LESSONS FOR THE WORLD WIDE WEB FROM THE TEXT ENCODING INITIATIVE

David T. Barnard, Lou Burnard, Steven J. DeRose, David G. Durand, C.M. Sperberg-McQueen

Abstract

Although HTML is widely used, it suffers from a serious limitation: it does not clearly distinguish between structural and typographical information. In fact, it is impossible to have a single simple standard for document encoding that can effectively satisfy the needs of all users of the World Wide Web. Multiple views of data, and thus multiple DTDs, are needed.

The Text Encoding Initiative (TEI) has produced a complex and sophisticated DTD that makes contributions both in terms of the content that it allows to be encoded and in the way that the DTD is structured. In particular, the TEI DTD provides a mechanism for describing hypertextual links that balances power and simplicity; it also provides the means for including information that can be used in resource description and discovery. The TEI DTD is designed as a number of components that can be assembled using standard SGML techniques, giving an overall result that is modular and extensible. ***Keywords:*** *SGML, modular DTDs, extensible DTDs, linking mechanisms, header*

Introduction

The World Wide Web is growing with amazing rapidity, and with it, HTML (Hypertext Markup Language) document encoding. However, even in the presence of this success, there are problems which are evidenced by the frequent, and frequently bitter, divisions over HTML style and the conflicting approaches to extending HTML. These divisions are caused, to a great extent, by the fact that HTML has an underlying confusion of categories that leads to abuse and misuse of tags. Or, perhaps more correctly, to different uses and interpretations of HTML, based on different priorities. These conflicts reflect the fact that HTML is partly a markup scheme for structural markup, and partly a scheme for presentational markup; these two tendencies are at war both in the HTML specification and in the usage of document publishers and software developers.

Although at its inception this was not true, HTML is now defined as an application of SGML (Standard Generalized Markup Language). SGML is a metalanguage for defining document markup; it is defined by an international standard [8], and there is a handbook that interprets the standard [6]. Even more information about SGML can be found in the World Wide Web page maintained by Robin Cover [4]. SGML allows the definition of a markup language applicable to a set of documents by specifying the components that the documents will contain, the ways in which components can be combined together to make larger components and entire documents, and the ways in which the boundaries of components will be indicated in the document.

The information added to a document to delineate the components is called markup. The various parts of the formal specification of a document class are gathered together in a *document type definition* (DTD). For example, a simple DTD for office memoranda might include definitions for a *heading* and a *body*, with the heading including *to*, *from*, *date*, and *subject* components and the body containing *paragraph* components.

A component is (usually) delineated by preceding it with its name in angle brackets and following it with its name preceded by a slash in angle brackets, as in

```
<heading> ... <subject>Salary Policy</subject> ... </heading>
```

HTML is now formally defined as an application of SGML. This means that a DTD defines the components of HTML documents, and their possible hierarchical relationships [2]. Future versions of HTML promise to be tied to the formal SGML setting in increasingly explicit ways.

Although it makes concessions for the encoding of processing information—such as layout commands—SGML is designed to allow systems to focus on the structure of documents, to precisely describe what is present, rather than how it will be processed. In the document-processing model adopted by SGML, the description of document formatting (or any other processing) is consciously and explicitly separated from the description of document structure.

The same claim cannot be made for HTML. It contains structural concepts, such as the <P> tag to describe a paragraph. But the Web still bears visible traces of the first version of HTML, in which the paragraph was not, strictly speaking, a structural unit that was contained in some units and could contain others. Instead, as commonly implemented, the paragraph tag indicates a point at which specific processing is to occur. HTML also contains tags for such typographic features as images (with alignment constraints to control a formatting process), horizontal rules, and type styles. Perhaps the most extreme example of nonstructural encoding in some network documents is an HTML extension indicating that text is to blink when presented on the screen—a formatting indication that does not even have a meaning if the document is to be printed.

Of course, the most obvious, perhaps most frequent, and design-anticipated use of documents encoded in HTML is to display them on a screen with a network browser. And it is not surprising that this intended application should be—or, at least should still be—implicit in the document encoding. But this means that even users who would prefer to use a structural encoding cannot do so. Absent (at the moment) style sheets for mapping structural categories to display characteristics, users frequently resort to "tag abuse"—using existing tags for their typographical effects rather than for their structural significance, if any. In fact, "tag abuse" is possibly the most common style of markup on the Web, especially given the needs of the commercial users now flocking to the Internet.

The Text Encoding Initiative (TEI) is a large international project sponsored by the Association for Computers and the Humanities, the Association for Computational Linguistics, and the Association for Literary and Linguistic Computing. The project began at a planning meeting late in 1987, which was attended by researchers involved in encoding texts for research purposes (such as the production of critical editions and linguistic analysis) and in producing software to deal with encoded texts. There was agreement among the participants that the chaotic diversity of encoding techniques in use made it needlessly difficult to share texts, software, and research results among colleagues.

At the meeting, ACH, ACL, and ALLC agreed to sponsor a project to develop a common standard for encoding texts of interest to the communities they represented (humanistic researchers, linguists, and others involved in "language industries"). They supported the project by providing members for a Steering Committee and raising funds for the development work. Over the next several years, the U.S. National Endowment for the Humanities, Directorate General XIII of the Commission of the European Communities, the Andrew W. Mellon Foundation, and the Social Science and Humanities Research Council of Canada all provided funds.

The project's design goals were that the Guidelines should:

- Define a standard format for data interchange
- Provide guidance for encoding texts in this format
- Support the encoding of all kinds of features of all texts studied by researchers
- Remain application independent

These goals led to a number of important design decisions, such as:

- The choice of SGML
- The provision of a large predefined tag set
- A distinction between required, recommended, and optional encoding practices
- Encodings for different views of text
- Alternative encodings for the same features
- Mechanisms for user-defined extensions to the scheme

The work of the project was carried out by scholars at institutions in North America and in Europe. The main result of the project is a document entitled *Guidelines for Electronic Text Encoding and Interchange (TEI P3)*, edited by Sperberg-McQueen and Burnard [10]. This large document (almost 1300 pages) describes a collection of SGML tag sets that together make up a modular and extensible DTD, with which one may encode a wide range of documents.

The Guidelines can be found online in several places. The official project repository, containing the Guidelines and other project documents, is at *ftp://ftp-tei.uic.edu/pub/tei* (for users in North America) and its mirror sites *ftp://ftp.ifi.uio.no/pub/SGML/TEI* (for users in Europe) and *ftp://TEI.IPC.Chiba-u.ac.jp/TEI/P3* (for users in Asia), or at *ftp://info.ex.ac.uk/pub/SGML/tei*. A searchable form is available via the World Wide Web at *http://etext.virginia.edu/T EI.html*; another Web form may be found at *http://www.ebt.com/usr-books/t eip3*.

The entire volume of *Computers and the Humanities* for 1995 is devoted to the TEI; the papers in that volume contain references to other TEI-related articles. In particular, the general papers in that volume are a good introduction to the project [7,11], and there is an introduction to SGML from the perspective of the project [3]. Although SGML has served the TEI well, we have identified some ways in which SGML could be improved [1].

The TEI is possibly the largest DTD created to date. And with world literature, dictionaries, and literary and linguistic analysis as its core concerns, it certainly covers the widest range of documents of any encoding standard. In the remainder of this paper we show how the creation of the TEI guidlines provides results that furnish key insights into the use of documents and document-encoding standards on the World Wide Web.

Using Multiple DTDs

It became clear early in the work of the TEI that a single comprehensive DTD that could encode every feature of interest to the communities contributing to the project would be so large as to be impossible to understand, and doubtless impossible to design. (Debates over HTML 3 suggest the same is true of a single DTD supporting all users of the Web.) Further, users of TEI documents are often interested in several views of a document at the same time, so that in effect multiple DTDs were required in any case.

As a result, the TEI DTD has been designed in a modular fashion. A particular document will use only those pieces of the DTD that apply to it. The selection of pieces to include is done using standard SGML mechanisms, so it can be specified to an SGML parser with minimal manual intervention and no additional software tools.

Further, the TEI DTD is extensible. Users can add other modules to it, again using standard SGML mechanisms. These extensions can be communicated to an SGML parser—and thus obviously to

other users—in a formal manner, so that the extensions can be specified and documented as fully as the basic DTD. The need for extensibility is a direct consequence of the richness and open-endedness of the application areas for electronic documents. No language with a finite vocabulary can ever hope to suffice for electronic documents in the long run. In spite of the considerable amount of effort that has gone into designing the TEI DTD, there will inevitably be uses for which it is not well suited and forms of information that cannot be conveniently encoded using its structures. Our approach to dealing with this has been explicitly to provide an extension mechanism.

The modular structure of the TEI DTD groups SGML elements into the following categories:

Core tag sets

Describe standard components of documents; they are included in all forms of the DTD. These include such things as paragraphs, lists, simple links and cross references, highlighting, and quotation, which are all familiar to users of HTML. The core tag set also includes tags for notes, indexes, bibliography entries, names, numbers and dates, and other commonly encountered textual phenomena.

Base tag sets

Include the basic structures needed for describing a specific text type. Usually one of these is selected for a given document, although there are ways to use several of them together for complex documents. There are base tag sets for prose, for verse, for drama, for print dictionaries, for the transcription of spoken material, and for terminological databases.

Additional tag sets

Define extra tags that are used for specific purposes. They are compatible with all the bases and with each other. Any combination of these tag sets can be used in a single document. At present, additional tag sets are defined for linking, segmentation, and alignment; encoding simple analytic mechanisms (linguistic segments); encoding critical apparatus associated with a text; handling graphics and tables; and several other purposes.

Documents explicitly indicate which extensions to the TEI DTD they use by identifying a base tag set and additional tag sets. The core tag sets are implicitly present, because they are included by the base. In a TEI document, a document parser is therefore able to check modifications to the DTD using standard SGML mechanisms, and the formal notation also serves the purpose of providing inline documentation of required changes to the defaults. The modifications are made possible by maintaining two versions of the DTD. There is a version for people to read, which is the version documented in the *Guidelines*. There is also a version for parsers to read; this version is derived programmatically from the first one by the introduction of SGML parameter entities for various purposes. Modifications to the DTD are made by changing the values of parameter entities, thus changing the DTD that is expanded in the parser.

The TEI DTD supports the following modifications:

- Deleting an element. An element defined in the TEI DTD can be suppressed so that it cannot be used in the document. An SGML parser will detect all uses of the tag as an error.
- Renaming an element. This can be used to rename the tags in a language other than English or to use local vocabulary within a project or collection of documents.
- Extending given classes. There are several predefined classes of tags in the TEI DTD. These classes typically share a set of attributes and thus can be treated in similar ways by applications. A particular document can specify that a tag is to be included in one of these classes.

- Specifying new content models. If the definition of what goes in an element is not sufficient for what needs to be expressed in a document, the element can be deleted and a new definition given. By introducing new names at this point, it is possible to extend the DTD with new tag sets for new applications.

It would be possible to use the parameter entity mechanism for other purposes as well, such as changing attribute names, redefining existing attributes, changing the inclusion and exclusion exceptions for an element, and so on. The set of modification possibilities given here was considered to be sufficient for most of the things that users claimed they needed to do.

The experience of the TEI in designing a complex DTD leads to several conclusions relevant to the World Wide Web community. First, a single fixed DTD, no matter how well it is designed, can never serve all users equally well. Users must have ways to specify structures not anticipated at DTD design time. Second, it is possible to design DTDs—or DTD families—that are modular and extensible. The TEI tagsets demonstrate one method of doing so. Third, a rich set of structures can already be described with the existing TEI DTD, and it can thus already be used for a rich variety of applications. We encourage readers to consider it for their applications.

We now turn to two specific content areas addressed by the TEI DTD that demonstrate helpful ways to use SGML for encoding information of value in World Wide Web applications. These are the specification of hypertext links and the description of documents and their contents.

Linking Mechanisms

The World Wide Web has grown because of its simplicity. In particular, the concept of a Uniform Resource Locator (URL) is a simple one: a text string provides an address of a location in a file on a machine on the network. However, the simplicity that contributes to rapid growth is limiting. URLs cannot locate a portion of text or a substructure in a document, they cannot easily specify how links might be related in sets, and they cannot specify any semantics to be associated with a link.

Another approach for specifying hypertext links is to use the HyTime standard [9] (the book by DeRose and Durand contains a description of HyTime [5]). HyTime does not suffer from being too simple. It is, in fact, very powerful; it allows for very general cases of hypermedia links to be specified. Links can be separated from objects (documents), complex relationships can be specified, coordinate systems can be defined, and parts of documents selected based on those coordinate systems, and so on.

In our view, URLs as they stand are too simple to meaningfully encode many of the structures that are common in and among documents on the World Wide Web (though they are perhaps adequate to implement most of these). One the other hand, HyTime provides (and requires) a more powerful mechanism than many applications will need. The TEI linking mechanisms provide what seems to us a better balance between simplicity and power.

The TEI DTD provides linking mechanisms for several different kinds of structure. Simple links within a document are formed using the SGML "id" and "idref" mechanism. Links between documents, or links within a document to locations which bear no ID attribute, are provided through *extended pointers*. These latter exist in two different forms:

- The <xpTR> tag provides a pointer to another location, either in the current document or some other document.
- The <xref> tag allows the inclusion of textual commentary with the specification of the pointer.

While these extended pointers build on the SGML id and idref mechanism, they are specified by giving strings as the values of attributes of

SGML tags. Like HTML tags and URLs, these strings need to be interpreted by application software that understands their significance.

The TEI's extended pointers allow links to be specified in terms of:

- Hierarchical references to structures in a document (in much the same way that files can be named in a hierarchical file system)
- More general structural relationships (such as the identification of the "next" node with a given generic identifier, which is to be found by a simple, clearly specified rule about tree traversal)
- Locations that are defined relative to the node making the reference
- Patterns that are to be applied when the link is traversed or activated
- Queries that are related to HyQ, the HyTime query language

We will not give the details of extended pointers here. These can be found in the *Guidelines*. What is of interest here is the kinds of structures that can be easily encoded using the mechanisms provided by the TEI DTD. Here are some examples.

- A *segment* is a portion of a document. It can be used as the point of attachment of a link. Any arbitrary structure can be defined as a segment.
- An *anchor* is an arbitrary point in a document. It can be used as the point of attachment of a link. (This is similar to the definition of a name on an anchor in HTML.)
- A *correspondence* can be established between one span of content and another. For example, there might be a correspondence between a fragment of a document, and someone's comments on that fragment.
- An *alignment* shows how two documents (or fragments) are related. For example, there could be an alignment between a document in one language and another document that is the translation into a second language. An alignment can be specified in a document outside the two documents (or fragments) that are to be aligned.
- A *synchronization* is a relationship that represents temporal rather than textual correspondence. For example, it is often necessary to synchronize overlapping text segments in a representation of speech where several speakers can be talking at the same time.
- An *aggregation* is a collection of fragments into a single logical whole. For example, the set of passages in a document relating to a specific topic, such as the set of paragraphs that discuss indexing in a paper on information retrieval, would be an aggregate.
- *Multiple hierarchies* occur, essentially, when more than one tree is to be considered as being built over the same textual frontier. For example, the logical structure of a document (chapters, sections, paragraphs) and its physical structure (pages, lines) are two different hierarchies over the same frontier. Although the SGML CONCUR feature can be used to specify structures of this sort, it has a number of associated problems: when a document is changed by the addition of a new view, it may be necessary to change existing markup (by the addition of a prefix indicating the view to which the existing tags correspond); the coding of tags becomes more verbose than otherwise, and many SGML applications at present do not implement the feature. There are tags provided to specify page and line boundaries, and thus in a rudimentary way to provide for this second commonly required hierarchy. The more general approach used is to mark boundaries of the elements in the multiple hierarchies and to reconstitute the view, essentially by using aggregates.

These structures that have been identified by participants in the TEI as useful ones for encoding documents for research purposes seem to us to be useful in many other contexts in the World Wide Web as well. The TEI DTD provides mechanisms for encoding these structures in relatively straightforward ways. These mechanisms could be used without having to provide all of the processing power in Web application software that is required to process HyTime.

Resource Identification and Discovery

The World Wide Web contains many documents in many locations. One of the major challenges in a complex distributed environment like this is the identification and discovery of documents that are relevant to some task. In a traditional library, resources are identified by the preparation of catalog information in a restricted but rich and dynamic domain of categories. Identifying relevant resources often involves the expertise of the person who needs information, various programs that have access to catalogs for relatively simple searches, and experts in the domain of interest (subject librarians). While the search techniques applied to catalogs are relatively simple, the catalogs contain explicitly coded information about subject areas so that searches are usually able to identify a useful collection of materials.

Information retrieval in collections of electronic documents similarly involves the expertise of the person who needs information, sophisticated search programs, and sometimes experts in the domain (subject librarians). Information can be labeled with various category attributes, but larger amounts of text (abstracts, and perhaps complete documents) can be searched. Because there is little or no explicit encoding of the information in the text, sophisticated algorithms are often used to attempt judgements about relevance of a document based on the occurrences of patterns in the text.

Identifying relevant resources on the World Wide Web can take several forms. It can involve searching through structured subject indexes as in traditional library access, as well as searching through the text of documents as in traditional information retrieval.

But because the Web contains so many documents—orders of magnitude more than most databases used with traditional search strategies—identifying relevant resources can be difficult. It would seem attractive to allow documents to describe themselves so that a rich domain of categories can be used and so that judgments about relevance do not need to be restricted to algorithmic approximations.

Documents encoded according to the TEI DTD must include a *TEI header* that contains information about the electronic document. The information in the header can be used to facilitate the identification of resources and their discovery by search programs and by manual browsing.

The header has four major parts:

- A *file description* contains a full bibliographical description of the electronic document. A standard bibliographic citation can be derived from this information, so it could be used to make a standard library catalog record. This part of the header also includes information about the source of the electronic document (for example, the document may be appearing originally in electronic form, it may be transcribed from a printed form, and so on).
- An *encoding description* describes the relationship between the source and the electronic document. This part of the header can describe any normalizations applied to the text, the specific kinds of analytic encoding that have been used, and so on.
- A *text profile* contains information that classifies the text and establishes its context. This part of the header describes the subjects addressed, the situation in which the text

was produced, those involved in producing it, and so on. This part can be used with a fixed vocabulary of subjects, for example, to catalog texts into some predefined subject structure. or it can be used more freely to allow a dynamic subject universe.

- A *revision history* allows the encoding of a history of changes made to the electronic document. This part of the header is useful for the identification and control of versions of a document.

Each part of the header is potentially complex, and can contain extensive amounts of information. Most parts of the header are optional, though, so exhaustive cataloging is not required. These fields need only be used when they are considered useful or necessary by document developers. A minimal header contains a file description including a title, publication statement, and source, together with a text profile identifying the language in which the document is written.

To take best advantage of the mass of information that is available on the Web, users must be able to find the documents that are relevant when they are looking for information. The best way to facilitate this is to have documents identify and describe themselves.

The TEI header is an example of how documents can be made to be self-identifying. Documents with a developer-created header can be indexed in the ways that are considered to be appropriate by their developers. The information that is provided can be used by readers of Web documents and by programs that search the Web to identify relevant resources for readers.

Conclusion

The World Wide Web is based on a set of simple tools and concepts, including HTML, that have made possible a phenomenal rate of acceptance and growth. These simple notions, though, will not be sufficient to support continued growth and a diversity of applications.

There are various ways in which full SGML can be provided on the Web, including server-side processing (such as mapping more complex structures to HTML for delivery to clients) and client-side processing (such as spawning applications that are capable of dealing with general SGML DTDs or a specific DTD).

The Text Encoding Initiative has developed a comprehensive specification for a DTD that provides a richer set of structures in a modular extensible framework. The DTD itself, together with its structuring principles and the specific contributions for hypertext links and for resource description, suggest fruitful approaches to developing and enhancing the World Wide Web. ■

References

1. Barnard, David T., Burnard, Lou, and Sperberg-McQueen, C.M., *Lessons Learned from Using SGML in the Text Encoding Initiative, Computer Standards and Interfaces* (accepted February 1995). Also appeared as Technical Report 95-375, Department of Computing and Information Science, Queen's University (1995).
2. Berners-Lee, T., and Connolly, D., *Hypertext Markup Language-2.0, <draft-ietf-html-spec-06.txt>*, Boston, HTML Working Group, September 1995.
3. Burnard, Lou, *What Is SGML and How Does It Help?, Computers and the Humanities 29,1*, 1995, 41-50.
4. Cover, Robin, *SGML Web Page, http://www.sil.org/sgml/sgml..html*, 1994.
5. DeRose, Steven J., and Durand, David G., *Making Hypermedia Work: A User's Guide to HyTime*, Boston/Dordrecht/London, Kluwer Academic Publishers, 1994.
6. Goldfarb, Charles, *The SGML Handbook*, Oxford, Oxford University Press, 1990. Contains the full annotated text of ISO 8879 (with amendments).
7. Ide, Nancy, and Sperberg-McQueen, C.M., *The Text Encoding Initiative: Its History, Goals, and Future Development, Computers and the Humanities 29,1*, 1995, 5-15.
8. ISO (International Organization for Standardization), *ISO 8879-1986 (E) Information Process-*

ing—Text and Office Systems—Standard Generalized Markup Language (SGML), Geneva, International Organization for Standardization, 1986.

9. ISO (International Organization for Standardization) *ISO/IEC 10744:1992 Information Technology—Hypermedia/Time-based Structuring Language (HyTime)*, Geneva, International Organization for Standardization, 1992.
10. Sperberg-McQueen, C.M., and Burnard, Lou (eds.), *Guidelines For Electronic Text Encoding and Interchange (TEI P3)*, Chicago and Oxford, ACH-ACL-ALLC Text Encoding Initiative, May 1994, 1290 pages.
11. Sperberg-McQueen, C.M., and Burnard, Lou, *The Design of the TEI Encoding Scheme, Computers and the Humanities 29,1*, 1995, 17-39.

About the Authors

David T. Barnard
[*http://www.quc is.queensu.ca/home/barnard/info.html*]
Queen's University, Kingston, Canada

David T. Barnard joined the Department of Computing and Information at Queen's University in 1977, having studied at the University of Toronto. He is now Professor in that Department. His research applies formal language analysis to treating documents as members of a formal language, and to compiling programming languages with a focus on using parallel machines. He chaired one of the working committees of the Text Encoding Initiative, and is now a member of the Steering Committee of the project.

Lou Burnard
Oxford University Computing Services, Oxford University, England

Lou Burnard is Humanities Computing Manager at Oxford University Computing Services. His responsibilities include the Oxford Text Archive, which he founded in 1976, and the British National Corpus. He is also European editor of the Text Encoding Initiative, and coauthor of a report proposing the establishment of a networked UK Arts and Humanities Data service.

Steven J. DeRose
Senior Systems Architect, Electronic Book Technologies, Inc.

Steven J. DeRose is one of the founders of Electronic Book Technologies. He holds a Ph.D. in Computational Linguistics and has published and spoken widely on descriptive markup, hypermedia, natural language processing, information retrieval, artificial intelligence, and other topics. He has consulted on commercial projects in related fields since 1982, and is active in several standardization efforts through organizations including TEI, SGML Open, IETF, ANSI, and ISO.

David G. Durand
[*http://cs-www.bu.edu: 80/students/grads/dgd/*]
Computer Science Department, Boston University

David Durand is a doctoral candidate at Boston University, working on collaborative editing in hypertext systems. He served on the TEI committees on Metalanguage and Syntax and Committee on Hypertext. He is also a Senior Analyst at *Dynamic Diagrams* working on analysis of Web documents for visusalization and navigation, and the integration of the Web with SGML-based publication processes.

C.M. Sperberg-McQueen
[*http://www-tei.uic.edu/~cmsmcq/*]
University of Illinois at Chicago

C. M. Sperberg-McQueen is a senior research programmer at the computer center of the University of Illinois at Chicago. He currently works in the Network Information Services group. He was trained in Germanic philology in the U.S. and Germany, and is a member of the Association for Computers and the Humanities, the Association for Literary and Linguistic Computing, and the Association for Computational Linguistics. Since 1988 he has been editor in chief of the ACH/ACL/ALLC Text Encoding Initiative.