

An Introduction to XML and XHTML

Group 3, CS 174

Aram, Ardee

Kimpo, Phillip Jr.

Lucero, Adelaida Sophia Marie

Roque, Jonas Fabian

(Developers of *MEDS/Pintig*)

Submitted to:

Ms. Joyce Emballa Avestro


Table of Contents

XML	page 3
XHTML	page 9
XML and XHTML Applications.....	page 15
References	page 20

XML

Basic HTML (HyperText Markup Language) does not provide any structure to Web pages, and the formatting is mixed with the content. To allow Web pages to be structured for automated processing (e.g. electronic commerce), the World Wide Web Consortium (W3C) developed an enhancement to HTML. The result were two new languages; one was XSL (eXtensible Style Language), and the other was XML (**eXtensible Markup Language**), a system for defining, validating, and sharing document formats on the Web.

I. History

The W3C, an organization devoted to developing the Web and standardizing protocols, formed an XML Working Group chaired by  Jon Bosak of Sun Microsystems in 1996. Several key industry players who were also included in the working group were Adobe, Hewlett-Packard, Microsoft, Netscape, and Fuji Xerox.

The group published a working draft for XML in November of the same year. Two years later, the W3C announced the release of the XML 1.0 specification.

The year 1999 found the release of two W3C Recommendations on XML. The first was entitled *Namespaces on XML*, and the other was *Associating Stylesheets with XML documents*. In January of 2001, the Internet Engineering Task Force (IETF) released a Proposed Standard on *XML Media Types*.

II. Profile

XML is an open, human-readable text format derived from the Standard Generalized Markup Language (SGML). Originally meant for large-scale electronic publishing, XML is now being used in the exchange of various types of data on the Web and elsewhere. It is also becoming a language of choice for communication between application programs.

The XML Working Group's design goals (taken from the W3C website) for XML were:

- 1) XML shall be straightforwardly usable over the Internet.
- 2) XML shall support a wide variety of applications.
- 3) XML shall be compatible with SGML.
- 4) It shall be easy to write programs that process XML documents.
- 5) The number of optional features in XML is to be kept to the absolute minimum, ideally zero.
- 6) XML documents should be human-legible and reasonably clear.
- 7) The XML design should be prepared quickly.
- 8) The design of XML shall be formal and concise.
- 9) XML documents shall be easy to create.
- 10) Terseness in XML markup is of minimal importance.

The markup language describes XML documents, which are a class of data objects. Moreover, XML also describes the behavior of software modules called XML processors. These are used to read XML documents and provide access to their content and structure.

III. Basic Development

Having an informed choice on whether or not to use XML on a web application, we shall now see how to create an XML document. In this text, we use the XML version 1.0.

For those familiar enough with HTML, an XML document will appear similar. However, XML is much stricter in than HTML, which we should see later on. For now, let us view a sample XML file for you to have an idea of the structure of an XML file.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mobileTechClass>
  <groupName id="1234234">CS174-Pintig</groupName>
  <description>Boronggo</description>
  <members>
    <name>Ardee Aram</name>
    <name>Phillip Kimpo</name>
    <name>Ia Lucero</name>
    <name>Jonas Roque</name>
  </members>
</mobileTechClass>
```

An XML document has the following characteristics:

- 1) Its smallest data unit is the "element". An element typically consists of two tags, a start tag and an end tag, possibly surrounding text and other elements. The start tag consists of a name surrounded by angle brackets, like "<name>"; the end tag consists of the same name surrounded by angle brackets, but with a forward slash preceding the name, like "</name>".

- 2) Elements could "nest", or contain another element. This gives XML the power to express hierarchical and recursive data structures.

- 3) An element could contain an "attribute", or a name-value pair inside the start tag of the element. An attribute is of the form name="value". Attributes provide flexibility and more data-expression power to XML.

4) An XML document must have an XML header of the form "<?xml version="1.0" encoding="ISO-8859-1"?>". The version attribute refers to the current XML Specification version being used, and the encoding refers to the character set being used in the document.

We mentioned a while ago that there are strict XML rules to be followed when creating an XML document. In fact, there are two measures of "correctness" for an XML document, its "well-formed"-ness, and its validity. For an XML to be well formed, it must follow these simple rules:

1) All XML elements must have a closing tag. There are cases when an element does not have a "content", or the text data between tags. In this case, there is no point to have a start-end tag pair, and the <tag /> format is used.

2) XML tags are case sensitive, and XML prefers lowercase tag names.

3) All XML elements must be properly nested. A properly nested element means that all child tags must end before their parent tag. Thus, <i>Hello</i> is not well formed.

4) All XML documents must have a "root element". There must be only one element that contains all other elements in a document. In our example above, it is <mobileTechClass>.

5) Attribute values must always be quoted. With XML, it is illegal to omit quotation marks around attribute values.

The W3C has defined a document on the specific criteria of a well-formed document.

Our previous XML example is well formed, but it is not valid. What is XML validity? Before anything else, we stress here that XML validity is a relative characteristic to some another external "model" that would "bless" our document

with validity, as opposed to the absoluteness of characteristic of being well formed. Thus, we say that an XML is valid for a certain document when that XML follows a certain model structure of information specified by the author on that document. That model structure is referred to as a "schema".

Why introduce validity? As we can see, our well-formed example above may be added with any tag you like, and by adhering to the above rules, our XML-based application will never complain. However, adding `<menuPrice>` to the document does not make any sense regarding the current context. Thus, we must be able to control what fields are possible, what are not, what are the possible children and attributes of each element, and so on. By doing this, we are sure that the document adheres to the specifications that the author has used, and the XML-based application will be able to easily recognize its needed data inside an XML document.

There are many kinds of XML validity schema languages; two of the most popular are the Data Type Definition (DTD) and the XML Schema Definition (XSD). Other options include RelaxNG, Document Structure Description (DSD), and Schematron. Though in-depth discussion of schemas is beyond the scope of XML and XHTML, we shall demonstrate how to add DTD validation to our sample XML document. By inserting this line of code:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE mobileTechClass SYSTEM "MamJoyce.dtd">
<mobileTechClass>
  <groupName id="1234234">CS174-Pintig</groupName>
  <description>Boronggo</description>
  <members>
    <name>Ardee Aram</name>
    <name>Phillip Kimpo</name>
    <name>Ia Lucero</name>
    <name>Jonas Roque</name>
  </members>
</mobileTechClass>
```

We declare that the XML document conforms to the data structure specified in "MamJoyce.dtd". You may run an external XML validator application (most XML parsers support XML validation) or even a browser (if it supports XML validity checking) to check if the document does indeed conform with MamJoyce.dtd.

Other validation schemas have their own format of declaring their schema definitions.

XHTML

Many Web pages today are poorly written. Syntactically incorrect HTML code may work in most browsers even if it does not follow HTML rules. Browsers employ heuristics to deal with these flawed Web pages; however, Web-enabled wireless devices (such as PDAs) cannot accommodate these hefty Web browsers. The next step in HTML's evolution comes in the form of XHTML (**eXtended Hypertext Markup Language**), which is basically a combination of HTML and XML.

I. History

As with XML, the W3C was the force behind XHTML's development. When XML was introduced, a two-day workshop was held to discuss whether a new version of HTML in XML was needed. The answer was a resounding "Yes."

The first W3C Recommendation to be published was *XHTML 1.0*, which reformulates HTML 4.0.1 in XML and combines the strengths of the two markup languages. The recommendation followed from earlier work on HTML 4.0.1, HTML 4.0, HTML 3.2, and HTML 2.0.

The second W3C Recommendation was *XHTML Basic*. It boasts of images, forms, basic tables, and object support. XHTML Basic is designed for Web clients that do not support the full set of XHTML features; examples of these clients are mobile phones, PDAs, pagers, and set-top boxes.

The third W3C Recommendation to come out was on the *Modularization of XHTML*. It provides a means for extending and creating subsets for XHTML. The modular design introduced by this recommendation underscores the invalidity of

the “one-size-fits-all” approach nowadays, especially with the advent of Web browsers that vary in capabilities (e.g. cellphone browser vs. desktop PC browser).

The fourth W3C Recommendation was *XHTML 1.1 (Module Based XHTML)*. Here, a new XHTML document type is defined based on the modular design of the third XHTML Recommendation.

II. Profile

XHTML is a family of current and future document types and modules that contains all of the HTML 4.0.1 elements combined with XML syntax. XHTML is classified as an XML Application, and thus possess many XML features.

XHTML, described by Tanenbaum as a “language that is Very Picky”, differs itself from HTML by its stricter syntax.

The current version of XHTML that is supported by browsers is XHTML 1.0, a W3C Recommendation discussed earlier. There are three variants to XHTML 1.0:

- 1) XHTML 1.0 Strict – used for exceptionally clean structural markup; the CSS (Cascading Style Sheet) language can be used with this variant to get the desired font, color, and layout effects
- 2) XHTML 1.0 Transitional – best option for Web authors with webpages meant for general public access; takes advantage of XHTML features including style sheets
- 3) XHTML 1.0 Frameset – used to partition the Web browser window into two or more frames

III. Basic Development

XHTML is not very different from HTML 4.01, so bringing the code up to 4.01 standards is a very good start to XHTML. Here are the most important differences between the XHTML and HTML 4.0.1:

1) Elements must be properly nested

Incorrect:

```
<b><i>XHTML is not very different form HTML</b></i>
```

Correct:

```
<b><i>XHTML is not very different form HTML</i></b>
```

2) Documents must be well formed

All XHTML elements must be nested within the <html> root element. All other elements can have children. The latter must be in pairs and correctly nested within their parent element.

Correct:

```
<html>
  <head> ... </head>
  <body> ... </body>
</html>
```

3) Tag names must be in lower case

Incorrect:

```
<BODY><P>The Best Page Ever</P></BODY>
```

Correct:

```
<body><p>The Best Page Ever</p></body>
```

4) All XHTML elements must be closed

Incorrect 1:

```
<p>This is a paragraph.<p>This is another paragraph.
```

Correct 1:

```
<p>This is a paragraph.</p><p>This is another paragraph.</p>
```

Incorrect 2:

```
This is a break<br>
Here comes a horizontal rule:<hr>
Here's an image 
```

Correct 2:

```
This is a break<br />
Here comes a horizontal rule:<hr />
Here's an image 
```

5) Attribute names must be in lower case

Incorrect:

```
<table WIDTH="100%">
```

Correct:

```
<table width="100%">
```

6) Attribute values must be quoted

Incorrect:

```
<td rowspan=3>
```

Correct:

```
<td rowspan="3">
```

7) Attribute minimization is forbidden

Incorrect:

```
<textarea readonly>READ-ONLY</textarea>
```

Correct:

```
<textarea readonly="readonly">READ-ONLY</textarea>
```

8) The "id" Attribute replaces the "name" attribute

Incorrect:

```

```

Correct:

```

```

9) Not specifying alternate text for images (using the alt attribute, which helps make pages accessible for devices that do not load images or screen-readers for the blind)

Incorrect:

```

```

Correct:

```

```

10) The XHTML DTD defines mandatory elements

All XHTML documents must have a DOCTYPE declaration. The html, head and body elements must be present, and the title must be present inside the head element.

Minimum XHTML document template:

```
<!DOCTYPE Doctype goes here>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Title goes here</title>
</head>
<body>
Body text goes here
</body>
</html>
```

The DOCTYPE declaration is not a part of the XHTML document itself. It is not an XHTML element, and it should not have a closing tag.

The xmlns attribute inside the <html> tag is required in XHTML. However, the validator on www.w3.org does not complain when this attribute is missing in an XHTML document. This is because "xmlns=http://www.w3.org/1999/xhtml" is a fixed value and will be added to the <html> tag even if you do not include it.

DOCTYPES:

(a) XHTML 1.0 Strict

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

(b) XHTML 1.0 Transitional

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

(c) XHTML 1.0 Frameset

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

(d) XHTML 1.1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

The following is an example of XHTML 1.0 Strict:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>XHTML Example</title>
  </head>
  <body>
    <p>This is tiny example of an XHTML usage.</p>
  </body>
</html>
```

**Developer's Tip:* HTML Tidy is an application that can transform any HTML document into an XHTML one. Amaya is a Web browser/editor that will save HTML documents as XHTML.

XML and XHTML Applications

The following is a brief compilation of applications using or based on XML and XHTML.

XML and SOAP

The Simple Object Access Protocol (SOAP) is a way for carrying out Remote Procedure Calls (RPCs) between applications, independent from language and system specifications. The client's request comes in the form of an XML message, which is then sent to the server using HTTP (HyperText Transfer Protocol). The server's reply is also in XML format. SOAP with XML allows applications on different platforms to communicate without hassles.

XML/XHTML as mobile markup language

Mobile phones have difficulty adapting to the current Web markup language since HTML is designed for large screens and powerful machines that can handle a decent amount of graphic rendering. Because of this, an alternative markup language must be designed to be able to cope with mobile phones' resource constraints, such as processing capabilities, screen size, and rendering capabilities. There are three major candidates: (1) *WML*, a widely used XML-based markup language, (2) *cHTML*, a stripped-down version of the current HTML 4.01 specifications, and (3) *XHTML Basic and XHTML Mobile Profile*, which aims to serve content for both the desktop computer and the mobile cellphone, without defining how it would appear on any of the said platforms.

VoiceXML

In 1999, four industry leaders – AT&T, IBM, Lucent, and Motorola – created the VoiceXML Forum



to support the development of VoiceXML. VoiceXML (VXML) is an extension to XML that defines voice segments and enables access to the Internet via telephones and other voice-activated devices.

VXML is the W3C's standard XML format for specifying interactive voice dialogues between a human and a computer. It is analogous to HTML in the sense that just as a visual web browser interprets HTML documents, a voice browser interprets VXML documents. With VXML, voice browsers can provide speech synthesis, automatic speech recognition, dialog management, and soundfile playback.

Here is a sample “Hello world!” VXML document (taken from the W3C):

```
<?xml version="1.0"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml">
  <form>
    <block>
      <prompt>
        Hello world!
      </prompt>
    </block>
  </form>
</vxml>
```

XHTML+Voice

Also known as X+V, XHTML+Voice is the combination of XHTML and VoiceXML to provide websites with voice capabilities. X+V permits Web-enabled handheld devices to interact with voice instead of the screen.

SALT

Speech Application Language Tags (SALT) extends existing Web markup languages to enable multimodal and telephony access to the Web. Although SALT itself is not an XHTML/XML document, its tags can be incorporated into these markup languages. One of XML-based markups that can be used with SALT is WML. Though XHTML+Voice and SALT are still in their early stages of development, it is projected that these technologies will be widely adapted by most mobile phones that provide text-to-speech services.

Multimedia Messaging Service as XML Mobile Application

Another XML-based technology/application for mobile is the Multimedia Messaging Service, or MMS. Made to combat SMS' (Simple Messaging Service) lack of multimedia features, MMS technology can send and receive picture messages, polyphonic ringtones, and wallpapers. MMS uses the Synchronized Multimedia Integration Language (SMIL) – a W3C standard – and the Wireless Application Protocol's WML as the languages to present styled text and color images in multipage messages.

SyncML: An XML-based application for synchronizing data between mobile clients

Because of the data storage- and exchange-intensive nature of mobile clients, transacting data and keeping it up-to-date is important. It is thus necessary for clients to have a common way of synchronizing data. This may prove easy for mobile units of the same model or manufacturer, but things get a little more complicated as we try to exchange and synchronize data between

vastly different mobile architectures, such as an ordinary mobile phone and a personal desktop assistant.

One way to solve this is to use a common human-readable protocol that can be easily implemented for almost every architecture. It seems clear that an XML-based application serves as a candidate for such, and SyncML might just be the winner. SyncML is an application that runs over infrared and Bluetooth connections to synchronize data, and uses XML to encode instructions and data. Current companies that support and sponsor SyncML are Ericsson, IBM, Lotus, Motorola, Nokia, Matsushita, Openwave, Psion, and Starfish Software.

XMLHTTP

XMLHTTP is a set of APIs usable by JavaScript, JScript, VBScript and other Web browser scripting languages to transfer XML and other data to and from a Web server using HTTP. Examples of XMLHTTP applications include Google's Gmail service and the Google Suggest dynamic lookup interface.



XHTML Friends Network

XFN™ (XHTML Friends Network) enables webpage authors indicate their relationship(s) to the people in their list of hyperlinks or blogrolls by simply adding a 'rel' attribute to <a href> tags. The following is an example:

```
<a href="http://rainiercastillo.net" rel="friend met">
```

XHTML+SMIL

XHTML+SMIL adds timing and media synchronization support to XHTML pages by integrating the Synchronized Multimedia Integration Language (SMIL) into the XHTML language. Images, video, and sounds can be added to an XHTML page. XHTML+SMIL allows users to quickly create multimedia-rich, interactive presentations, and view them in a web browser without installing a SMIL player.

OpenOffice.org

OpenOffice.org is a popular open source, front office applications suite released by Sun Microsystems. It contains a word processor, spreadsheet application, slide show application, HTML editor, drawing application, and mathematical formula editor. OpenOffice.org boasts of a saved file format based on an open XML DTD. This gives users and developers great flexibility and power in dealing with work produced in OpenOffice.org.

OpenOffice.org Viewer

The OpenOffice.org Viewer is an application developed by UP Diliman Computer Science students



as their thesis. It is a Java-based application capable of viewing files created in OpenOffice.org Writer, OpenOffice.org Calc, and OpenOffice.org Impress. The application's basic premise is made possible by OpenOffice.org's use of an open XML DTD as its saved file format.

References

- 1) Tanenbaum, Andrew. *Computer Networks (4th Edition)*. 2003
- 2) The World Wide Web Consortium. <http://www.w3.org>
- 3) Extensible Markup Language Home Page. <http://www.w3.org/XML>
- 4) HyperText Markup Language Home Page. <http://www.w3.org/MarkUp>
- 5) Extensible Markup Language W3C Working Draft. November 1996.
<http://www.w3.org/TR/WD-xml-961114.html>
- 6) XML 1.0 Press Release. February 1998.
<http://www.w3.org/Press/1998/XML10-REC>
- 7) Answers.com. <http://www.answers.com>
- 8) XHTML Web Development Article. <http://eedev.palominoweb.com/xhtml>
- 9) Jurvis, Jeff. "Get to the Top With Ten Wireless Technologies".
http://www.fawcette.com/wireless/jurvis/default_pf.asp
- 10) VoiceXML Forum. <http://www.voicexml.org>
- 11) SALT Forum. <http://www.saltforum.org>
- 12) XHTML Friends Network. <http://www.gmpg.org/xfn>
- 13) XHTML+SMIL Profile.
<http://www.w3.org/TR/2002/NOTE-XHTMLplusSMIL-20020131>
- 14) OpenOffice.org. <http://www.openoffice.org>