

## Thinking outside conventional aerospace and defense technical publications using Standard Generalized Markup Language (SGML)

Nikhil Lobo

BAeHAL (Joint venture of Hindustan Aeronautics Limited, India and BAE Systems, UK)

*nikhillobo@baeal.com*

**Abstract:** In aerospace and defense, documentation is of a very large size, highly structured and needs constant updating. Managing this documentation has been a constant challenge to this industry. At present, documentation is being created using traditional publishing software resulting in wastage of time and effort. Standard Generalized Mark-up Language (SGML) allows a document to be broken up into modules allowing reusability. SGML enforces content to be developed in a structured manner maintaining consistency across publications. This structured approach is achieved using a Document Type Definition (DTD). Separation of content from formatting is achieved using Format Output Specification Instance (FOSI).

**Keywords:** SGML, Document Type Definition, Format Output Specification Instance, technical publications.

### Introduction

The aircraft and defense industry is faced continuously with a challenge to manage information and publishing requirements. The documents particularly in this industry are generally of a large size, highly structured and regularly updated. Since this information is used to train aircraft personnel for maintenance, training and operation, it is mandatory to ensure that the information is easily available and technically accurate. A single aircraft maintenance manual can even exceed 40,000 pages. Content is constantly being revised to represent new configurations of aircrafts, enhancements in technical procedures and changes in engineering practices. This requires documents to be updated and reissued every 120 days. Moreover this information needs to be delivered to a wide range of customers in different output forms like paper, microfilm, digital interchange formats, CD-ROM and Web.

Since the advent of desktop publishing tools and word processing software for creation of technical publications, processes have been labor intensive resulting in long publishing life cycles. Each time a document is created or updated formatting has to be applied manually. Moreover preparing documents for print or web requires complete reformatting. There is a tendency for authors to recreate content instead of reusing it resulting in an increase in both time and cost. This results in authors updating and reformatting every instance the information is repeated. This imposes a burden on authors and prevents them from frequent updating of information. Moreover inconsistencies in structure of information across similar documents make the content difficult for readers to understand and update.

### Standard Generalized Markup Language

The word "markup" was originally handwritten instructions on an author's manuscript to describe the

appearance of text on a printed page like a wavy underline to indicate boldface or special symbols for passages to be omitted or printed in a particular font (Burnard, L. 1991. <http://xml.coverpages.org/edw25.html>). With the introduction of computers these markup instructions were embedded in the text of the document itself by a process known as "specific markup". These markup instructions were very obscure characters to distinguish them from the body text. These had to be entered manually and were very time consuming. Moreover every new phototypesetting system had its own proprietary markup language thereby binding consumers to a particular language and vendor.

The Graphics Communications Association (GCA) in the 1980's set out to define a standard markup language known as 'GenCode'. However, the GCA was faced with the problem of building a tag set that would address the needs of all phototypesetting systems. As the GCA was working on this problem, an ANSI committee was creating a standard called Generalized Markup Language (GML). In this standard a document was represented as a hierarchical tree of different elements having each formatted in a certain way. These two organizations combined their efforts to build a single standard and in December 1986 introduced Standard Generalized Markup Language (SGML). Let's look at each word in SGML to see what it adds to the concept.

**Standard** - SGML is an internationally recognized standard (ISO 8879).

**Generalized** - SGML allows documents to be stored in a neutral format (Smith, 1992). This neutrality is achieved by using generic identifiers instead of application specific identifiers. The fact that SGML is independent of systems, devices and languages enforces the term generalized.

**Markup** - Markup is text added to the data of a document to convey information about it and to indicate its structure. The procedure and rules to markup a document is stated in SGML.

**Language** - SGML is actually not a language but is a metalanguage. It has syntax but no semantics (Heimbürger, 1994). It is used to describe a document. A document can be broken into three layers: structure, content and style (Wright, 1992; Gilmore, 1993). SGML separates these three layers and mainly deals with the relationship between structure and content.

SGML is an open standard which is both platform and application independent. SGML files are ASCII text files which can be used on virtually any platform. SGML relies on the principles of descriptive markup - where the markup is used to indicate the nature, function or content of the data in a file, rather than saying how that data

should be processed. SGML has introduced innovations that drastically improve the publication process and is a language for content creation that formalizes markup and frees it of any system processing dependencies. Reuse of information across multiple publications by splitting documents into smaller modules enable authors to assemble information in different combinations for various purposes and audiences. Content is structured using a Document Type Definition (DTD) and separation of content from formatting is achieved using a Format Output Specification Instance (FOSI). Some of the benefits of SGML includes: a) Documents can be exchanged with users having different software applications, computer networks and hardware platforms. An SGML document will outlive existing and future technologies. b) Since content is separated from style using SGML, an author concentrates more on content rather than appearance. The author can also reuse pieces of a document saving the effort needed for recreation. c) Data can be updated easily especially when copies of a publication are located in several places and failure to update all copies of a publication can lead to operational failures. d) Using SGML many applications can access the same set of information. Data can be broken into chunks of information and a chunk may appear in technical manuals, training guides and publications. Updating of a chunk of information ensures that many applications that use the same set of information have the latest update. e) Conversion of documents created using SGML is not needed when hardware or software becomes obsolete. SGML's standard file format always allows availability of information once a document is created. f) Since documents created in SGML are structured, it is possible to perform complex searches in a document. Database technology uses fields and records to store information while SGML uses elements such as chapters, sections, titles and paragraphs instead of fields and records. To find all publications in which the first paragraph start with the words "OEM Compliance" is a complex task in paper publications. This task is made simpler using SGML.

SGML implementation is ideal for aerospace and defense industries since it addresses the following requirements:

- Having huge volumes of data. For example at present Airbus A320 has documentation representing 1.2 gigabytes in 39 different manuals.
- Parts of information appearing in more than one document.
- Information must be accessible across succeeding generations of computer hardware and software.
- Information is considered critical and an important corporate asset.
- Frequently updated and revision to information.
- Linkages to granular levels of a document like words, paragraphs, sections, etc.
- Requiring complex approval processes and sign offs.

- Information coming from different sources and is in different formats.
- Requiring integration of components of information produced by multiple authors.
- Information to be shared among several organizations.
- Integration of components that depend upon external factors like economic or environmental factors.
- Data targeted to different user groups each having specific security classifications.
- Information can be interchanged across heterogeneous systems regardless of the software application or hardware platform for which it was originally created (Turner, 1994).
- Must be produced in many different formats, such as paper, CDROM, Web, etc.

### Modular approach

The publication process initiates with identification of modules that comprise the publication under development. A module is a self-contained unit of data containing text and/or illustration. A technical publication of equipment can be comprised of modules conveying information on its description, operation, assembly, disassembly, cleaning, repair and storage. Each module has metadata associated with it which specifies information that uniquely identifies it from other modules viz. module number, title, issue number and date.

In Fig.1 below technical information of a Main Rotor Blade is divided into independent modules like description, operation, assembly, disassembly and cleaning. Each of these modules is an independent file containing text and/or illustration. These files are created using Standard Generalized Markup Language, Document Type Definition, and Formatting Output Specification Instance.

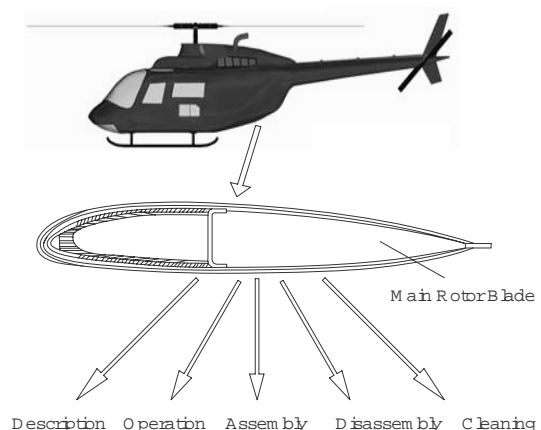


Fig. 1. Main Rotor Blade Modules

### Structured content

In each module, SGML enforces content to be developed in a structured manner. This maintains consistency across publications. The idea of SGML is to structure text by adding explicit structure descriptors

(McCormick, 1999). This structured approach is achieved using a DTD. Since SGML requires explicit and consistent structure, the DTD defined for the documents will impose rules that may not have been considered when the author created the original document (Gross, 1993).

In a SGML module a DTD defines elements allowed, their associated attributes and relationships among elements. An instance is any text created according to the rules dictated by the DTD. These instances are validated with respect to the DTD using compilation software called a SGML parser. For example, an installation module can be defined having the following elements:

- Module Number is a number that uniquely identifies the equipment and its module type i.e. description, operation, installation, removal or cleaning.
- Title specifies the equipment name and module name.
- Issue Number shows the issue number.
- Issue date is the date that the module is released and comprises the "day", "month" and "year".
- Preliminary Steps either indicate that no required conditions are to be carried out before starting the procedure or it gives a brief description of the requirements.
- Required Persons identifies persons required to perform the task. It comprises of trades of the persons required (Mechanical, Electrical, Radio, etc.) and skill level of the person (Basic, Intermediate or Advanced).
- Support Equipment shall contain a list of special tools and general test equipment used to carry out the procedure. The list shall contain entries for Part No., Nomenclature and Quantity for each of the tools listed.
- Supplies shall contain a list of consumables used to carry out the procedure. The list shall contain entries for Part No., Nomenclature and Quantity for each of the consumables.
- Spares shall contain a list of replacement parts required to carry out the procedure. The list shall contain entries for Part No., Nomenclature and Quantity for each of the supplies.
- Safety identifies any safety conditions applicable to the task being undertaken. It includes any Warnings, Cautions and Notes that are applicable to the entire procedure. A warning is used to alert the reader to possible hazards, which may cause loss of life, physical injury or ill health. A caution is used to denote a possibility of damage to material but not danger to personnel. A note is used to convey information that is extraneous to the immediate subject of the text.
- Procedural Steps tells the reader how to perform the task.
- Final Checks is used to identify any additional requirements to be carried out after completion of the procedure.

Once the structure of a module is finalized, installation data of all components will follow this structure. Since the data is structured it makes it easier to read and understand. Moreover structured information allows reuse and automatic formatting using a Format Output Specification Instance (FOSI).

### Defining a document

A DTD defines the structure of a SGML module. It is recommended when designing a DTD, the user first analyze the type of documents for which the DTD is required. A DTD is constructed so that it includes every element that is common to any given type of document (Stern, 1995). The DTD functions mainly as a template identifying the type of module - description, operation, assembly, disassembly or cleaning and the order of elements appearing in the module. A DTD that is created can be general enough so that it can apply to all modules of the same type. In other words if a DTD is structured properly it can be applied to most modules of the same type. There will be no need to recreate a DTD each time a user writes an installation module. The original installation DTD can be reused over and over again. It is composed of elements, attributes and entities. A DTD defines the following:

- Elements that are allowable and how often an element may appear.
- Order of elements appearing.
- Attributes and their content type.
- Names of all entities that can be used.

SGML parsers are programs that check if the markup in a document satisfies the rules defined by the DTD. Consider a portion of the installation module structure shown in Fig.2.

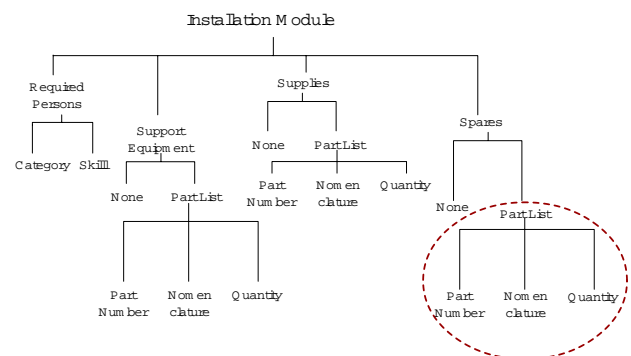


Fig 2. Portion of Installation Module Structure

The DTD and SGML fragment for this portion is as follows:

SGML DTD

```
<!ELEMENT partlist - (part)+>
```

```
<!ELEMENT part - (partno, nomen, qty)>
```

```
<!ELEMENT (partno, nomen, qty) - (#PCDATA)>
```

SGML Fragment

```
<partlist>
```

```
<part>
```

```
<partno>1234567</partno>
```

```
<nomen>Aaaa</nomen>
<qty>444</qty>
</part>
<part><partno>2345678</partno>
<nomen>Bbbb</nomen>
<qty>5,555</qty>
</part>
<partno>3456789</partno>
<nomen>Cccc</nomen>
<qty>666,666</qty>
</part>
</partlist>
```

### Reusability

Modularization allows technical publications to be broken up into modules allowing reusability of content instead of recreation. When an author does not find content that already exists, then it is recreated. On the otherhand if the content is found, it is typically copied and pasted into a new document. Both of these approaches result in wastage of effort and time in finding and updating all repeated passages of content. Rewriting instead of reusing also poses a risk of redundant information being inconsistent across documents.

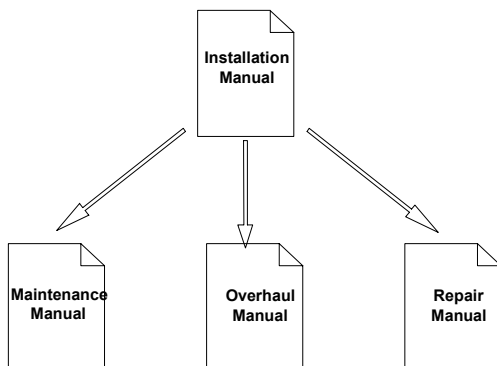


Fig. 3. Installation Module Reusability

In Fig.3 an installation module created for a component can be used across maintenance, overhaul and repair manuals without recreation and reformatting.

### Illustrations

Illustrations also play an important role in the development of data modules. Each illustration in a data module is identified with a unique id, title and board number. A board number is used to link an illustration to a SGML illustration element. To add an illustration to a SGML file we must first define a board number for each illustration and link it to the corresponding externally stored illustration file as shown below.

Here the board number (HEL-ROTORSYS-MRB-001) is linked to the illustration file "main-rotor-blade.CG4".

```
<!DOCTYPE installmod SYSTEM "install.dtd">
[
<!ENTITY HEL-ROTORSYS-MRB-001 SYSTEM "main-
```

```
rotor-blade.CG4" NDATA FAX>
]>
```

Once the board number is linked to an illustration file, we can specify in the SGML file where we would like the illustration to appear.

```
<fig id="F1">
<title>Main Rotor Blade</title>
<grap board_no=" HEL-ROTORSYS-MRB-001">
</fig>
```

Commonly used illustration formats are Computer Graphics Metafile (CGM) and Tagged Image File Format (TIFF).

### Independent formatting

Formatting Output Specification Instance (FOSI) is a style sheet language developed to control the formatting of SGML technical data. These FOSI stylesheets are written in SGML to describe the presentation of structured documents. Since the content has been structured using a DTD, the document can be presented in different styles. For each DTD there should be at least one FOSI. However multiple FOSIs for the same DTD can be developed to produce different publishing formats for the same SGML instance.

Examples:

Each line starts with <eic gi="..."> that contains formatting characteristics of an element defined in the DTD. An "eic" denotes "element in context" and "gi" denotes "generic identifier".

To make the title bold

```
<eic gi="title">
<charlist>
<font weight="bold">
```

To start a procedure on a new page with text

```
PROCEDURE
<eic gi="proc">
<charlist>
<textbrk startpg="1">
<puttext="PROCEDURE">
```

To indent paragraphs appearing in a caution by 10 points

```
<eic gi="para" context="caution">
<charlist>
<indent leftindent="10pt">
```

To make paragraphs appearing in a warning as bold

```
<eic gi="para" context="warning">
<charlist>
<font weight="bold">
```

### Integration of SGML and World Wide Web (WWW)

Large audiences can be reached out to information that is time critical and changes rapidly using the WWW. This helps to reduce high costs associated with reproduction and distribution of information in hard copy format. SGML is playing an important role in information available on the Internet. The WWW is a hypertext system based on SGML. Hypertext Markup Language (HTML) based on SGML has its own DTD and also relies on tags. These tags are read in the HTML file and



displayed on the computer screen of the user using a WWW browser.

All WWW browsers may not be capable of reading SGML documents. Since the documents on the WWW are in HTML format, SGML documents can be converted to HTML using a parser that meets the requirements of a specific browser. This conversion would ensure that the document conforms to the HTML DTD for use on the WWW.

In some cases SGML documents may be large in size with respect to the number of pages of information held in an SGML file. WWW browsers work best with documents having few pages. These SGML documents can be broken down further and linked to enable users to access components they desire without having to access a large document.

WWW will facilitate universal access to information that is unrestricted to personnel on a daily basis. Distributing information via the WWW requires information to be published in only one place and once published this information is available to any of the million web users. A central repository for electronic document distribution simplifies the updating procedure and version control. Old versions are replaced with new releases and any one accessing these documents gets the latest information. Users can also query documents and search for information or related topics.

#### SGML and users

##### Her Majesty's Stationery Office

Her Majesty's Stationery Office (HMSO) is the UK Government publisher and is responsible for printing statutes, as they are passed by Parliament. From 1987, statutes have been prepared using SGML with the intention that every new statute will appear both in printed and electronic form. All versions of legislation will be stored with cross-referencing achieved using SGML.

##### The Commission of the European Communities

A system called FORMEX (Formalized Exchange of Electronic Documents) has been developed by the Official Publications Office of the European Communities to facilitate in the production of the publications of the Commission along with archiving of these documents in electronic form. SGML along with suitable DTDs was chosen for developing various documents.

##### Oxford University Press

SGML has been used in the production of electronic version of the Oxford English Dictionary. The 12 volumes and supplement (4 volumes) comprising of 21 thousand pages have been coded into SGML. Using SGML all future editions will be maintained along with different variants of the dictionary.

##### McGraw-Hill

The Encyclopedia of Science and Technology of McGraw-Hill was prepared using SGML. This made it possible to be produced both in printed and electronic forms (CD-ROM). This textual material is also now available in an on-line public database.

#### Hewlett-Packard

More than fifty writing departments worldwide are involved in the production of documentation for Hewlett-Packard computers, software and instruments and exchange of information in electronic form between departments is a requirement. SGML has been used to address this requirement and a parser called MARKUP developed in-house is being used.

##### Ontario Hydro (Canada's largest utility)

SGML was chosen as a technology to convert 20,000 pages of documentation comprising of 11,000 pages of operator's manuals, 8,500 pages of training manuals, and 300 pages of schematics. This has been used by maintenance technicians in troubleshooting and printing portions of manuals associated with the problem.

#### Comparison between traditional publishing and SGML

Table 1. Comparison between traditional publishing and SGML

Publishing System Essentials	Traditional Publishing System	Standard Generalized Markup Language
Modularization	Entire document treated as a single module.	SGML allows a document to be broken up into modules allowing reusability.
Structured Content	Content is not structured across similar types of publications resulting in no consistency.	SGML enforces content to be developed in a structured manner maintaining consistency across publications. This structured approach is achieved using a Document Type Definition.
Independent Formatting	Each time a document is created or updated formatting has to be applied manually.	Separation of content from formatting is achieved using Format Output Specification Instances (FOSIs). Content can be constantly updated without the need to worry about changes in formatting.
Addressing different media like print, web, etc.	Preparing documents for print or web requires complete reformatting.	Since the content has been structured using a DTD, the document can be presented in different styles using different FOSIs.

Many experts in the field of SGML (Gilmore, 1993; Lunemann, 1995; Stern, 1995; Wood, 1995) note that there are several benefits of using SGML. Based on a study from journals, publications and the WWW, most common benefits of SGML were

- Content being separated from style
- Structured data
- Hardware and software independence
- Reusability of data
- SGML's well-defined structure makes it possible to perform complex searches in documents (Lunemann, 1995).
- Sharing of components of information
- Quick updating of information
- Publishing as different output formats

### Use of SGML in government and commercial enterprises

Government and Commercial enterprises are already in the process of migrating from paper to electronic publishing. Some of the commercial enterprises are listed below:

- Her Majesty's Stationery Office: Statutes passed by Parliament.
- Commission of the European Communities: Publications of the Commission.
- Hewlett-Packard: Documentation for computers, software and instruments.
- Oxford University Press: English Dictionary.
- McGraw-Hill: Encyclopedia of Science and Technology.
- Ontario Hydro: Operator's manuals, training manuals, and schematics
- Shell UK Exploration & Production: Safety Manuals.
- Silicon Graphics - IRIS InSight: Online documentation.

### SGML editors

Creation of SGML documents conforming to a DTD requires both an SGML editor and a parser. The editor is used to enter information and insert markup into the document and the parser checks the markup conforms to the rules specified in the DTD.

Following are the popular SGML Editors:

- SoftQuad's Author/Editor
- Arbortext's ADEPT editor
- GRIF's SGML editor
- InContext
- TimeLux's EditTime

James Clark's SP Parser is a popular SGML parser.

SGML editors are of two major types, native and structured. The difference between them lies in how they manipulate and store the data. Native SGML editors import, manipulate and store the data in SGML format (ASCII text). In contrast, structured SGML editors use their own internal formats.

### Conclusion

On the basis of analysis of the information gathered, there is an indication that SGML is a powerful tool that aerospace and defense publication departments need to use to improve the efficiency and effectiveness of its publication process. SGML reduces the time taken to publish technical publications, standardize document structure and eliminates the storage of paper-based documents besides providing the latest accurate information which permit its availability electronically on personal computers.

The documentation life cycle (creating, distributing, retrieving and reusing) has been constantly affected by new generations of hardware and software. SGML also solves a problem most people do not know they have. Not until an organization reaches a crisis – the inability to deal with backlog, the high cost of converting to a new

system, or the failure to keep information accurate and up-to-date— does the organization realize there is a problem (Ensign, 1993). SGML has overcome these hurdles with benefits in terms of portability of information, reusability, availability and an increase in productivity.

SGML also plays an important role in availability of information through the WWW which enables customers to access latest information around the clock. This reduces hurdles faced with hardcopy updating and distribution. Retrieval of information is done in a matter of seconds as compared to the days when a hardcopy had to be requested from the publication distribution office. Search and retrieval of information is achieved through links to various documents that are interrelated. Aerospace and defense publications can be linked to each other and to other documents such as forms that are referenced by them. Linkages of documents enable accessing of publications with a supplement very simple. Since documents are linked, the supplemented information can appear in a requested publication without the need of accessing two separate documents.

Once a decision is made to adopt SGML, commercially available editors and parsers are available for conversion and authoring of documents to SGML. An abundance of information on SGML products and tools are available on WWW.

### References

1. Cleveland L (1995) SGML: A Solution to Content Portability, The Magazine for Magazine Management, March 1 1995.
2. Ensign C (1993) SGML By Evolution. *Technical Communication*. pp: 387-393.
3. Heimburger A (1994) Introduction to Standard Generalized Markup Language (SGML). *Microcomputers for Information Management*. pp: 239-260.
4. Gilmore E (1993) Introducing Today's SGML. *Technical Communication*. pp: 210-218.
5. Gross, M. (1993) Getting Your Data Into SGML. *Technical Communication*. pp: 219-225.
6. Lunemann, S. (1995) Nuts and Bolts of SGML. *Technical Communication*. pp: 344-345.
7. McCormick B (1999) SGML document introducing you to SGML, [online], Cloud 9, <http://www.cloud9.net/~bradmcc/WhatIsSGML.html>.
8. Smith M (1992) SGML and Related Standards: Document Description and Processing Languages, New York: Ellis Horwood Limited.
9. Stern D (1995) SGML Documents: A Better System for Communicating Knowledge. *Special Libraries*. pp: 117-124.
10. Turner L (1994) What Every CIO Should Know About SGML, CALS/Enterprise.
11. Wood, M. (1995) So What is SGML? *J. Systems Management*. pp: 24-29.
12. Wright H (1992) SGML Frees Information. *Byte*. pp: 279-286.