Book review: *README.1ST: SGML for Writers and Editors*

Lynne A. Price

Ronald C. Turner, Timothy A. Douglass, and Audrey J. Turner, *README.1ST: SGML for Writers and Editors*. Prentice-Hall, Upper Saddle River, NJ, 1996, 241 pages + diskette, ISBN 0-13-432717-9.

The Standard Generalized Markup Language (SGML) is an increasingly popular foundation for text processing applications such as editing, formatting, archiving, and interchanging documents as well as managing document databases. While the most widely known application of SGML is HTML, the Hypertext Markup Language used to prepare documents for the Internet's World Wide Web, SGML predates the Web and is used in many other contexts as well. U.S. government agencies that use SGML include the Internal Revenue Service and the Department of Defense, while industry-wide applications of SGML exist in the aircraft, telecommunications, semiconductor, and computing communities.

The essence of SGML is that a document consists of a hierarchy of structural elements, each containing a sequence of other elements or text — a book consists of a title followed by a sequence of chapters; each chapter has a title followed by paragraphs that might be interspersed with itemized lists, figures, and tables; the paragraphs contain text, emphasized phrases, and new terms. The author using an SGML-based word processor, even a what-you-see-is-what-you-get word processor, is not required to specify the format of each element in the document; formatting is applied automatically based on the context in which the element occurs. Writing with SGML thus resembles using a well-controlled TEX macro package in which the only control sequences an author uses are those that indicate the start and end of elements. In fact, using TEX and SGML together is common. This reviewer wrote about the combination in the July 1987 issue of *TUGboat* and spoke about using TEX as a back-end in an SGML-based document production system at the TEX Users Group Annual Meeting in 1988.

SGML does not define the possible elements that can occur in a class of documents. Instead, each SGML document includes a document type definition or DTD that defines the possible elements that can occur, as well as the contexts in which each element is permitted. The DTD is followed by a document instance that contains data and markup that indicates how the document instance conforms to the DTD; the bulk of this markup consists of tags indicating the start and end of structural elements. The DTD lets software report errors if information is missing or out of place in the document instance. SGML software can also anticipate required elements to guide the user in creating document instances that conform to the DTD.

SGML is defined in International Standard ISO 8879:1986 Information Processing — Text and office systems — Standard Generalized Markup Language (SGML) and its amendment, ISO 8879:1986/A1:1988. When this standard was originally published in 1986, its terse and formal prose was the only source of information on SGML. In the years since, however, several ways to learn about SGML have evolved. There are SGML conferences and tutorials. Both public and commercial SGML software is now available. Several books describing SGML have been written. Prentice-Hall is now publishing a series of books, edited by Charles Goldfarb, editor of ISO 8879, on open information management. These books deal with managing information — particularly textual data — so that it can be processed by software beyond that planned when the data were created. SGML is a key part of this strategy. In addition to tutorials on SGML and related standards for the technical reader, the series addresses business-related topics such as justification for open information management and commerce in electronic information. The first volume in the series to appear is *README.1ST: SGML for Writers and Editors* by Ronald C. Turner, Timothy A. Douglass, and Audrey J. Turner.

In his foreword to this book, Goldfarb observes it is "easy and fun to read." This reviewer agrees. It is a fun book. It is well written and presents ideas in a logical progression. It is not, however, a textbook on SGML. Rather, it is a textbook on information management. It presents the motivation, philosophy, and history that led to the development of SGML. It describes the value of standards, the benefits of open-ended document processing, and the advantages of formal tagging. In the process, highlights of SGML are introduced where appropriate in the context of the ongoing discussion.

The book begins with background, discussing writing with standards, the SGML paradigm, and features and benefits of the language. The topic next switches to document analysis, as the authors explore the structure of a short sample document to determine the structural elements that occur within it and the contexts in which those elements occur. After a discussion of document type definitions and their use, this analysis is then used to build a DTD which the authors call WAE for "Writers and Editors". The next several chapters present major aspects of DTDs, gradually extending the WAE DTD as new concepts are introduced. The ideas of element declarations, entities, attributes, hypertext, and marked sections are gradually developed. The attribute chapter, for instance, gives several practical examples of attributes, instead of the one or two quick examples given in some other expositions of SGML.

Once this SGML foundation is established, the authors test it by walking the reader through a reading of a more complex DTD. The one they select is the publicly available DocBook DTD, intended for software documentation. The book concludes with a discussion of HTML as an SGML application and a quick introduction to hypermedia applications of SGML using another standard, the Hypermedia/Time-based Structuring Language or HyTime (ISO/IEC 10744:1992). Several appendices include the SGML markup for one chapter in the book and the DTD used to produce it. A diskette accompanying the book provides the DOS user with several sample SGML documents, a viewer for inspecting formatted versions of WAE documents, an SGML parser, the DocBook and HTML DTDs, and a list of Web sites with SGML information.

Much of the material in the book could be used to stimulate the kind of invigorating debate that sometimes occurs around a cafeteria lunch table. For example, the authors claim that the advent of desktop publishing was not a true paradigm shift, because users of desktop publishing systems think of their task as the production of pages, just as did earlier authors. They believe that use of SGML is a more fundamental change, because its users think differently about the deliverable they are producing — it is an information product that may or may not be realized on physical pages. While classifying the profundity of evolving technology is clearly subjective, the authors overlook an important characteristic of computerized text processing. Before desktop publishing, an author expected his work to be typeset by a printer — someone else had to be involved in making the work available to its audience. One of the important differences enabled by current tools is the ability of the author to produce camera-ready copy. Furthermore, the authors point out that SGML divorces a document from processes performed upon it. They neglect to point out that editing a document is a process. While observing that formatting a document for printing or screen display can be completely independent of the string of characters a writer sees while editing, they fail to admit that a tagged character file is formatted, too. They miss the opportunity to discuss what-you-see-is-what-you-get editing of SGML documents. Does reading markup affect an author's perception of his own words? They even recommend against indenting SGML markup to indicate the hierarchy of elements in a document because of possible misinterpretation of the resulting white space. Instead, they should have suggested that DTD developers consider the markup style of their users and that SGML formatting applications discard extra space and tab characters.

The book stresses the benefits of standardized DTDs. All the DTDs that appear in the book are very general. This emphasis eliminates one of the advantages of SGML. A DTD can enforce many of an organization's writing standards, requiring, for instance, that at least one introductory paragraph precede the first figure in a section, or that there be at least two sections at each level of subdivision. The WAE DTD is so flexible that it does not even require that a manual have a title. While this approach is a perfectly legitimate application of SGML, the book would have been improved by illustrating a more restrictive DTD philosophy as well. Naming style is another area of personal preference long debated in programming circles. The WAE DTD uses abbreviated names that the reader must repeatedly translate: `ti` for title, `au` for author, and so forth. Ironically, the DTD the authors used to produce the book, provided in Appendix C, includes more complete names. There is no mention of recursive element types. A subdivision of a section is a subsection (or an `ssec`) and not also a section. There are valid reasons for both approaches, and a richer presentation would have at least illustrated both strategies.

Minor improvements could be made in a handful of other areas. The authors refer several times to markup minimization, a general term that refers to several optional SGML conventions for abbreviating markup. One form of markup minimization is tag omission in which the markup that indicates the start or end of an element is omitted. The authors refer several times to this form of minimization without explaining any of the others. Furthermore, discussion of the "omitted tag minimization parameter" suggests that this construct allows tags to be omitted. It should have been made very clear that tags can only be omitted when the DTD provides for their omission and the document instance is such that an SGML parser, following a rigorously defined algorithm, can recognize the start or end of the element without the tags.

Better comments could be used throughout the sample DTDs. The WAE DTD contains only a few comments that label major syntactic components. There are no comments explaining the abbreviations used in the short names or the intended semantics of the declared elements. There are no comments at all in the DTD used to markup the sample chapter. Finally, a windows-oriented interface to the material on the diskette would make these samples more accessible to the intended nontechnical end users.

Despite these criticisms, or perhaps because the subject matter encourages the SGML-knowledgeable reader to be aware of stylistic preferences, the book is of interest to both SGML novices and to experienced practitioners. Should *README.1ST* be read first? That depends on the audience. It is intended for nontechnical end users: people who will be writing (and perhaps reading) document instances, who may read DTDs, but who will not be writing them. As the title suggests, the level of detail is directed at writers and editors. The book is not a tutorial for DTD developers, implementors of SGML applications, programmers who want to write SGML parsers, TeX macro writers, or even writers and editors who have already grasped the philosophy underlying SGML. Readers in the latter group will prefer to learn SGML from a text that presents SGML detail more directly. Once they are familiar with SGML syntax, however, they may well enjoy the perspective of *README.1ST*.

◇ Lynne A. Price
Text Structure Consulting
`lprice@ix.netcom.com`