Information Theory and Introduction to XML

This document aims to help you better understand why you have so many XML - 'Extended Markup Language' files - on your computer - and why you will see more.

Experts in Information Theory suggest that to maintain the information value or meaning of a 'Document' or 'Message' (whatever the form: Electronic file, video, data stream, etc) - 3 aspects must be considered. We need to understand their terminology:

Content: This is the text, bytes or signals within a message, for example the content of a conversation with a friend would be the words we speak and hear. If we use different words we often change the meaning of our message. But sometimes we will use different words, or more words, to help our friend to understand - there are often different ways of saying the same thing. In the IT world we often change the original information into different data formats for processing and transmission. Encoding, Compression, Encryption are just a few of the techniques used on the message author's input. Error detection and correction are used to ensure output is faithful to input. Meta-content (information about the information) may be used to help this.

Structure: A Word Processed document may be divided into Headings, sections, paragraphs, tables, appendices and so on. It might be easy to understand that say a number appearing in the 'Price' column of a Spreadsheet might give the Spreadsheet a different meaning if we carelessly shifted those numbers into the 'Quantity' column, but the same importance should be attached to all of the structure elements the author intended.

Presentation: Often we give meaning to the data we see say in Bold or Red, or Block Capitals. We pay attention to Bold Type, we worry when we see Red numbers and we get offended if somebody 'shouts' in Blocks. Therefore Colour, Fount typeface (I'll use US 'Font' in future) or Font-size add value to the document. If applications do not share the same standards and configuration, Presentation, and therefore meaning may be changed. There are even laws that state that important information should not be hidden 'in the fine print', that is with a small font.

Problems with portability of data, even between supposedly integrated applications from the same manufacturer, are widely recognised. Current technology often does not separate Structure from Presentation. Try 'Cutting and Pasting' some data from a WP or Spreadsheet document and placing it in another part of the document. Did the presentation (colour, font etc) do what we wanted it to do? Did it merge with its new home or keep its old format? Was the meaning consistent with what we wanted? There is always an element of subjective understanding in any message, but this should be under the control of the author, respected by handlers.

The XML international standards of the World Wide Web organisation aim to address the problems mentioned above. They are mature, widely accepted, and freely available. HTML - used for Web page definition, is a standard which came from the same stable as XML, but mainly covers document presentation, and is often poorly implemented. X/HTML is a transition standard for Web pages compatible with most modern browsers.

XML standards include the following:

XML for Structure.

This allows the definition of data structure in a simple, flexible, manner, allowing an infinite number of dimensions in a single document. The power to include or import networked documents increases the opportunities dramatically. The use of 'namespace' - the ability to

identify any part of a document with any name-family might seem complicated but gives tremendous control over processing power. It takes some getting used to the fact that a document section is quite happy communicating with another document section the other side of the world but may or may not be able to 'see' the line above if it is in a different namespace.

CSS and XSLT for Presentation control.

Managing the look of static XML-type documents (including HTML) with Cascading Style Sheets is well known. XML Stylesheet Language Transformation (XSLT) allows dynamic transformation of XML to Text,Web, PDF, graphics, Office and so on. There has been some debate about whether XSLT should be classed as a programming language, but new standards in production have defined it as such.

Document Type Definition and XML Schema for Content definition. DTD standards to define some document content have been around for almost 20 years, but are limited in functionality, and significantly are not defined in XML format. XML Schema - accepted as a standard in 1999 changes this. The power to define data in about 40 defined XML-types, plus the ability to infinitely define new types, along with XML import and namespace features, takes information control to new areas. Microsoft publishes XML Schema for import and export to and from versions of its major applications, and provides a GUI XML Mapping feature for Import and Export definition. Major database application providers support XML Schema, and the range of XML Schema processors, including Trax, JAXP, Saxon, Xerces; (all available freely) is growing.

A criticism sometimes made against the adoption of XML is that it is too verbose, compared with plain text data files. This may be true, but is an acceptable penalty given the power and confidence XML processing can provide compared with text. Where this may be a problem, even in these days of cheap storage and high bandwidth, judicious use of tag names, or even encoding, such as WBXML proposed for mobile devices, is appropriate. Abstract Syntax Notation, like ASN.1 used in SNMP Network Management, may also be useful. Being text-based means XML is highly compressible. Finally XML data compared with an Office 'data' file is likely to be a fraction of the file size.

XML in brief

You will find many XML documents on your PC. XML files may be opened with a Web Browser which will Parse the Document and throw an error if it is not 'Well-formed', that is does not conform to simple rules:

XML header:

Documents start with **<?xml version="1.0" encoding="UTF-8"?>**- this is often omitted or ignored nowadays but encoding may be important.

'Tag' rules:

Tags are used to mark up sections - starting with a tag eg

<anyOldTagName>followed by data, and terminated with the closing tag **</anyOldTagName>**

Nested Tags structure:

Closing tags must occur within the document in reverse order to the order of opening. That is: Last tag opened must be Next one closed:

<firstTag>Put XML data Here if you want

<secondTag>or Here

<thirdTag>or even put data Here**</thirdTag>**

</secondTag>

</firstTag>

The section including and between tags comprises an 'Element'. Further Element information may be given by one or more 'Attributes' in the opening tag. Attribute values should always be in single or double quotes, example:

...<secondTag **myAttribute="this is a good Attribute within the Element"**>...

Allowable characters: Care must be taken that Data Content does not corrupt the Document and fool the Parser. Special characters such as **'<' '&'** and others must be 'escaped'. Any text editor can be used to create an XML document but XML editors and applications will prevent and detect errors.

Once XML is parsed and found to be 'well-formed', it may then be further tested or 'validated' against a 'DTD' (still widely used), or against an XML schema for content.

See for technical standards and for applications.

See **http://www.terry-comms.com** or **http://www.telform.co** for Mobile devices.