

# Extensible Markup Language (XML) - Principles

Michel Goossens  
IT/API

XML Detector Description Workshop - 14 April 2000

## XML - first a few examples

<pre>&lt;?xml version="1.0"?&gt; &lt;memo xml:lang="en"&gt;   &lt;to&gt;Detector Group&lt;/to&gt;   &lt;from&gt;Steven Goldfarb&lt;/from&gt;   &lt;date&gt;10/04/2000&lt;/date&gt;   &lt;body&gt;     &lt;p&gt;Meeting confirmed       for &lt;em&gt;Friday&lt;/em&gt; 2pm.     &lt;/p&gt;   &lt;/body&gt; &lt;/memo&gt;</pre>	<pre>&lt;?xml version="1.0" encoding="ISO-8859-1"?&gt; &lt;para&gt;L'équation d'Einstein est   &lt;math&gt;     &lt;mi&gt;E&lt;/mi&gt;     &lt;mo&gt;=&lt;/mo&gt;     &lt;mrow&gt;       &lt;mi&gt;m&lt;/mi&gt;       &lt;mo&gt;&amp;InvisibleTimes;&lt;/mo&gt;       &lt;msup&gt;         &lt;mi&gt;c&lt;/mi&gt;&lt;mn&gt;2&lt;/mn&gt;       &lt;/msup&gt;     &lt;/mrow&gt;   &lt;/math&gt;. &lt;/para&gt;</pre>
<pre>&lt;?xml version="1.0"?&gt; &lt;customer&gt;   &lt;name&gt;...&lt;/name&gt;   &lt;order&gt;...&lt;/order&gt;   &lt;order&gt;...&lt;/order&gt; &lt;/customer&gt;</pre>	<pre>&lt;?xml version="1.0"?&gt; &lt;par&gt;   &lt;video id="a" src="movie1" /&gt;   &lt;seq&gt;     &lt;audio src="audio1" /&gt;     &lt;audio begin="5s" src="audio2"/&gt;   &lt;/seq&gt; &lt;/par&gt;</pre>

## What is XML?

- document format;
- data format;
- meta language;
- method for structuring information;
- activity coordinated by the *World Wide Web Consortium* (W3C).

## Why XML?

## I18N electronic publishing

- platform independent;
- language independent;
- media independent.

## Web applications

- exchange format between databases;
- managing a collection of documents;
- distributed document handling;
- client-side data manipulation;
- commercial transactions.

## The origins of XML

- 30 years of research in the area of document handling;
- important landmarks were: GML, Scribe;
- in ISO international standard: SGML (ISO 8879);
- 15 years experience with SGML;
- a popular SGML application: HTML;
- need for a richer and more specialised set of languages;
- a simplified version of SGML for the Web: XML.

## XML and the users

- can be adapted to all sorts of applications;
- uses a readable text-based syntax;
- easily parsable by machines;
- light-weight deployment;
- full I18N support;
- guaranteed interoperability: content, style, behaviour.

## XML and W3C

### World Wide Web Consortium

- over 400 members world-wide;
- develop the base technologies for the Web;
- openness and consensus.

### XML at the W3C

- **XML Core WG:** XML syntax (XML, namespaces, *Xinclude*, *Xbase*);
- **XML Linking WG:** next generation hypertext (*Xpointer*, *Xlink*);
- **XML Schema WG:** successor for DTDs (much more general);
- **XML Query WG:** define a query language for XML (gather requirements);
- **XSL WG:** transformation and style sheet language (*XPath*, *XSLT*, *XSL-FO*);
- **DOM WG:** object-oriented API for XML.

## XML fundamentals

- XML is not a language, but a meta-language;
- XML supports an infinite number of languages for science, industry, business, etc.;
- all such XML-based languages can be handled by the same light-weight parser;
- two language syntaxes:
  - markup language;
  - document type definition language (DTD).

## XML as a markup language

- tree structure;
- element types;
- attributes: name-value pairs;
- identification of element instances by unique ID;
- entities;
- full I18N for names and content via Unicode (ISO/IEC 10646).

### Example

```
<?xml version="1.0"?>
<memo xml:lang="en" security="cern">
  <to>Detector Group</to>
  <from>Steven Goldfarb</from>
  <date>10/04/2000</date>
  <body>
    <p>Meeting confirmed
      for <em>Friday</em> 2pm.
    </p>
  </body>
</memo>
```

## The DTD syntax

- document type definition (defines *type* of documents);
- simplified version of SGML;
- defines the *document grammar*;
- defines element names, attributes and entities;
- makes it possible to verify documents.

## DTD example

```

<!ELEMENT memo      (to, from, date, subject?, body) >
<!ATTLIST memo      security (public | cern ) 'public'>
<!ELEMENT to        (#PCDATA) >
<!ELEMENT from      (#PCDATA) >
<!ELEMENT date      (#PCDATA) >
<!ELEMENT subject   (#PCDATA) >
<!ELEMENT body      (p+) >
<!ELEMENT p         (#PCDATA | em)* >
<!ELEMENT em        (#PCDATA) >

```

## Well-formed and valid documents

- a DTD is not obligatory;
- a document must at least be *well-formed* :
- elements correctly nested inside root element.
- if the document references a DTD it must also be *valid*:
- respect the specifications in the DTD.

## XML Schemas

*DTDs have limitations:*

- too constraining (everything must be defined);
- no content typing (e.g., date, numeric range);
- non-XML syntax.

*XML Schemas*

- structural constraints: type definitions, classes and sub-classes, content models, import, include;
- data typing: `string`, `boolean`, `binary`, `uri`, `number`, `integer`, `decimal`, `real`, `date`, `time`, `timePeriod`;
- use XML syntax.

## Remarks

- XML only defines a syntax, no semantics;
- XML element names have no pre-defined meaning nor associated presentation;
- XML defines only content and structure, no behaviour.

## XML - Applications

### W3C specifications based on XML

- **MathML**: mathematical expressions;
- **SMIL**: Synchronized Multimedia Integration Languages;

- **SVG**: 2-D vector graphics;
- **XHTML**: XML definition of HTML;
- **RDF**: metadata;
- **XSL**: stylesheet language.

## Other languages using XML

Dozens of XML-based languages exist already, and more are created almost daily.

- **WAP**: Wireless Application Protocol;
- **AML**: Astronomical Markup Language;
- **BIOML**: Biopolymer Markup Language;
- **GEML**: Gene Expression Markup Language;
- **X3D**: 3-D graphics, virtual reality (VRML);
- **MusiXML**: music notation;
- **VXML**: vocal dialogues;
- **CML**: Chemical Markup Language;
- **FpML**: Financial products Markup Language;
- **TEI**: Text Encoding Initiative;
- **DocBook**: Computer documentation;
- etc.

### MathML example

$$(a + b)^2$$

```
<msup>
  <mrow>
    <mo>( </mo>
    <mi>a</mi>
    <mo>+</mo>
    <mi>b</mi>
    <mo>)</mo>
  </mrow>
  <mn>2</mn>
</msup>

<apply>
  <power/>
  <apply>
    <plus/>
    <ci>a</ci>
    <ci>b</ci>
  </apply>
  <cn>2</cn>
</apply>
```

### SMIL example

```
<par>
  <video id="a" begin="6s" src="movie1" />
  
  <seq>
    <audio src="audio1" />
    <audio begin="5s" src="audio2" />
  </seq>
</par>
```

```

      par
|-----|
| 6s      movie1 |
|<----->|-----|
|      4s      img |
|<---->|-----|
| audio1      5s      audio2 |
|-----|<----->|-----|
```

# XML Namespaces

- modular design;
- use several XML grammars in the same document (without element name clashes).

Example :

```
<?xml version='1.0'?>
...
<particles>
  <!-- Default namespace is set to HTML -->
  <table xmlns='http://www.w3.org/TR/REC-html40'>
    <tr><td>Particle</td><td>Mass</td><td>Details</td></tr>
    <tr>
      <td><em>neutron</em></td>
      <td>939.56 MeV</td>
      <td>
        <pdg xmlns="mypdgnamespace">
          <!-- HTML namespace is no longer used -->
          <quarks>udd</quarks><lifetime>886.7 s</lifetime>
          <decay>p,e,anue</decay>
        </pdg>
      </td>
    </tr>
  </table>
</particles>
...
```

## Style sheets

**Separate** document style from content/structure:

- without modifications a document can be presented in different ways depending upon the reader, the output medium, the target application;
- a set of documents can be presented in a homogeneous way (house style);
- this has the advantage that one can maintain documents for the Web independent on platforms, performance, etc.

## CSS: Cascading Style Sheets

Work with XML as well as with HTML.

- **CSS 1:** basic style components: fonts, colour, size, spacing, margins, etc.;
- **CSS 2:** fine-tune placement, dynamic loading of fonts, printing, audio output, I18N, tables, numbering, etc.;
- **CSS 3:** further extensions for CSS (selectors, user interface, colour specs) are being specified by the W3C styling WG.

## XSL: Extensible Stylesheet Language

Works only with XML.

- **XPath:** address parts of an XML document;
- **XSLT:** transform XML documents; can generate toc, index, etc.;

- **XSL-FO:** abstract output format description language. It defines formatting objects and their associated style attributes in XML syntax. It can handle complex page structures, tables, running headings, etc. Includes all functions of CSS and DSSSL.

## Example of an XSLT transformation

Transform HTML element into

An XML XSLT stylesheet is a series of match patterns to select the component in the source tree to be transformed (using the XPath language). Then, for a given selection a template indicates the action to be taken. For instance, one can transform the HTML element into using the following XSLT command:

```
<xsl:template match="h1">
  <xsl:element name="heading">
    <xsl:apply-templates/>
  </xsl:element>
</xsl:template>
```

**Input:**\_\_

**output:**\_\_

## Conclusion

- XML is a flexible language for describing structured data and international documents;
- XML provides for optimal interoperability by its full integration with the Web;
- XML is used by many dozens of applications in all areas of human activity and can be handled by an ever-growing set of (mostly) free tools;
- XML (markup language and DTD), Schemas and XSL are only the first of a series of members of the XML family of languages.

"XML is the ASCII of the 21st century" - Tim Bray

More information about XML:

- [W3C XML activity](#);
- [OASIS XML pages](#);
- [XML news and resources](#).