manages data at the structured element level rather than at the file level. These components can be accessed through Astoria's text searching and retrieval functions. Because Astoria stores information at the level of document components (parts of documents), it can reuse those components in different documents. Astoria also has the capability of storing and tracking unstructured documents.

BASIS SGMLserver

The BASIS SGMLserver is an SGML document management layer built on top of an already existing SQL relational BASISplus database. SGMLserver can find disparate SGML objects from the database and build them into a document on the fly, building a hypertext-searchable table of contents for the document or document parts at access-time. SGMLserver also supports searches of the database based on SGML elements and presents the results of the search (the document objects) in a full-featured browser.

Dynabase

Dynabase, created by Electronic Book Technologies — a company recently acquired by Inso Corporation — is an application designed to manage data for World Wide Web distribution. Because Web documents are not static, but can be interlinked with thousands of other documents, managing the data on a Web site can be difficult. Dynabase handles file maintenance, the sharing of files among a team of editors, and the automatic updating of those files and any files that link to them. Built to be used in a distributed work group environment, it is also suitable for managing data on an institutional intranet (in-house network). One drawback for now is that it is designed to work only with HTML files.

Information Manager

The Texcel Information Manager (IM) document management system consists of an integrated set of applications for dynamic authoring, managing, and configuration of document information. Texcel IM's workflow system operates on SGML-compliant data and an object-relational repository. Texcel's marketing literature describes the application in these terms: "Texcel Information Manager is a multi-tiered software suite. The Repository Manager, a secure, multi-user, SGML data server, is based on a scalable object-relational database. End-user applications for workflow, SGML editing collaborative authoring, electronic review, and document assembly interact with the Repository Manager. Applications communicate with the Repository Manager through the Application Manager. Documented APIs are available to customize all aspects of the system."

Informix

Informix has two products, Universal Server and Illustra Server, that accept software "snap-ins" that extend the functionality of the core server so that it can handle a variety of new datatypes. Informix calls these snap-ins DataBlade modules, and third party vendors have been contracted to write the DataBlades. A DataBlade incorporating search engine technology from Verity already exists for the handling of HTML data and SGML data. Since the DataBlades are linked directly to the Informix database, the query capabilities of the database can be used to search and retrieve SGML documents from the database. This technology and the Oracle technology discussed below represent the first recognition by the major database vendors that SGML is an important database storage format.

OpenText

OpenText's database technology was developed in 1985 at the University of Waterloo to build a database around the Oxford English Dictionary. Given the size and complexity of the OED, no existing technology was adequate for the purpose. What began as a collaborative effort between the University of Waterloo and Oxford University Press has grown into one of the leading SGML-based database systems on the market today. The text and retrieval schemes are designed to handle high-speed searches of large databases and search and retrieval of structured text. The OpenText system includes a string-based text search engine and is supported as a process in UNIX environments, and as a Dynamic Link Library (DLL) on MSWindows platforms. The search engine's interface is an API-oriented platform independent query language. A nice feature of OpenText is its capability of indexing and developing a database of both SGML and non-SGML data, making migration to SGML an easier task.

Oracle

Oracle Corporation markets two products that help users manage their SGML data directly. (Oracle also markets a powerful SQL relational database that is used by many SGML integrators as a tool for the storage and retrieval of SGML data.) Oracle Book is an electronic document delivery system that allows viewers to search text, browse tables of content and indices, follow hypertext links, and add annotations to documents. On top of its Book product Oracle provides the SGML Designer tool, which is a GUI interface that aids authors in the automatic mapping of any DTD to Oracle Book documents.

Parlance Document Manager

Xyvision's Parlance Document Manager (PDM) product is a database for SGML objects. It also manages workflow when the objects have been pulled out of the database for editing. Xyvision was traditionally a manufacturer of high-end paper publishing software that maintained data in a proprietary procedural markup format. They eventually added SGML-support to their product suite, adding filtering capabilities to convert their customers' SGML to their proprietary format. To manage the SGML data and to control workflow, Xyvision built PDM. Document elements are stored in a central database where they can be shared and reused in multiple documents, whether those documents be formatted for paper output, CD-Rom format, or on-line viewing. PDM also provides version control (the ability to compare various edited versions of a document object).

SearchServer

SearchServer from Fulcrum, Inc. is a full text search and retrieval engine — similar in functionality to OpenText, but designed to work in conjunction with a relational database. SearchServer features natural language search techniques and an SQL-based query language that are easy for users to learn.

Sybase

Sybase, like Informix, has recently licensed the Verity search engine technology in order to extend its own SQL relational database to work with SGML and HTML data. Additionally, Sybase has signed an agreement with World Wide Web browser supplier, Netscape Communications, Inc., to tie its product in more tightly to Web applications.

WorkSMART

InfoDesign Corporation's WorkSMART is a suite of tools that manages both SGML documents and the workflow associated with those documents. The WorkSMART database, based on OpenText technology, supports dynamic schema evolution — models of objects in the database can be modified without the requirement that all the data affected be reloaded, as would be the case in a relational database. WorkSMART also supports multiple concurrent revision trees for an SGML object. (WorkSMART was recently bought by the OpenText Corporation.)

Database and Document Management Tools Table*

Product Name	Notes	Supported Platforms	Vendor Name and Web Site	Unit Price
Astoria	Object-oriented management; formerly marketed by Xsoft	UNIX server, Windows client	Chrystal Software www.chrystal.com	$50,000 for 1 server and 10 licenses (PC and UNIX) (1996)
Basis SGMLserver	Relational search, retrieval; built on Basisplus engine	DEC, IBM, UNIX servers, Windows client	Information Dimensions Inc. www.idi.oclc.org	$32,000 (PC and UNIX) (1995)
DynaBase	Works only with HTML, not SGML	UNIX	Electronic Book Technologies www.ebt.com	for pricing visit http://www.inso.com
Information Manager	Workflow, data management; customization required	UNIX, Windows client	Texcel www.texcel.no	$25,000 for server and 4 licenses (UNIX server, UNIX or PC licenses (1997)
Informix	Verity search engine licensed for handling SGML and HTML	UNIX, Windows client	Informix www.informix.com	$1,554 (PC and UNIX) other versions go to $20,000
OpenText	Database; supports indexing of SGML and non-SGML systems	DEC, IBM, UNIX, Windows client	OpenText Corp. www.opentext.com	$20,000
Oracle	SQL relational database, integrated with many SGML systems	UNIX, Windows	Oracle Corporation www.oracle.com	$1,475 for 5-user license (UNIX and PC) (1997)
Parlance Document Manager	Data management and workflow system for editorial workgroups; object layer built on Informix	DEC, UNIX, Windows client	Xyvision www.xyvision.com	$63,000 for 8 users (UNIX and PC) (1997)
SearchServer	Supports indexing of SGML documents	Windows	Fulcrum Technologies www.fulcrum.com	visit http:// www. fulcrum.com
Sybase	Verity search engine licensed for handling SGML and HTML	UNIX server, Windows client	Sybase, Inc. www.sybase.com.au	$499 (PC only), about $1,500 for a low-end UNIX version
WorkSMART	Data management and workflow; recently purchased by OpenText Corp.	DEC, UNIX, Windows client	InfoDesign Corp. www.idc.com	visit http:// www.opentext.com for pricing

*Check Web sites for most up-to-date information

Browser Tools

Browser software allows a user to view documents, usually in HTML format, on a computer screen without the SGML markup. Browsers format the documents in "WYSIWYG" style (what you see is what you get) for ease of viewing. Browsers also exploit the SGML feature of linking documents so that the user can navigate between documents easily, whether those documents are located on a Local Area Network or are scattered in different sites on remote servers. The largest share of the browser market is devoted to the presentation of HTML documents on the World Wide Web, but there are browsers that work with other DTDs—besides the HTML DTD—as well. The challenge in creating SGML browsers lies in the fact that it is difficult to build in support for arbitrary DTDs. The HTML browsers are all built around one DTD (HTML), and it is a relatively simple DTD at that. A fully robust SGML browser has to include support for any DTD. A fallback position is to build in support for one or a few select DTDs, but this would limit the browser's utility. Additionally other browsers are available for viewing files in alternate formats. An example is the Acrobat viewer from Adobe, which allows users to view PDF-formatted files they may find on the Web. The problem is currently handled by supporting "plug-ins" to the main browser. In this scenario, if a user selects an SGML file on the Web, the HTML browser would launch a separate SGML viewer to display that file. Similar plug-ins are available for PDF and other formats. The major browser makers are working on designs that will allow the main browser to display a larger variety of formats without launching a separate plug-in module.

A growing problem is caused by the various browsers adding support for non-standard markup to achieve special display effects. What works in one browser may not work in another browser, so the overall standards for the World Wide Web may eventually deteriorate into a battle for competing proprietary standards. As George Alexander and Liora Alschuler state in the Seybold Report, "In short, there was a brief period when standardization of HTML around SGML principles was a reachable goal. That goal is now rapidly receding. SGML partisans seem to be regrouping around two less desirable but more achievable goals: first, using SGML as a neutral format from which Web pages can be derived; and, second, using SGML directly on the Web in some cases." The fact that XML will emerge soon may provide an alternative.

DynaText

See **Publishing Tools,** below.

Internet Explorer

Internet Explorer is Microsoft Corporation's entry in the browser marketplace. In most ways similar to Netscape Navigator, the Internet Explorer distinguishes itself by supporting some of the newer HTML 3.2 features before its rival. The Internet Explorer also has many more keyboard shortcuts built into it than does the Netscape Navigator.

Mosaic

Mosaic was the first full-featured HTML browser available. Originally created by researchers at the National Center for Supercomputing Applications (NCSA), Mosaic had many of the features that are now standard in other browser applications—the point-and-click file navigation, the WYSIWYG viewing of documents, and the ability to handle multi-media files—just to name a few examples. Netscape Navigator has since eclipsed Mosaic as the *de facto* standard for HTML browsers. In fact, most of the senior designers who built Mosaic later went to work for Netscape, repeating and augmenting upon their earlier success.

Multidoc Pro

Multidoc Pro from Citec Information Technology allows users to build document sets from structured SGML files and/or fragments and view them in a browser window. It also supports, as does the Panorama product suite, simultaneous multiple views of the SGML documents.

Netscape Navigator

Netscape Communications, Inc. has, by far, the largest share of the of the HTML browser market. Besides its browser tool, the Netscape Navigator, Netscape also markets tools that aid in the creation and maintenance of World Wide Web sites. Netscape Navigator, and many other browsers, also support the integration of tools for viewing special files. For instance, sound files stored in the WAVE format can be linked to an HTML file. When a user clicks on the link, Netscape can invoke separate software of the user's choosing that can play the sound file through the computer's speakers. The browser can be extended to work with animation files—or any type of presentation file, for that matter—by tying in extra software. The other files are not HTML files, but they can be easily linked to the HTML files.

Panorama and Panorama Pro

Panorama, available for free on the World Wide Web, and Panorama Pro, which has a richer feature set, are SGML browsers created by SoftQuad. The browsers are capable of downloading an SGML document and its attendant DTD on the fly and then displaying the formatted document in the Panorama browser window. Typically, browsers like these are configured to be launched when an HTML browser links to an SGML file. In this way, the more powerful HTML browser can be used for viewing most Web documents, but the alternate browser can be launched on the fly whenever needed.

Softquad Explorer

See **Publishing Tools**, below.

WorldView

See table below.

Browser Tools Table*

Product Name	Notes	Supported Platforms	Vendor Name and Web Site	Unit Price
DynaText	SGML WEB publishing system with viewer; data stylesheet required	Mac, UNIX, Windows	Electronic Book Technologies www.ebt.com	$2,100 for 10 licenses (1995)
Internet Explorer	HTML browser; bundled with W95	UNIX, Windows	Microsoft Corp. www.microsoft.com	$699 (server license), client licenses free
Mosaic	HTML browser	UNIX, Windows	Several sites on the World Wide Web	Public Domain
Multidoc Pro	SGML browser	UNIX, Windows	Citec Information Technology www.citec.fi	Not published
Netscape Navigator	HTML browser	Mac, UNIX, Windows	Netscape Communications	$699 (server license) client free
Panorama	HTML and SGML WEB browser	Windows	SoftQuad Inc. www.sq.com	Public Domain
Panorama Pro	HTML and SGML WEB browser	Windows	SoftQuad Inc. www.sq.com	$195 (1996)
SoftQuad Explorer	SGML CD-ROM publishing system with viewer	Windows	SoftQuad Inc. www.sq.com	$9,995; $4,997 Academic License (1994)
WorldView	Distribution system with browser	Mac, UNIX, Windows	Interleaf, Inc. www.ileaf.com	$195 (PC only) (1997)

*Check Web sites for most up-to-date information

Publishing Tools

Publishing tools for SGML-tagged data come in three basic types: tools that are used to create paper output, tools that used to create files for electronic delivery on a network or on a CD-ROM, and tools that can deliver both print and electronic output. Some of the more common publishing tools are listed below.

Adept Publisher

The Adept Publisher from ArborText includes a style editor for defining FOSIs (Formatting Output Specification Instance). The formatting uses existing formatters and the final output can be previewed on screen. By automatically balancing page "fullness" with the need to keep related elements together, Adept Publisher produces attractive pages with no need for manual intervention or inspection. Adept Publisher saves its files in an SGML format, but uses PostScript as its output format. Additionally, the Adept Publisher product shares all the features found in the Adept Editor editing product (see description below in **Editing Tools**).

DynaText

DynaText is, according to Electronic Book Technologies' marketing materials, "an electronic book publishing system that accepts ISO 8879 Standard Generalized Markup Language (SGML) text directly, allowing in-house publication groups to turn existing reference documentation into dynamic electronic books—cost-effectively." The DynaText system consists of two basic components—an SGML-aware indexer and a browser. The Indexer is an electronic book builder that can index a series of SGML documents, creating hyperlink markers throughout the resultant output file so that users can easily navigate through the electronic book. DynaText's browser is tailored to work well with its indexer's output.

FrameMaker+SGML

Adobe Systems Inc. has added a comprehensive set of tools to its publishing product by adding the ability to use SGML in FrameMaker, its versatile document-publishing program. Adobe's product integrates SGML capabilities with FrameMaker so that those who already use FrameMaker or FrameBuilder can edit SGML documents in the Frame product. The product, however, stores data in a non-SGML compliant file.

Softquad Explorer

Softquad's Explorer Publisher and Explorer Viewer are similar to the Dynatext product, in that these tools facilitate the building of "electronic books" from a variety of source files and provide a viewer to browse the material. SoftQuad makes the distinction in its marketing literature that the Explorer software is meant to be used for developing CD-ROM products, while its Panorama software (described in **Browser Tools,** above) is for World Wide Web browsing.

Xyvision Parlance Publisher

See table below.

Publishing Tools Table*

Product Name	Notes	Supported Platforms	Vendor Name and Web Site	Unit Price
Adept Publisher	DTD design, compilation; FOSI design, validation; includes Adept Editor	UNIX	ArborText, Inc. www.arbortext.com	$4,950 (UNIX only)
DynaText	SGML WEB publishing system with viewer; incorporates Balise	Mac, UNIX, Windows	Electronic Book Technologies www.ebt.com	$2,100 for 10 licenses (1995)
FrameMaker + SGML	WYSIWYG editing and publishing tool	Mac, UNIX, Windows	Adobe Systems Inc. www.adobe.com	$1,495 - $1,995 (1996)
SoftQuad Explorer	SGML CD-ROM publishing system	Windows	SoftQuad Inc.	$9,995; $4,997 Academic License (1994)
Xyvision Parlance Publisher	OmniMark input filter	UNIX	Xyvision Inc. www.xyvision.com	$63,000 for 8 users (UNIX and PC) (1997)

*Check Web sites for most up-to-date information

Editing Tools

A wide variety of tools are available for editing SGML documents. Most try to present the SGML information in a way that is not too difficult for users to work with. Some hide the SGML markup from users altogether, relying on displaying type in different sizes, colors, and spatial placement depending on which SGML elements are on the screen. Some check the validity of SGML-tagging in real time, as users enter the data. Others have a validation step that must be executed at the end of an editing session. Finally, some editors store and work with SGML in its native format, while others work with a proprietary format and have export programs to output SGML files. Editors that store files in a proprietary format generally do not support as much of the SGML standard as editors that store files in native SGML.

Adept Editor

Adept Editor from ArborText is one of the few powerful SGML editors that store SGML files in their native SGML format. Adept Editor has the added capabilities to edit tables and equations. SGML markup is visible to a user editing text, but tags can be hidden on user request. Select elements can be set so that the text in them cannot be changed. New documents can be created with a ready-made template of tags for the user to fill in. Adept Editor has a built-in SGML parser that parses the document in real time as it is being edited, but this parser also can be turned off for editing in a non-parsing mode. Adept Editor also has a powerful command language, which allows users to fully customize the keyboard and menus. Additionally it has a robust set of API functions that enable integrators to tightly link the Adept Editor application to other SGML applications, document management systems, and databases.

Author/Editor

Author/Editor from SoftQuad is an SGML editor that validates the document as it is being edited. An element list is used to pick elements to insert. The representation on screen is done with a style editor, which generates style files. Author/Editor also allows users to expand a document's multi-media capabilities by launching external applications from graphics and sound to mathematical equations, chemical formulas or any non-SGML format.

Framemaker+SGML

See **Publishing Tools**, above.

Grif SGML Editor

The Grif SGML Editor, designed by the French Grif company, supports WYSIWYG editing of native SGML documents, contextual menus to guide authors through the editing process, and interactive parsing to ensure respect of document structure at all times.

HotMetal and HotMetal Pro

SoftQuad's HotMetal is the premiere HTML editing tool. HotMetal has a powerful validating editor, which allows users to change tag attributes, create long documents, and perform some final adjustments to graphics. HotMetal's user interface is also a cut above those in most Web-authoring tools.

InContext Spider

InContext Spider from InContext Systems is another HTML-only editor. It provides a structural view of a document that helps users create documents which are correct in form and style. Its link manager aids users in establishing links between documents.

Near & Far Author

Near & Far Author from Microstar is another Word for Windows add-on product. It saves documents in the proprietary Word format, but has the capability to export documents in an SGML format.

psgml

Psgml is a shareware package that works with the latest versions of the UNIX-based GNU EMACS text-editing program. It includes an SGML parser. It offers several menus and commands for inserting tags with only the contextually valid tags, identification of structural errors, editing of attribute values in a separate window with information about types and defaults, and structure-based editing.

SGML Author for Word

Microsoft's SGML Author for Word add-on package has the same advantage that the SGML Edition of WordPerfect enjoys — a large user base that already knows how to use the Word product. It has the same shortcomings as well — files stored in a proprietary format and no interactive validation.

TagWizard

TagWizard, manufactured by the French company, Nice, is a Word for Windows add-on package that supports the insertion and validation of SGML tagging. One shortcoming of the program is that it does not use an SGML doctype declaration at the beginning of a document file, making it difficult to import files processed with TagWizard into other SGML applications.

WordPerfect SGML Edition

The SGML Edition of WordPerfect is an add-on the regular WordPerfect application. It takes advantage of the fact that a base of WordPerfect users might not want to switch to another text processing package when they switch to authoring in SGML. By choosing WordPerfect, sophisticated users can write macros that can automate some SGML element insertion tasks. The application does not store files in an SGML format and it does not interactively validate (parse) files that are currently being edited.

Editing Tools Table*

Product Name	Notes	Supported Platforms	Vendor Name and Web Site	Unit Price
Adept Editor	Includes table and equation editor interface; HTML support; DTD design, formatting add-ons; entity reference management	OS/2, UNIX, Windows	ArborText Inc. www.arbortext.com	$1,350 (PC) $2,950 (UNIX) (1997)
Author/Editor	Rules Builder for DTD compilation	Mac, UNIX, Windows	SoftQuad Inc. www.sq.com	$995-$1,495 (1995)
FrameMaker + SGML	Exports PDF, HTML; link and entity reference management	UNIX, Windows	Adobe Systems, Inc. www.adobe.com	$1,495 (1996)
Grif SGML Editor	API for composition formats	UNIX, Windows	Grif S.A. www.grif.fr	$4,750 (1994)
HotMetal validation	HTML editing tool; interactive	UNIX, Windows	SoftQuad Inc. www.sq.com	HotMetal- Free, HotMetal Pro $159
InContext Spider	HTML editing tool; link manager; uses Microsoft Excel for table editing	Windows	InContext Systems www.incontext.ca/	$99 (1995)
Near & Far Author	Word for Windows add-on; stores files in Word format; SGML export capability; supports composition	Windows	Microstar Software Ltd. www.microstar.com	$275 (1996)
psgml	UNIX-based GNU EMACS text editor	DOS, UNIX	www.lysator.liu.se	Public Domain
SGML Author for Word	Word for Windows add-on; requires customized macros; no interactive validation; integrates add-ons for conversion	Windows	Microsoft Corp. www.microsoft.com	$495 (1996)
TagWizard	Word for Windows add-on; product may be withdrawn from market	Windows	Nice Technologies info@nicetech.com	$180 (PC only) (1995)
WordPerfect SGML Edition	WordPerfect add-on; DTD editor; requires customized macros; no interactive document validation	Windows	Corel Corporation www.corel.com	$175 (1996)

*Check Web sites for most up-to-date information

Appendix 4: A Vendor Looks at Cost Metrics

Following is an analysis of the questions to be considered when making the choice between PDF, HTML, and SGML, provided by P.G. Bartlett, vice president of marketing for ArborText, Inc.

Consider the answers to the following questions:

1. What is the maximum level of reuse of information that you could achieve? What's your current cost for creating and maintaining duplicate information? What's the cost to set up processes and train your people to use a modular approach to information to eliminate redundancy and increase efficiency?

   [We've seen companies that expect their new documents to represent as much as 80% "old" information with systems where less than 20% of the information is reused. Those companies can achieve huge savings by eliminating all that redundancy, both in the creation and in the maintenance of their information.]

2. How much time do your authors currently spend on formatting?

   [We've seen improvements in author productivity of between 30% and 100% by freeing authors from the "paragraph squeezing" that they do to create attractive page breaks.]

3. What kind of document structure do you need? Can you use a readily available DTD such as DocBook or do you need your own DTD? If you need your own, how many tags do you expect to use?

4. What kind of outputs do you need? Do you need printed and electronic versions? If you need printed versions, what kind of styling do you need? Do you have very specific and detailed requirements for the appearance of your printed pages, or do you have a lot of flexibility?

5. What opportunities exist for automating the creation, review, maintenance, and distribution of your information? Where are the labor-saving opportunities?

6. What is the demand for repurposing your existing information to create new information products? To the extent that an SGML-based system allows you to create new products that would otherwise be prohibitively expensive, you gain additional advantage (Bartlett, November 1996).

The Commission on Preservation and Access
A program of the Council on Library and Information Resources
1400 16th Street, NW, Suite 740
Washington, DC 20036-2217