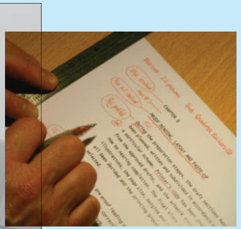
BCS

issues in informatics

Mark-up: From styling to structuring information

HTML is the language of the Web, and XML has become a favoured way of structuring information. But not many people realise that these languages owe their origins to typesetting….

Photos: Conrad Taylor



Publication designers used to spend a lot of their time applying written markup to typescripts.

Quadritek typesetting terminal *circa* 1980. The results would be seen only when output to a photographic setting machine.

1. Control codes for setting type

In the 1970s, the cast-lead system of setting type for publishing was increasingly replaced by the use of **phototypesetting machines**, in which negative images of letterforms were projected onto photographic paper, or 'bromide'. These machines used computers as input and editing terminals, and developed in parallel with early word-processing.

A graphic designer would **mark up** a typescript using a handwritten 'language' of symbols and specifications. The operator of a typesetting terminal re-typed the text into the machine, at the same time embedding formatting codes into it. This electronic mark-up carried instructions about font choice, font size, letter- and line-spacing and so on, and controlled the printing-out process.

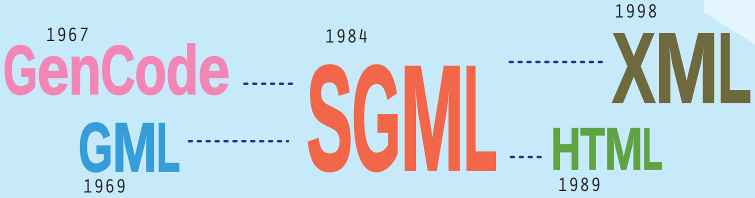
2. Typesetting mark-up comes to the PC

In the late 1970s, digital photosetting machines and laser printers arrived, creating page images out of tiny dots. This inspired computer scientists to produce typesetting systems for general-purpose computers.

Donald Knuth designed the sophisticated **TEX** typesetting system to produce quality text books, especially mathematical ones.

Brian Kernighan developed the **troff** system (typesetter run-off) for Unix.

The late 1980s brought desktop publishing software like PageMaker, Ventura and QuarkXPress, and WYSIWYG word-processors like MacWrite and Word. Although these liberated the user from having to type mark-up codes into the text, the mark-up was still there – but in the background.



1967 GenCode

1969 GML

1984 SGML

1989 HTML

1998 XML

3. Markup Revolution – the rise and rise of 'generic coding' of text

GenCode

In 1967, printer **William Tunnicliffe** suggested at a conference of the Canadian Government Printing Office that publishers should encode texts with **generic markup** – identifying parts of texts by their function, such as 'heading' or 'quotation', rather than coding for typeset appearance.

Tunnicliffe later led a project of the Graphic Communication Association, which devised a generic markup standard called **GenCode**.

Benefits of generic mark-up

In the 60s and 70s, computerised typesetting systems used a plethora of incompatible formatting languages. By translating from GenCode into a specific vendor's formatting codes only near the end of the publishing process, publishers were liberated from being locked in to suppliers' formats.

Generic coding is an approach with obvious benefits for reference books, directories and Yellow Pages, technical documentation and other texts which are frequently revised. It also makes it easier to publish in multiple formats – including on-screen media.

GML

Charles Goldfarb was charged by IBM with developing a text management system for law practices. He saw that the GenCode idea would both allow a flexible approach to formatting, and help with document retrieval.

With colleagues Mosher and Laurie, Goldfarb created **GML: Generalized Markup Language**. IBM developed this for their BookMaster documentation suite.

IBM publishing systems were developed for the IBM System/370

SGML

Goldfarb led the drive for development of the **Standard Generalized Markup Language** or **SGML**. He co-ordinated the efforts of ANSI and ISO committees, and as a result SGML became the ISO standard ISO 8879 in 1984.

Acceptance of SGML got a boost when the US Department of Defense made it a mandatory requirement for weapons systems documentation to be SGML-encoded. They reasoned this would future-proof it, and enable information to be accessed more conveniently using on-line reference systems.

What is the magic of generic markup?

- Tags are codes which are inserted in text <example>like this</example>.
- They explicitly identify parts of the text in accordance with their function, not their appearance.
- It's a very flexible way to give structure to information in text form, for manipulation by computer software.

HTML and the Web

Tim Berners-Lee worked at the European Particle Physics Laboratory (CERN) and was influenced by Anders Berglund, responsible for CERN publishing, who had played a key role in the development of SGML and software to implement it. Tim was inspired to use SGML as a starting-point to develop HTML, **HyperText Markup Language**, and other structures which led to the World Wide Web.

XML: simplicity plus 'eXtensibility'

XML (the **Extensible Markup Language**) was an initiative of people who were enthusiastic about the way in which SGML markup vocabularies can be extended to support various specific communities' needs, but dismayed about the low take-up of SGML. (This was partly due to the overly-flexible rules of SGML, which made it hard to support the standard practically with software.)

In essence, XML is the same as SGML, but with more rigorous rules about how tagged text should be presented and interpreted. XML has been hugely adopted as a way for structuring texts, and is the encoding method for the data in various knowledge-organisation standards such as RDF, XMP and Dublin Core.