INFORMATION TECHNOLOGY INSTITUTE

XML
&
Allied Technologies

What is XML?

- eXtensible Markup Language
- W3C Initiative.
- XML describes a syntax for marking up documents so that complex structures may be easily described.
- XML is one of the most widely-used formats for sharing structured information today between programs both locally and across networks.
- It provides a standard format for computer documents that is flexible to be customized for diverse domains as Websites & Electronic data interchange

XML Document

- XML is a mark up language:

```
<?xml version="1.0" encoding="UTF-8" ?>
<note>
   <to>ahmed</to>
   <from>mohamed</from>
   <heading>Reminder</heading>
   <content>watch the football match </content>
 </note>
```

Markup Languages

- The word “Markup” is derived from the printing industry:
  - Detailed stylistic instructions for typesetting
  - Usually hand-written on the copy (eg underlining some text that is to be set in italics).
- Markup languages do the same job for computerised documentation systems.
- Markup adds logical structure to a document, or indicates how it is to be laid out (on paper or screen).

Markup Languages (cont.)

- For example (in HTML)
  - This is <B>bold</B> and this is <I>italic</I>
  - <TITLE>This is the title.</TITLE>
- Examples of Markup languages:
  - SGML
  - HTML
  - XML

SGML - History

- Standard Generalised Markup Language
- 1969 - GML from IBM
- 1980 SGML first published
- 1986 - ISO standard
- SGML: is a standard for how to specify a document markup language or tag set.

SGML

- SGML is itself a document type definition (***DTD***). It is a standard for how to specify a tag set.
- SGML is not in itself a document language, but a description of how to specify language. It is ***metalanguage***.
- SGML documents contain **structural elements** that can be described without consideration of how they are displayed.
- We use SGML to write other languages that are used by programmers
- **HTML** is an SGML application.

The need for extensibility

Problems with SGML:

- Complexity.
- SGML specifications was too large to learn to write with it documents
- No S/W has fully implemented the specs.

Problems with HTML:

- Fixed set of tags.
- HTML was not designed for current use
- Poor at representing specialised data: Maths, Music

The need for extensibility (cont.)

- **Linking protocols are crude**
  - links are all simple one way pointers
  - no distinction between different link types
- **HTML is a display format**
  - good for rendering information - but...
  - it contains no information about document structure
- **Style and content are intrinsically linked**
  - large scale maintenance is difficult
  - information may be lost because only its appearance is described, not its meaning (semantics)

What XML is?!

- XML is based upon SGML, but is substantially simplified for use on the WWW.
- XML is a metalanguage:
  - Doesn’t have a fixed set of tags and elements.
  - Syntax may *optionally* be described by a DTD
  - A DTD specifies the legal markups and when & how the mark up may be used
    - **Valid documents** - have a DTD
    - **Well formed documents** do not have a DTD
- Style and content are completely separate
    - **XML documents** contain **Content**
    - **Style** is specified by **Stylesheets**

What XML is?! (cont.)

- **XML is :**
  - Language independant.
  - platform independent.
  - Application independent.
  - Foundation for several next-gen Web Technologies (XHTML , RSS, AJAX, Web services)

Advantages of XML

- **XML :**
  - Uses human ,not computer Language.
  - Is readable and understandable.
  - 100% portable.
  - Extensible
- **XML can be used in Data manipulation**
  - Data from relational data can be converted to XML for easy manipulation

XML Document Example

```
<?XML version="1.0"?>
<!DOCTYPE memo PUBLIC "memo.dtd">
<!--A very simple XML document -->
<MEMO>
   <FROM>A.M.E</FROM>
   <TO>A.N. Student</TO>
   <SUBJECT>Your Work</SUBJECT>
   <DATE>14th February, 2000</DATE>
   <MESSAGE>
This is to confirm that I received your work
   </MESSAGE>
</MEMO>
```

XML Document Example (cont.)

```
<?XML version="1.0"?>
<!DOCTYPE memo PUBLIC "memo.dtd">
<!--A very simple XML document -->
<MEMO>
   <FROM>A.M.E</FROM>
   <TO>A.N. Student</TO>
   <SUBJECT>Your Work</SUBJECT>
   <DATE
   <MESS
This is to
   </MES
</MEMO
```

```
<!ELEMENT MEMO (FROM, TO, SUBJECT, DATE, MESSAGE)>
<!ELEMENT FROM (#PCDATA)>
<!ELEMENT TO (#PCDATA)>
<!ELEMENT SUBJECT (#PCDATA)>
<!ELEMENT DATE (#PCDATA)>
<!ELEMENT MESSAGE (#PCDATA)>
<!ELEMENT P (#PCDATA)>
```

Specialised XML Applications

- **MathML** - Mathematical Equations
- **CML** - Chemistry
- **MusicML** - Sheet Music
- **FpML** - Financial Products
- **RETML** - Real Estate Transactions
- **SMIL** - Synchronised Multimedia Integeration

What XML is not

- No compiler that generates executables.XML is a text language.
- XML is not a network protocol, i.e another S/W has to do the sending while data can be stored as XML.
- XML is not a database, data can be stored in an XML format,but the engine has to exist.

Usage of XML

- **Web publishing**
- **Web Searching and automating web tasks**
- **Metadata Applications**

  It’s so easy to build on XML applications.
- **Standardization**

XML provides a standard method to access information making it easier for manipulation

Content vs. Style

- XML tags contain meaning *not* appearance.
- This allows extra information to be extracted
- Consider the example of the scientific names of animals.
  - scientific names are in Latin and they are always printed in italics

The *scientific* name of the domestic dog is *Canis familiaris*, and of the domestic cat is *Felis catus*.

Content vs. Style

In HTML :

```
<P>The <I>scientific</I>
name of the domestic dog
is <I>Canis familiaris</I>,
and of the domestic cat
is <I>Felis catus.</I></P>
```

NB:

there is no distinction between scientific names and emphasis.

The *scientific* name of the domestic dog is *Canis familiaris*, and of the domestic cat is *Felis catus*.

Content vs. Style

In XML :

```
<p>The scientific
name of the domestic dog
is <Dog>Canis familiaris</Dog>,
and of the domestic cat
is <Cat>Felis catus.</Cat></p>
```

NB emphasis and scientific names are different tags. They may both be displayed as italic, but they can be treated separately.

The *scientific* name of the domestic dog is *Canis familiaris*, and of the domestic cat is *Felis catus*.

Stylesheets

- Style in XML is defined by stylesheets
- Stylesheets define the **physical appearance** of a document, and its **behaviour**

- Stylesheet languages
  - **CSS (Cascading StyleSheets) - developed for HTML**
  - **XSL - developed specifically for XML**

CSS

- Cascading Stylesheets (CSS) is a language for defining stylesheets that was developed for HTML
- W3C recommendation, and now very widely used
- CSS defines appearance of a document - eg:
  - Fonts & appearance of text
  - Colours
  - Layout (eg margins, indentation, positioning etc)
  - Behaviour (extremely limited extent)

Linking CSS to a document

- CSS may be embedded into HTML

  (using <STYLE> tag or stored in a separate file
- One CSS file can control the appearance of any number of HTML documents
- Changing a CSS file will change the appearance of *all* documents that use this file
- CSS files are attached to HTML using the <LINK> tag in the <HEAD>

```
<LINK HREF=“fname.css” REL="stylesheet"
   TYPE="text/css”>
```

CSS & XSL with XML

- CSS may be used with XML *exactly* as it is used with HTML.
- CSS can control appearance, but not behaviour.
- **XSL** (Extensible Style Language) is a sophisticated stylesheet programming language developed specifically for XML.
- CSS and XSL are complementary - CSS can be used *within* XSL.