JUST WHAT IS XML —and how did it happen?

Prof. David F. Brailsford

School of Computer Science & IT
University of Nottingham, UK

Standard Generalised Markup Language (SGML)

- SGML is the parent of XML. It started as GML (within IBM) in late 70's. Charles Goldfarb was the major architect
- The vast range of possible document tags could not be described in a single specification.
- SGML is thus a *metalanguage* used to describe tags with which documents can be marked up.
- So SGML is not a fixed tagset. It describes a standard set of 'punctuation' for tags, plus char. sets to be used etc.
- SGML ISO standard dates from 1986.

SGML: A simple Tagged Memorandum

```
<MEMO>
<TO> Tony Blair </TO>
<FROM> The White House </FROM>
<BODY>
<P> The President says,
<Q> "Thank you for your support!" </Q>
</P>
</MEMO>
```

- Note omission of **</BODY>** i.e. ‘end of the body’
  This needs to be allowed in the ‘tag spec.’ (the DTD)
- Even permissible to omit start tags if DTD allows it!

SGML: A simple DTD for a memo

```
<!ELEMENT MEMO -- ((TO & FROM),BODY) >
<!ELEMENT TO -O (#PCDATA) >
<!ELEMENT FROM -O (#PCDATA) >
<!ELEMENT BODY -O (P)* >
<!ELEMENT P -O (#PCDATA | Q)* >
<!ELEMENT Q -- (#PCDATA) >
```

Document Type Definitions (DTDs)

- A set of tag definitions forms a DTD.
- The DTD's own metasyntax is *similar* to that of the tags—but not totally identical.
- SGML metasyntax similar in purpose to BNF for defining programming languages
- A DTD for a *memo* will obviously be different from a DTD for a *menu* or a DTD for a *report*.
- There are many existing DTDs (e.g. for HTML and in publishing) Also many 'in house' DTDs.

SGML Parsing

- SGML document needs to know character set to be used (e.g. UTF8)
- SGML requires DTD to be available at parsing time
- SGML parses a document using a given tagset with respect to the DTD that defines that tagset.
- SGML parser can check that DTD conforms to the SGML standard
- Optional *tag minimisation* can make SGML parser's job very difficult indeed.
- Reliable parsers for full SGML began to appear in the 80's (Usually cost a fortune ... used by document professionals only)

SGML: Structure vs. Appearance

- The memorandum's *appearance* cannot be determined from the tagged source.
- SGML and XML often used to define abstract 'structural' markup—but not always so (see SVG later on).
- **Tag names have no intrinsic meaning, but all programs must agree on the *semantics* of what they mean**
- The structure of the document as defined by the tags makes it easy to apply database technology to class of documents defined in DTD.

SGML: 'Multi-purposing' an SGML document

- Possible to start with SGML tagged doct.
  Process SGML to Quark or Pagemaker for typesetting.
- But there is a big 'semantic gap' between
  SGML and PostScript (say).
- Hard to control typeset details from SGML
  (DTD gets enormous if you try).
- SGML best as source of structured data which
  can then be manipulated in a variety of ways.
- Styling best left to stylesheets (e.g. CSS2
  or XSL-FO for the Web).

SGML applications

- A specific tagset specified via a SGML DTD is called an *application* of SGML e.g. CALS, TEI.
- Some WYSIWYG software can generate SGML tagged docs. from their own internal data structures
- Examples include software from ArborText, SoftQuad, Adobe (Frame + SGML) etc.
- This software is *never* cheap! Writing full SGML parsers is *hard*!
- HTML was the first 'mass market' SGML application. People at last realised what SGML notation could achieve.

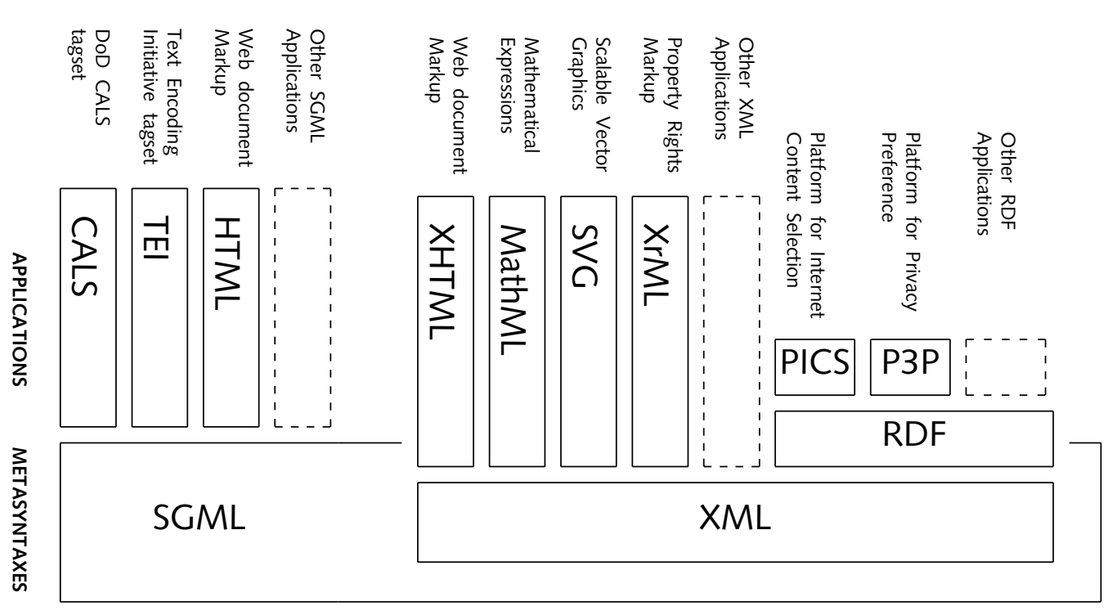


Diagram to show SGML/XML *application* and *subset* relationships.

Note that:

- XML is a *subset* of SGML
- HTML etc. are *applications* of SGML
- XHTML etc. are *applications* of XML
- RDF is an *application* of XML
- PICS, P3P etc. are *applications* of RDF

What about HTML?

- HTML adopted SGML metasyntax from the outset so it's an *application* of SGML.
- But it is essentially a *fixed* tagset (though this became arbitrarily extended, in different ways, by MS Internet Explorer and by NetScape).
- In early days Tim Berners-Lee and browser vendors didn't fully realise importance of having DTD for HTML
- Net result was chaos. IE and Netscape have different tags and different minimisation possibilities
- Allowing 'overlapping hierarchies' as well as omitted end tags is *deadly*. More than 95% of Web HTML is illegal!

More problems in HTML

- Lack of a DTD and SGML knowledge means IE and Netscape allow overlapping hierarchies
  e.g **`<P> <BOLD> ... </P> </BOLD>`**
- SGML is essentially a specification for tree-structured docts. with nested features.
- Overlapping hierarchies coupled with end tag omissions are BAD news.
- HTMLs problems led to a call for it to be re-specified in SGML and for DTD to be available to browsers and rigidly enforced!
- Browser vendors said emphatic *No!* to full SGML parser in all browsers. So – enter XML.

SGML/XML: What is XML?

- Work started in 1996 on e**X**tensible **M**arkup **L**anguage.
- 80 SGML experts under aegis of W3C and chaired by Jon Bosak (Sun Microsystems) aided by Tim Bray.
- XML to be easy to parse and yet be proper subset of SGML
- XML parser to be able to check *well-formedness* (without DTD) and *validity* (with DTD).
- Must support stylesheet mechanism and syntax for hyperlinking and *namespaces* e.g **`<memo:from>`**

SGML/XML: XML design goals

(1) XML to be easy to use over Internet

(2) XML to support wide variety of apps.

(3) XML to be proper subset of SGML

(4) Must be easy to write progs. that process XML

(5) XML 'optional features' to be kept to minimum.

SGML/XML: More XML design goals

(6) XML to be human legible and reasonably clear

(7) XML design to be prepared quickly

(8) Design of XML to be formal and concise

(9) XML docs. to be easy to create

(10) Terseness in XML markup to be of minimal importance

Some SGML/XML differences

- In XML start and end tags must **always** be present.
- This allows well-formedness check without a DTD
- No ‘comments within comments’ or ‘comments within element declarations’. *Nightmare* to parse in SGML.
- Element attributes e.g. **`colour="blue"`** must be quoted
- **`&`** connector forbidden in element declarations. Must use **`,`** and **`|`** only.
- Lots of other more detailed differences (e.g. no ‘inclusions’ and ‘exclusions’)

Revised (XML) DTD for a memo

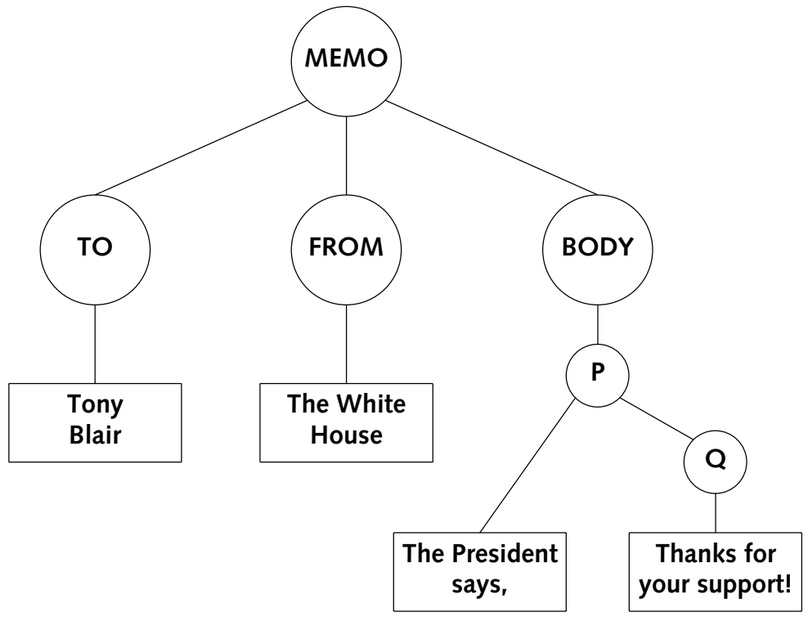
```
<!-- Note that -O for ‘omittability’ now absent-->
<!-- Note that & has vanished in MEMO element declaration-->
<!ELEMENT MEMO (TO,FROM,BODY) >
<!ELEMENT TO (#PCDATA) >
<!ELEMENT FROM (#PCDATA) >
<!ELEMENT BODY (P)* >
<!ELEMENT P (#PCDATA | Q)* >
<!ELEMENT Q (#PCDATA) >
```

Revised XML-compliant memorandum

```
<?xml version="1.0"?>
<!DOCTYPE MEMO SYSTEM "memo.dtd">
<MEMO>
<TO> Tony Blair </TO>
<FROM> The White House </FROM>
<BODY>
<P> The President says,
<Q> "Thanks for your support!" </Q>
</P>
</BODY>
<!-- Notice the above line now essential -->
</MEMO>
```

- Note the *required markup declaration* (RMD) for version of XML to be used
- The next line (the *document type declaration*) must stipulate where DTD is to be found if the doct. is to be *validated*

The XHTML memo as a tree



- NOTE: All valid XML docts. have a tree structure.

XML ATTRIBUTES

- Attributes of XML elements are very similar to ‘parameters’ or ‘arguments’ in conventional programming languages.
- The list of attributes (if any) for an element is given as an ATTLIST for the named element within the DTD.
- When tagging up the document itself the attribute values for a tag are quoted within the opening tag of the element, using a **`name="value"`** syntax.
- As an example:

```
<T-SHIRT COLOUR="red" SIZE="xlarge">My Armani T-shirt</T-SHIRT>
```

- Also can specify whether attributes are **#FIXED** **#REQUIRED** or **#IMPLIED**.

DTD drawbacks

- DTDs were part of original SGML. Their specification syntax is similar to SGML itself but not quite the same.
- Drawback of DTD is that it lacks *type information*
  E.g. can't specify that a date be two digits in range 0–31.
- Tendency to group all data as either **#PCDATA** or as **CDATA**. Often not precise enough
- Attributes are useful but can't themselves be structured
  It would also be nice to give type information to attributes beyond the enumerated types already there.
- For these reasons *XML Schemas* are now gaining ground

XML Schemas

- These specify the legal XML tags and attributes but in an XML-compliant syntax.
- Therefore XML parsers can, in principle, parse schemas without need for DTD ‘metametasyntax’.
- Schemas allow attributes etc. to be typed.
- Schema proposals have now ‘settled down’. Consult **`www.w3c.org`** for recommendation proposal.

EXTENDED XML DTD FOR ‘MEMO’

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT memo (header, body)>
<!ATTLIST memo
    priority (low | medium | high | urgent) #IMPLIED
>
<!ELEMENT header (date, from+, to+, subject, reference?)>
<!ELEMENT body (para | quote | list)+>
<!ELEMENT date (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT reference (#PCDATA)>
<!ELEMENT subject (#PCDATA)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT list (item)+>
<!ATTLIST list
    type (ordered | unordered) #IMPLIED
>
<!ELEMENT item (#PCDATA)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT quote (para | quote | list)+>
```

FROM HTML → XHTML

- There is a strong push from W3C to clean up HTML into XHTML.
- Most obvious change is putting in *all* end tags.
- Empty elements such as **
** must either appear as **
</BR>** or as **
**
- Browsers are generally happy with XHTML—but Netscape still needs internal space within self-closing tags i.e. **
**
- W3C Website has facility for checking whether your own Web pages are XML compliant (!)

SIMPLE WEB PAGE IN HTML

```
<TITLE>David Brailsford's Home Page</TITLE>
<H1>Professor David Brailsford</H1>
    <TITLE>David Brailsford's Home Page</TITLE>
    <H1>Professor David Brailsford</H1>
<IMG SRC="dfbphoto.jpg" height="250">
<ADDRESS>
Department of Computer Science<BR>
University of Nottingham<BR>
Nottingham NG7 2RD<BR>
UK<BR>
<BR>
+44 115 951 4251<BR>
dfb@cs.nott.ac.uk
</ADDRESS>
```

REVISED XHTML COMPLIANT WEB PAGE

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
Transitional//EN""DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/TR/xhtml1">
<head>
<TITLE>David Brailsford's Home Page</TITLE>
</head>
<body bgcolor="#FFFFCC">
<H1>Professor David Brailsford</H1>
<IMG SRC="dfbphoto.jpg" height="250"
    alt="photo of David Brailsford" />
<ADDRESS>
School of Computer Science and IT<BR />
University of Nottingham<BR />
Jubilee Campus<BR />
Nottingham NG8 1BB<BR />
UK<BR />
<BR />
+44 115 951 4251<BR />
dfb@cs.nott.ac.uk
</ADDRESS>
</body>
</html>
```

THE XML TREE MODEL

- Remember that a well-formed XML document is essentially a linearised (flattened) tree.
- All **`ELEMENT`**s become nodes in the tree. A sequence of elements is a left-to-right node sequence.
- Attributes are a way of investing each tree node with ‘properties’.
- DTDs (and Schemas even more so) are an abstract way of specifying ‘all valid trees for all valid documents’
- XLink, XPath, Xpointer, XSL, XSLT and DOM all rely on this tree property.

XSL-FO and XSLT

- The XML Stylesheet Language (XSL) is based on an earlier SGML-related activity called DSSSL.
- The idea is to allow an XML-based syntax for specifying appearance of a document (e.g. on Web pages).
- Existing HTML styling standard is Cascading Style Sheets version 2 (CSS2) but this is *not* an XML syntax.
- Sophisticated formatting needs not just to add extra attributes to a tree node. It often needs to *re-shape* the tree.
- XSL-FO ('formatting objects') is the candidate styling recommendation of W3C.

XSLT

- The XSL committee soon realised that the act of 'styling' was really reshaping an XML tree.
- So a sub-committee developed idea of XSLT for tree-to-tree transformations. But the transformation itself is expressed in XSLT (an XML application)
- XSLT draft came out *before* XSL-FO
- An XSLT script could , for example, map a marked up memo from its own tagset into XHTML
- Very handy because can be done on the server. Eg. create an XML document of PC prices etc. for a Web page from a database using a Java/Perl/VB script
- The use XSLT script to convert these XML marked prices to XHTML to be served up to everyone.

Interpret XML client side in your browser?

- Latest IE and Netscape beginning to be able to do this
- Download e.g. the ‘memo’ XML, or the ‘PC prices’ XML direct to the user with CSS or XSL-FO stylesheet
- Cuts dowen server burden — but clients must have up to date browser.
- Currently XML + CSS works OK in Explorer.
- Currently XSL-FO is at ‘proposed recommendation’ stage Not implemented in browser clients yet.

XML Applications

- Explosive growth in these since 1997. Now hundreds of them. Some examples:

  — SVG Scalable Vector Graphics for the Web (currently needs browser plugin from Adobe)

  — Extensible Business Reporting (XBRL)

  — Financial Products Markup Language (FPML)

  — LegalXML, NewsML, MathML etc. etc. etc

Finally ...

- XML gives us a common set of symbols and punctuation for markup.
- Strict rules for XML metasyntax means we can have general purpose XML parsers. But XML can be unwieldy for things that are not at all 'tree structured'.
- XML is not just for the Web! Datbase to database conversion is now a big use for XML tagsets
  All sorts of other 're-purposing' too
- Take Care! All of these new 'MLs' are tagsets that are XML applications. NOT new metasyntaxes
- Take Care! The (meta)syntax is now sorted but what about the tag *semantics*. That's the hard bit ...