Home | Search | What's New | Products | Survey | Help

Printer Friendly Version

UNIVERSITY OF FLORIDA
IFAS EXTENSION

edis

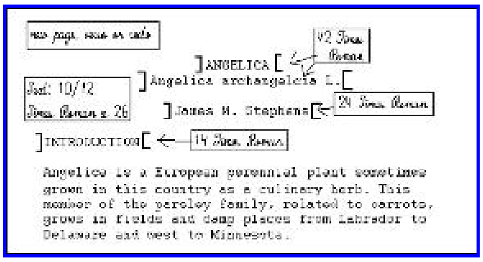
Brief History of Document Markup[1]

Dennis G. Watson[2]

INTRODUCTION

Document markup is the process of adding codes to a document to identify the structure of a document or the format in which it is to appear. Document markup is a communication form that has existed for many years. Until the computerization of the printing industry, markup was primarily done by a copy editor writing instructions on a manuscript for a typesetter to follow (Figure 1). Over a period of time, a standard set of symbols was developed and used by copy editors to communicate with typesetters.

As typesetting functions became computerized, text formatting languages were written. A typesetter would convert the copy editor's markup into the appropriate markup for the text formatting language being used. Figure 2 is an example of the commands used by a typesetting system to print the chapter heading of a document.


42 Times Roman
Text: 10/12 Times Roman x 26
24 Times Roman
14 Times Roman

ANGELICA

Angelica archangelica L.

James M. Stephens

INTRODUCTION

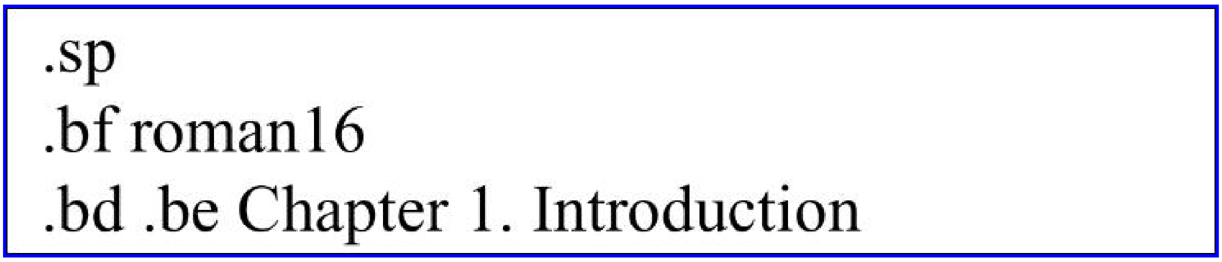
Angelica is a European perennial plant sometimes grown in this country as a culinary herb. This member of the parsley family, related to carrots, grows in fields and damp places from Labrador to Delaware and west to Minnesota.

Figure 1. Portion of a marked-up page with instructions for a typesetter.

The style of the document described in Figure 2 calls for the chapter heading to be one line below the normal starting point, the font to be changed to 16 point roman, and the heading to be centered and printed in bold. Since many text formatting languages were developed, many different instruction sets exist.

As computers became widely available, authors began using word processing software to write and edit their documents. Each word processing program had its own method of markup. Most electronic devices which store text for later recall and output use some form of markup. The markup may or may not be apparent to the user. The markup may be visible, hidden, entered by the user, or automatically generated. In some cases document markup takes the form of alphanumeric text characters and in other cases it is stored as binary data. In most cases, some form of delimiter is used to indicate the beginning of the markup and optionally the end of the markup. Although the power of a word processing program to format a document is impressive, it is also a burden when one wants to switch to the next generation computer and software. When upgrading equipment and software, old electronic documents must be converted to the new format. Sometimes this is handled automatically by the new software. At other times, it is not and an author must reenter formatting information, if not the entire text.

Two categories of document markup are specific markup and generic markup. Specific markup uses instructions which are specific to the software being used to prepare or output a document. Generalized markup describes the structure of a document. For example, specific markup might include marking a heading as centered and bold. With generic markup, a first level heading could simply be marked as "head1."



```
.sp
.bf roman16
.bd .be Chapter 1. Introduction
```

Figure 2. Sample commands for a text formatting language which describe the appearance of a chapter heading.

Figure 3. Sample commands for a text formatting language which describe the appearance of a chapter heading.

Text formatting languages are another example of specific markup (Figure 2). The commands used by the language are specific to the text processing software and usually have to be reentered in order for the text to be processed by a different text processing program.

Interactive word processors allow an author to add specific markup to a document as it is being written. This allows an author to generate a visually pleasing document, but the markup is specific to the word processing program being used.

GENERIC MARKUP

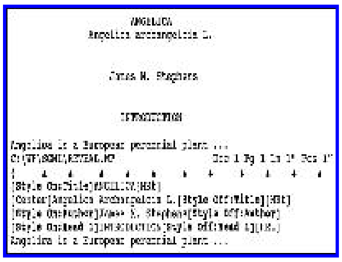
Generic or generalized markup is the term that describes the process of assigning generic names to markup. Figure 3 is an example of a generalized markup applied to the same text as Figure 2 .

Macros for typesetting languages were the beginning of generic markup. A macro is a software instruction which executes a series of instructions (a type of shorthand notation). A "chapter" macro can be defined with the contents of the instructions in Figure 2 , to print a chapter heading. This explains the simplicity of the markup in Figure 3 . The macro can also keep track of chapter numbers. If the publisher wants to change the appearance of the chapter heading, only the macro would need to be changed, not each individual heading in the document. The style information on the appearance of a document is kept separate from its structure and content. The assumption behind generic markup is that documents have a structure consisting of logical components.

With word processors that use style sheets, the same level of generic markup can be obtained. A style sheet describes the appearance of a document. Titles can be marked with a "title" style, authors' names can be marked with an "author" style and so on. Figure 4 is an example of WordPerfect document in reveal codes with style codes being used. A publishing group can have multiple style sheets with the same style names, with each style sheet being used for a different appearance of the printed page. Generic markup specifies the logical structure rather than the appearance of a document.

TOWARD A STANDARD

With the availability of computers and capabilities of transferring electronic data to typesetters, large printing customers began looking at ways to reduce costs. First, customers began sending computer files to publishers, thinking this would save them the cost of the typesetter having to rekey the manuscript. They expected big savings, but the publisher offered only a nominal discount. The publisher argued that they had to go through the document character by character and insert typesetting codes for typeface, font sizes, bold, italic, foreign characters, figures, tables, charts, etc.



```
                    ANGELICA
             Angelica archangelica L.


                James M. Stephens


                  INTRODUCTION

Angelica is a European perennial plant ...
C:\WP\SGML\REVEAL.WP                Doc 1 Pg 1 Ln 1" Pos 1"
[Style On:Title]ANGELICA[HRt]
[Center]Angelica Archangelica L.[Style Off:Title][HRt]
[Style On:Author]James M. Stephens[Style Off:Author]
[Style On:Head 1]INTRODUCTION[Style Off:Head 1][HRt]
Angelica is a European perennial plant ...
```

Figure 4. WordPerfect document in reveal codes, illustrating generic markup of title, author, and first level heading with styles (style boldface).

Some customers decided to ask for the typesetters' codes so they could enter these codes at the time the document was being edited. Typically, multiple typesetters were used, and as the typesetters responded, several facts became obvious. Different typesetters used completely different coding schemes. One required a `.bo' command preceding a word or words to be printed in bold, whereas another required a `{fbo##' command. For another typesetter, the command had to be on a line by itself. Typesetters typically reminded the customers that they would still have to charge for proofing each typesetting command. Another limitation was that the only way to save on reprinting costs was to use the same publisher as for the original printing, because of the different coding schemes in use.

Obviously, a better method was needed to encourage the sharing of data among organizations without having to redo the typesetting codes each time. Another need was for a marked document to have a life longer than the software for which it was originally coded.

STANDARDIZATION TIMELINE

The need for longevity of markup and the need for marking documents for electronic database storage led to the development of the international Standard Generalized Markup Language (SGML). Following is a summary of key events which led to SGML.

1967 - William Tunnicliffe spoke on separation of information content of documents from their format, at a meeting at the Canadian Government Printing Office.

Late 60s - Stanley Rice, a New York book designer, proposed a set of parameterized `editorial structure' tags. The Graphic Communications Association (GCA) helped sponsor workshops, seminars, and committees to further develop the concept. From these efforts grew the original GCA GenCode committee. GenCode defined a generalized markup approach based on a document's hierarchy. The markup approach was integrated with a generic coding that emphasized descriptive rather than procedural coding.

1969 - Charles Goldfarb, Edward Mosher, and Raymond Lorie invented the Generalized Markup Language (GML) for IBM. GML was based on the generic coding ideas of Rice and Tunnicliffe.

1970 - Goldfarb proposed a generalized markup language based on the premises that: 1) markup should describe a document's structure rather than its physical characteristics, and 2) markup should be rigorous so that it can be unambiguously understood by a program or a human interpreter.

1978 - An ANSI working group was formed and supported by GenCode and subsequently led by Goldfarb to provide an unambiguous format for text interchange and a markup language rich enough to permit future processing. Their work was based on GML. As the committee considered methods of handling markup, they realized that it needed to be generic. One of the concepts they decided on was to mark a title as <title> rather than <bold> and <center>. By marking a title as <title>, database searches could be limited to titles. This was the beginning of SGML, which represents the structure of a document. Another innovation of the committee was that one could specify the order in which objects could appear in a document (e.g. TO before FROM in a memo). The committee borrowed the concept of header files (or referenced files) from programming languages (such as C) and included tags which the markup procedure would use in a header file and the actual use of the markup in a separate file.

1980 - First draft of SGML, by ANSI committee.

1983 - Sixth working draft of SGML released. Adopted by the Internal Revenue Service and the US Department of Defense.

1984 - SGML working group reorganized under ISO and ANSI concurrently with Goldfarb serving as the technical leader and editor for both groups. The ISO working group was ISO/IEC JTCI/SC18/WG8.

1985 - Draft international standard published.

1986 - SGML approved as ISO international standard 8879.

1991 - Procedure initiated for 5-year review of SGML.

SUMMARY

The process of adding markup to documents has evolved from specific typesetting codes which were often different from one typesetter to another, to a standardized generic markup language. SGML (Standard Generalized Markup Language - ISO 8879) has achieved widespread acceptance and is being used by major government agencies such as the Department of Defense and the Internal Revenue Service. With generic markup as defined by SGML, the process of converting an electronic document from one computer system to another can be automated with computer software. Automated processing of text for inclusion in large information databases is a significant step for providing timely, pertinent information in the information age.

Footnotes

1. This document is Circular 1086, one of a series of the Agricultural and Biological Engineering Department, Florida

Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida. Original publication date November, 1992. Reviewed July, 2002. Visit the EDIS Web Site at http://edis.ifas.ufl.edu.

2. Dennis G. Watson, associate professor, Agricultural Engineering Department, Cooperative Extension Service, Institute of Food and Agricultural Sciences, University of Florida, Gainesville FL 32611.

Florida Cooperative Extension Service / Institute of Food and Agricultural Sciences / University of Florida / Larry R. Arrington, Interim Dean

Copyright Information