

# SGML

## SGML — Questions and Answers

Reinhard Wonneberger and Frank Mittelbach

### Abstract

This paper explains SGML fundamentals in a concise Questions & Answers way.

More detailed information can be found in the works that are listed in the bibliography.

Abbreviations and acronyms are explained in a glossary.

Usage of proprietary names in this paper must not be construed to mean that they are free of rights.

### 1 Functionality

**?** *What does SGML mean?*

SGML stands for *Standard Generalized Markup Language*. It is the most important standard for document processing [App89].

**?** *What is General Markup?*

General Markup is a special sort of text that will tell a reader or a program about the the logical function of ordinary text. In the L<sup>A</sup>T<sub>E</sub>X sequence `\section{Headline}`, the markup `\section` states that the text `Headline` is to be considered as a section heading. General Markup is described in more detail in [CRD87].

**?** *How can SGML be converted into print?*

The proper way is to use a parsing and translating program, which will check that the document corresponds to the Document Type Definition (DTD) and then will translate the document into one of the available document processing languages, e.g. DCF-GML, T<sub>E</sub>X, TROFF, or some other system, or into internal code with systems like Interleaf.

There are also some programs that can interpret a subset of SGML-conformant markup, due to the fact that they allow implementation of *General Markup*, see below. This is especially true for DCF and T<sub>E</sub>X [Wonn]. Interpreting SGML, however, does not allow checking for conformance with a DTD.

**?** *What is the difference between GML and SGML?*

SGML Markup will normally be parsed by an independent program to assure conformance with the

Document Type Definition (DTD) and then converted to some processing language, like DCF-GML or L<sup>A</sup>T<sub>E</sub>X.

GML Markup is one of the processing languages, and is tied to DCF, a proprietary program of IBM.

**?** *What other use can be made of SGML?*

One main application area is databases. SGML markup can be mapped to database fields and *vice versa*, and specific information can be extracted from SGML-tagged sources automatically. Another application is hypertext [Det91].

### 2 Usage

**?** *Who are the key users of SGML?*

Many producers of large quantities of documents, among them the US Department of Defense (DoD) and many other government authorities in several countries, the European Community, several research institutions, etc.

**?** *Is SGML proprietary?*

SGML is a Standard, that has been adopted by Standards committees on different levels, and as such it is non-proprietary. The software to process SGML may or may not be proprietary, depending on the software supplier.

**?** *Is there a business impact of SGML?*

Definitely yes, as SGML has been made an integral part of CALS, which is a bundle of requirements for submitting bids to the already mentioned DoD in electronic form. Due to the influence of the DoD, many vendors are obliged to meet these requirements, and other institutions will follow suit in imposing this standard.

**?** *Can SGML documents be exchanged across computer platforms?*

Yes, if you have true SGML software. Some programs up to now, however, support only a special predefined set of DTDs.

**?** *Is SGML human readable?*

That is one of its main advantages over the so-called WYSIWYG systems. You might even read a complete SGML source on the phone!

**?** *Does SGML run on my system?*

As SGML-tagged documents do not contain hidden characters, you can generate and edit them on any computer system. SGML parsers for many computer

platforms are already available or under development. See also section 4 for some information on available software.

**[?]** *How do I enter an SGML document?*

By using your favourite text editor, e.g. ISPF for MVS systems. There are also editors that help you with entering correct SGML syntax (that's the luxury class).

**[?]** *Where can I get more information on SGML?*

Just refer to the following books [Gol90, Her90, Bry88] and articles [Bar89] or [Laa91, Pop91].

### 3 Applications

**[?]** *How can we benefit from SGML?*

Our documents if tagged with SGML may have a good chance to survive us, to say nothing of the many benefits of the General Markup approach in general [CRD87] as they are well-known from systems like DCF or L<sup>A</sup>T<sub>E</sub>X. In addition, SGML gives independence of specific programming environments, providing full portability of sources like T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X. Unlike T<sub>E</sub>X and L<sup>A</sup>T<sub>E</sub>X, however, SGML is a true international standard, not only a *de facto* standard set by usage, and a standard that is backed by powerful institutions. The fundamental advantages of General Markup have been described so often that we don't dare to repeat that here, cf. [Won90].

**[?]** *Are applications confined to document processing?*

No. Another important area of application is databases. Because tags can specify the contents of a field, they can also be used to load information into a database. The other way round, information from a database can be output with tags so that it can be formatted automatically into a specific document.

**[?]** *How could SGML interact with a database?*

In some databases, there will be 'free-form texts'. In order to get specific items from such fields for reporting, one might use SGML tagging inside free-form text fields, then write out the contents of these fields to a file and process the file with an SGML parser and typesetter. In addition, fields defined in the database might be written out with generated tags. Like this, the benefits of a database can be combined with the benefits of Structured Document Processing.

### 4 Products

**[?]** *How can SGML be used on a mainframe host?*

If an MVS host is implied, there are several possibilities we know about:

- Use an SGML parser from IBM.
- Emulate SGML with DCF, provided your input obeys a few restrictions.
- Use the DAPHNE software from DFN for translation into T<sub>E</sub>X, TROFF, or DCF [SC88].
- Interpret SGML with T<sub>E</sub>X, which might be an attractive solution; cf. [Won92] for details.

**[?]** *What about workstations?*

A well-known product is Interleaf, which is also available from IBM under the name TPS. Another important product is the Publisher from ArborText.

### Glossary

AAP American Association of Publishers.

*Attribute Modifier* of a Markup Tag, used to specify different values.

CALS Computer-Aided Acquisition and Logistics Support.

DCF Document Composition Facility (IBM product), ⇒GML.

DTD Document Type Definition.

*General Markup* Meta-information specifying the logical function of ordinary text.

GML General Markup Language, part of ⇒DCF.

ISO International Organisation for Standardisation.  
ISPF Interactive System Productivity Facility, a user interface for ⇒MVS.

L<sup>A</sup>T<sub>E</sub>X Lamport's T<sub>E</sub>X, markup language and ⇒T<sub>E</sub>X macro package by Leslie Lamport.

MVS Multiple Virtual Storage, production-oriented IBM operating system for large mainframes.

*Parser* Program performing syntax analysis on some source.

SGML Standard Generalized Markup Language.

*Tag* Explicit label that marks start (and end) of a source entity.

T<sub>E</sub>X (from the Greek *τέχνη*) typesetting program by Donald E. Knuth.

WYSIWYG 'What you see is what you get'; slogan describing screen-oriented text processing tools, often considered as the opposite approach to ⇒General Markup.

### Bibliography

[App89] W[olfgang] Appelt. Normen im Bereich der Dokumentverarbeitung. *Informatik-Spektrum*, 12:321-330, 1989.

[Bar89] David Barron. Why use SGML? *Electronic Publishing (EPodd)*, 2(1):3-24, April 1989.

- [Bry88] Martin Bryan. *SGML: An Author's Guide to the Standard Generalized Markup Language*. Addison-Wesley, Woking, England; Reading, Massachusetts, second edition, 1988.
- [CRD87] James H. Coombs, Allen H. Renear, and Steve J. DeRose. Markup systems and the future of scholarly text processing. *Communications of the ACM*, 30(11):933–947, November 1987.
- [Det91] Christine Detig. T<sub>E</sub>X & Hypertext — The future of electronic publishing. In Guenther [Gue91], pages 8–12. TUGboat 12 (March 1991) Number 1.
- [Gol90] Charles Goldfarb. *The SGML Handbook*. Clarendon Press, Oxford, 1990.
- [Gue91] Mary Guenther, editor. *T<sub>E</sub>X90 Conference Proceedings; University College; Cork, Ireland, September 10–13, 1990*, Providence, Rhode Island, U.S.A., March 1991. T<sub>E</sub>X Users Group. TUGboat 12 (March 1991) Number 1.
- [Her90] Eric van Herwijnen. *Practical SGML*. Kluwer, Dordrecht, NL, 1990.
- [Laa91] C.G. [Kees] van der Laan. SGML(, T<sub>E</sub>X and ...). In Guenther [Gue91], pages 90–104. TUGboat 12 (March 1991) Number 1.
- [Pop91] N.A.F.M. Poppelier. SGML and T<sub>E</sub>X in scientific publishing. In Guenther [Gue91], pages 105–109. TUGboat 12 (March 1991) Number 1.
- [SC88] A. Scheller and C. Smith. *DAPHNE; Document Application Processing in a Heterogeneous Network Environment; Dezentrale Verarbeitung von Dokumenten auf der Basis von SGML; Benutzeranleitung (Version 3.0)*. GMD-FOKUS, Berlin, im Auftrag des Vereins zur Förderung eines deutschen Forschungsnetzes e.V., April 1988.
- [Won90] Reinhard Wonneberger. Structured document processing: the L<sup>A</sup>T<sub>E</sub>X approach. In J. Nadrchal, editor, *Man-Machine Interface in the Scientific Environment. Proceedings of the 8th European Summer School on Computing Techniques in Physics. Skalský Dvůr, Czechoslovakia, 19–28 September 1989*, volume 61 of *Computer Physics Communications*, pages 177–189. North Holland Publishing Company; Elsevier Science Publishers B.V., 1990.
- [Won92] Reinhard Wonneberger. Approaching SGML from T<sub>E</sub>X. *TUGboat* 13(3):223 (July 1992).
- [Wonng] Reinhard Wonneberger. T<sub>E</sub>X in an industrial environment. In Anne Brüggemann-Klein, editor, *Proceedings of the 4th European T<sub>E</sub>X Conference, September 11th–13th, 1989*, Karlsruhe, forthcoming.

◊ Reinhard Wonneberger and  
Frank Mittelbach  
EDS Electronic Data Systems  
(Deutschland) GmbH  
Eisenstraße 56 (N15)  
D-6090 Rüsselsheim  
Federal Republic of Germany

## Dreamboat

Editor's note: This column heading hasn't appeared for years, but it seemed an appropriate corner in which to collect ideas and suggestions related to the topic "Where do we go from here?" In addition to the following articles, which were written before the formal recognition of interest in future directions, Philip Taylor has reported in this issue (p. 138) on the first meeting of the working group coordinating the discussion.

### T<sub>E</sub>X wish list

Michael Barr

It is the rare user of T<sub>E</sub>X who has not, at some time, felt that T<sub>E</sub>X lacks some feature or other. Since Knuth has announced that T<sub>E</sub>X is now frozen, save for an occasional bug fix, it is up to the T<sub>E</sub>X community to give thought to the kinds of features that we want in any successor to T<sub>E</sub>X.

I do not expect that my wish list will be exhaustive or that the future program will implement every one of my suggestions. I am merely trying to start a dialog on the kind of program we want in the future.

Let me say a few words about what I don't want. I don't expect to see a WYSIWYG program, although a multitasked previewer would be nice. I don't expect to see a page layout program. In fact, I don't want to think about page design at all. Ideally, future T<sub>E</sub>X will take care of all design details itself. It is a *tour de force* to lay out *TV Guide* in T<sub>E</sub>X, but T<sub>E</sub>X is not the tool I would have chosen for the job.

Here are some of the things that I have felt lacking in T<sub>E</sub>X, in no particular order. I divide them into two groups, depending on whether or not they could be made compatible with current device drivers. The reason is that there is basically only one T<sub>E</sub>X program, but as many device drivers, and more, as there are devices. Thus the amount of work that is involved in upgrading the latter is orders of magnitude larger than that which is involved in upgrading T<sub>E</sub>X itself.

### Features that could be implemented without changing device drivers

**A smart \put.** By a smart \put, I mean a procedure similar to the \point defined on page 389