Transformation of XML Documents Using Technologies over the Web

Fareed Kaleem Khaiser
Senior Lecturer
School of Information Technology
& Mass Communications
TMC Academy, Singapore

Abstract

This article demonstrates the transformation of eXtensible Markup Language (XML) Documents using Technologies over the Web. It will encapsulate about the technologies such as Document Type Definition (DTD) and XML Schema Definition (XSD) in the declaration of rules in mapping with the XML documents. The discussion will also contain the integration of eXtensible Stylesheet Language Transformations (XSLT) with XSD. On a business note, how businesses leverage these technologies and thrive in their businesses online with the help of these technologies and a review on how corporate training on such technologies helps the business will be discussed.

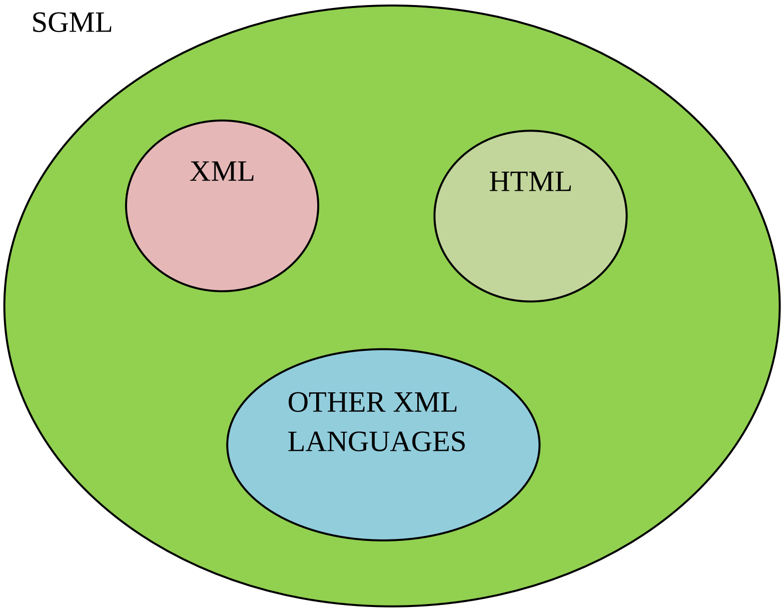
XML – An Introduction

The expansion of XML is e**X**tensible **M**arkup **L**anguage. The Child or the subset of the Standard Generalized Markup Language (SGML) is the XML. The World Wide Web Consortium (W3C) is the consortium responsible body for defining the standards for XML. XML is the universal language for describing and exchanging data on the Web. XML is the industry buzz of the moment, and companies are using it for their business. Companies are using XML as a data-transfer format. XML is a good solution if number of applications required to be moved between enterprises. It is easy to send data across the Internet and through firewalls by using the standard HTTP protocol. XML is non platform specific. XML is a good choice if you want to ensure that your application or data source is robust even if the data schema changes.

SGML – An Introduction

A markup language embeds information about a document within the documents. The Standard Generalized Markup Language (SGML) was designed to reduce the cost and to increase the efficiency of document management. SGML was widely used in high-end publishing systems. SGML's perceived complexity prevented its widespread adoption across the industry. SGML is the parent of both HTML and XML. SGML provides a stable platform for managing data exchange. SGML files can be transformed for publication in a variety of media. The use of SGML preserves textual information independent of how and when it is presented.

Diagrammatic Representation of SGML, XML and HTML:



DIFFERENCE BETWEEN XML AND HTML:

XML	HTML
Information **content**	Information **presentation**
Extendable set of tags	**Fixed** set of tags
Data **exchange** language	Data **presentation** language
Greater hypertext linking	**Limited** hypertext linking

DTD – An Introduction

DTD, a Document Type Definition, has a set of declarative rules in order to map up with the XML. The primary aim of a DTD (Document Type Definition) is to define the legal building blocks of an XML document. A DTD defines the document structure with a list of legal elements and attributes.

Well-formed Document:

A Well-formed XML Document requires following several generic rules, such as:

- The XML Document may only have one root element (<book>...</book>).
- All tags have an ending tag or are themselves self-ending (<author>...</author> or <author/>).

- Tags never overlap, so this arrangement would produce an error: <author><name>...</author></name>.

It is important to make an XML Document well-formed. In fact, W3C doesn't consider an XML Document to be XML unless it is well-formed. If the XML Document contains a reference to a schema (e.g., a DTD, Relax NG Schema, Schematron Schema or an XML Schema), it's better to validate the XML Document.

Valid Document:

A valid XML Document, in addition to being well-formed, conforms to all the constraints imposed by a schema (e.g., a DTD, Relax NG Schema, Schematron Schema or an XML Schema). In other words, it does not contain any tags that are not permitted by the schema, and the order of the tags conforms to the schema's specifications.

There are three categories by which the DTD can be divided as:

(1.) Internal DTD with XML <= 1 file with standalone="yes" in the DTD code
(2.) External DTD with XML <= 2 files with standalone="no" in the DTD code
(3.) Hybrid DTD with XML <= 2 files with standalone="no" in the DTD code. The code in one XML file is a bit complex with the internal and external DTD code. The other DTD file contains the mapping up of an XML document.

(1.) Internal DTD with XML (1 File : **InternalDTD_bib.xml)** Example

```
<?xml version="1.0" standalone="yes"?>

<!--Internal DTD and XML as one file InternalDTD_bib.xml -->

<!DOCTYPE bibliography [
        <!ELEMENT bibliography (header, bibItem*)>
        <!ELEMENT header (#PCDATA)> <!-- parsed character data -->
        <!ELEMENT bibItem (#PCDATA)>
]>

<bibliography>
        <header>References</header>
        <bibItem>Technology books</bibItem>
</bibliography>
```

Rules of DTD:

!DOCTYPE bibliography defines that the root element of this document is bibliography

! ELEMENT header defines the header element to be of type "#PCDATA"

! ELEMENT bibItem defines the bibItem element to be of type "#PCDATA”. The * means that bibItem can have zero or more items.

(2.) External DTD with XML (2 Files : 1.**ExternalXML_bib.xml, 2.bib.dtd)** Example

1.ExternalXML_bib.xml

```
<?xml version="1.0" standalone="no"?>

<!--External XML as one file ExternalXML_bib.xml -->

<!DOCTYPE bibliography SYSTEM "bib.dtd">

<bibliography>
        <header>References</header>
        <bibItem>Technology books</bibItem>
</bibliography>
```

SYSTEM “bib.dtd” refers to the file bib.dtd in the process to verify and validate the mapping of the DTD with the XML contents.

2. bib.dtd

```
<!ELEMENT bibliography (header, bibItem*)>

<!ELEMENT header (#PCDATA)> <!-- parsed character data -->
<!ELEMENT bibItem (#PCDATA)>
```

(3.) Hybrid DTD with XML (2 Files : 1.**hybrid.xml, 2.bib.dtd)** Example

1.hybrid.xml

```
<?xml version = "1.0" standalone = "no"?>

<!DOCTYPE bibliography SYSTEM "bib.dtd" [
<!ELEMENT article (title, (author)*, journal, date)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT journal (#PCDATA)>
<!ELEMENT date (#PCDATA)>
 ]>

<bibliography>
        <header> References </header>
        <bibItem>Jeremy Rifkin, "The Empathic Civilization", 1st Edition, pp.20</bibItem>
                <!-- various elements & content -->
<article>
<title>Learning XML</title>
<author>Eric Ray</author>
<journal>Scholarly Journals in the New Digital World</journal>

<date>26th November 2012</date>

</article>
</bibliography>
```

2.bib.dtd

```
<!ELEMENT bibliography (header, bibItem*,article)>

<!ELEMENT header (#PCDATA)>
<!ELEMENT bibItem (#PCDATA)>

<!--  more declarations here  -->
```

Use of DTD:

(1.) In the usage of DTD, each of your XML files can carry a description of its own format.

(2.) Independent groups of people can agree to use a standard DTD for interchanging data.

(3.) Your application can use a standard DTD to verify that the data you receive from the outside world is valid.

(4.) You can also use a DTD to verify your own data.

XML SCHEMA DEFINITION (XSD) – An Introduction

XML Schema is a successor of DTD technology and is an XML-based alternative to DTD. It describes the structure of an XML document. It is also referred to as XML Schema Definition (XSD). A user of an XML Schema should have to have a basic understanding of HTML/XHTML, XML and a basic understanding of DTD. XML Schema became a W3C Recommendation on 2nd May 2001.

Purpose of an XML Schema

The purpose of an XML Schema is to define the legal building blocks of an XML document, just like a DTD.

An XML Schema can:

- ✓ define the elements that can appear in a document
- ✓ define the attributes that can appear in a document
- ✓ define the which elements are child elements
- ✓ define the order of child elements
- ✓ define the number of child elements
- ✓ define whether an element is empty or can include text
- ✓ define the data types for elements and attributes
- ✓ define the default and fixed values for elements and attributes

XML Schemas are the Successors of DTDs

With the advent of XML Schema, most Web applications will use it as a replacement for DTDs. In contrast with DTD, there are few reasons why XML Schemas are better as:

- ✓ XML Schemas are extensible to future additions
- ✓ XML Schemas are richer and more powerful than DTDs
- ✓ XML Schemas are written in XML
- ✓ XML Schemas support data types
- ✓ XML Schemas support namespaces

XSD EXAMPLE:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="unqualified">
<xsd:element name="tourGuide">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="city" type="cityDetails" minOccurs="1" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:complexType name="cityDetails">
<xsd:sequence>
<xsd:element name="cityName" type="xsd:string"/>
<xsd:element name="adminUnit" type="xsd:string"/>
<xsd:element name="country" type="countryDetails" minOccurs="1" maxOccurs="1"/>
<xsd:element name="population" type="xsd:integer" default="0"/>
<xsd:element name="area" type="xsd:integer"/>
<xsd:element name="elevation" type="xsd:integer"/>
<xsd:element name="longitude" type="xsd:decimal"/>
<xsd:element name="latitude" type="xsd:decimal"/>
<xsd:element name="description" type="xsd:string"/>
<xsd:element name="history" type="xsd:string"/>
<xsd:element name="hotel" type="hotelDetails" minOccurs="0" maxOccurs="unbounded" />
</xsd:sequence>
</xsd:complexType>
<xsd:complexType name="countryDetails">
<xsd:sequence>
<xsd:element name="countryName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
<xsd:element name="population" type="xsd:integer" minOccurs="0" maxOccurs="1"
default="0"/>
<xsd:element name="continent" minOccurs="0" maxOccurs="1">
```

```
<xsd:simpleType>
<xsd:restriction base="xsd:string">
<xsd:enumeration value="Asia"/>
<xsd:enumeration value="Africa"/>
<xsd:enumeration value="Australasia"/>
<xsd:enumeration value="Europe"/>
<xsd:enumeration value="North America"/>
<xsd:enumeration value="South America"/>
<xsd:enumeration value="Antarctica"/>
</xsd:restriction>
</xsd:simpleType>
</xsd:element>

<xsd:element name="topDestination" type="destinationDetails" minOccurs="0" maxOccurs="1"/>
<xsd:element name="destination" type="destinationDetails" minOccurs="0"
maxOccurs="unbounded"/>
</xsd:sequence>
</xsd:complexType>

<xsd:complexType name="destinationDetails">
<xsd:all>
<xsd:element name="destinationName" type="xsd:string"/>
<xsd:element name="description" type="xsd:string"/>
<xsd:element name="streetAddress" type="xsd:string" minOccurs="0"/>
<xsd:element name="telephoneNumber" type="xsd:string"   minOccurs="0"/>
<xsd:element name="websiteURL" type="xsd:anyURI" minOccurs="0"/>
</xsd:all>
</xsd:complexType>
</xsd:schema>
```

XML EXAMPLE:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="country.xsl" media="screen"?>
<tourGuide xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
 xsi:noNamespaceSchemaLocation='file:/C:/Users/HOMPAN/Documents/xml/country.xsd'>

<city>
<cityName>Kuala Lumpur</cityName>
<adminUnit>Selangor</adminUnit>
<country>
<countryName>Malaysia</countryName>
<population>22229040</population>
<continent>Asia</continent>
<topDestination>
<description>A popular duty-free island north of Penang.</description>
<destinationName>Pulau Langkawi</destinationName>
</topDestination>
<destination>
<destinationName>Muzium Di-Raja</destinationName>
<description>The original palace of the Sultan</description>
</destination>
<destination>
<destinationName>Kinabalu National Park</destinationName>
<description>A national park</description>
</destination>
</country>
```

```
<population>1448600</population>
<area>243</area>
<elevation>111</elevation>
<longitude>101.71</longitude>
<latitude>3.16</latitude>
<description>Kuala Lumpur is the capital of Malaysia and is the largest city in the nation.
</description>
<history>The city was founded in 1857 by Chinese tin miners and superseded Klang. In 1880 the
British government transferred their headquarters from Klang to Kuala Lumpur, and in 1896 it
became the capital of Malaysia.</history>
</city>
<city>
<cityName>Belmopan</cityName>
<adminUnit>Cayo</adminUnit>
<country>
<countryName>Belize</countryName>
<population>249183</population>
<continent>South America</continent>
<topDestination>
<destinationName>San Pedro</destinationName>
<description>San Pedro is an island off the coast of Belize</description>
</topDestination>
<destination>
<destinationName>Belize City</destinationName>
<description>Belize City is the former capital of Belize</description>
</destination>
<destination>
<destinationName>Xunantunich</destinationName>
<description>Mayan ruins</description>
</destination>
</country>
<population>11100</population>
<area>5</area>
<elevation>130</elevation>
<longitude>12.3</longitude>
<latitude>123.4</latitude>
<description>Belmopan is the capital of Belize</description>
<history>Belmopan was established following devastation of the former capitol, Belize City, by
Hurricane Hattie in 1965. High ground and open space influenced the choice and ground-breaking
began in 1966. By 1970 most government offices and operations had already moved to the new
location.</history>
<hotel>
<hotelPicture filename="bull_frog_inn.jpg" size="80" value="Image of Bull Frog Inn"
imageURL="http://www.bullfroginn.com"/>
<hotelName>Bull Frog Inn</hotelName>
<streetAddress>25 Half Moon Avenue</streetAddress>
<telephoneNumber>501-822-3425</telephoneNumber>
<emailAddress>bullfrog@btl.net</emailAddress>
<websiteURL>http://www.bullfroginn.com/</websiteURL>
<hotelRating>4</hotelRating>
</hotel>
</city>
</tourGuide>
```

XML STYLESHEET LANGUAGE (XSL) – An Introduction

XSL stands for eXtensible Stylesheet Language, and is a style sheet language for XML documents. XSLT stands for XSL Transformations, with which XML documents can be transformed into XHTML documents or to other XML documents so that it can appear on the webpage using any browser. With XSLT you can add/remove elements and attributes to or from the output file. You can also rearrange and sort elements, perform tests and make decisions about which elements to hide and display, and a lot other tasks. XSLT became a W3C Recommendation on 16th November 1999.

A common way to describe the transformation process is that XSLT transforms an XML source-tree into an XML result-tree.

XSL uses XPath which is a language for navigating elements and attributes in XML documents.

In the transformation process, XSLT uses XPath to define parts of the source document that should match one or more predefined templates. When a match is found, XSLT will transform the matching part of the source document into the result document.

XSL EXAMPLE:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html"/>
<xsl:template match="/">
<HTML>
<HEAD>
<TITLE>Tour Guide</TITLE>
<STYLE TYPE="text/css">
H2 {TEXT-ALIGN:CENTER;}
.greenBackground {BACKGROUND-COLOR:LIGHTGREEN; TEXT-ALIGN:CENTER;}
.yellowBackground {BACKGROUND-COLOR:YELLOW; TEXT-ALIGN:CENTER; FONT-WEIGHT:BOLD; FONT-SIZE:14pt;}
.salmonBackground {BACKGROUND-COLOR:LIGHTSALMON; TEXT-ALIGN:CENTER; FONT-SIZE:12pt;}
</STYLE>
</HEAD>
<BODY>
<H2>Top Tourist Destinations</H2>
<xsl:apply-templates select="tourGuide"/>
</BODY>
</HTML>
```

```
</xsl:template>
<xsl:template match="tourGuide">
<TABLE BORDER="1" WIDTH="100%">
<xsl:for-each select="city/country">
<xsl:sort select="countryName"/>
<xsl:if test="population &gt; 10000">
<TR>
<TD CLASS="greenBackground"><BR/>
<xsl:text>Country: </xsl:text><xsl:value-of select="countryName"/></TD><BR/>
<TD CLASS="greenBackground"><BR/>
<xsl:text>Population: </xsl:text><xsl:value-of select="population"/>
</TD>
<xsl:for-each select="topDestination">
<TD CLASS="yellowBackground"><BR/>
<xsl:text>Top Destination: </xsl:text><xsl:value-of select="destinationName"/>
</TD>
</xsl:for-each>
<xsl:for-each select="destination">
<TD CLASS="yellowBackground"><BR/>
<xsl:text>Destination: </xsl:text><xsl:value-of select="destinationName"/>
</TD>
</xsl:for-each>
</TR>
</xsl:if>
</xsl:for-each>
</TABLE>
</xsl:template>
</xsl:stylesheet>
```

Once the schema is defined and XML is written mapping the contents of the schema, it becomes quintessential to transform these documents into XSL with which it can be represented in a web browser.

This article is not a tutorial and is not intended for novice users and it is necessary that in order to know more about DTD, XML, XSD and XSLT, one has to refer to the language reference to understand the syntax and the usage of the keywords of the specific language. This article, however, provides examples to be used using any online XML editors to check for a well formed and a valid document.

XML Documents Transformation – A Business Perspective

Documents when transformed and shared over the web, a specific standard such as XML standard with reference to the W3C recommendations requires the documents to securely reach their destinations over a network connection. In large organizations, the data shared or transformed requires involves huge space and essential data sharing constraints for the users of the system and any wrong data entry with the help of an application interface for adding up the XML data required to be checked as it may involve financial as well as confidential data. It can lead to manpower and time constraints to fulfil a request from third parties and customers with huge amount of information to be entered into the system. Even with these constraints, businesses required to thrive and to remain in the business without the need for outsourcing as the cost of it would not be minimal. Hence, the need to have corporate training for the employees in XML technologies, by which the organizations, can utilize the XML technologies effectively and efficiently.

Corporate Training – A review

As a corporate trainer and the author of this article, with the experience to having taught XML technologies to one of a reputed organization in Singapore and it is a well informed choice to train the employees in these XML technologies so that it may involve less time in the usage of such technologies thereby saving time and money. It also gives the employees with the first-hand information and also to clarify their way of working in a real-time and how best they can perform their work. For managers, they are more concerned with how the data can be entered effectively with the help of customized interfaces available in the organization, which at times has a disadvantage of error correction. Hence, it was suggested to have a grid like structure available in exchanger XML software so that error can be easily checked and corrected.

In conclusion, it is the need of an organization by which a technology can help with a foresight of short-term or long-term goals. Without outsourcing, organizations can still thrive and develop their businesses by retaining their employees by providing them training which will be long-term goal and also in the retention of the employees in the long run.

References

Refsnes, Jan Egil (n.d.) (1999-2013). Introduction to DTD. [ONLINE] Available at: http://www.w3schools.com/dtd/dtd_intro.asp. [Accessed 21 November 2012].

Refsnes, Jan Egil (n.d.) (1999-2013). *XML Validation*. [ONLINE] Available at: http://www.w3schools.com/xml/xml_dtd.asp. [Accessed 21 November 2012].

Refsnes, Jan Egil (n.d.) (1999-2013). *Introduction to XML Schemas*. [ONLINE] Available at: http://www.w3schools.com/schema/schema_intro.asp. [Accessed 21 November 2012].

Refsnes, Jan Egil (n.d.) (1999-2013). *XML Schema Tutorials*. [ONLINE] Available at: http://www.w3schools.com/schema/schema_intro.asp. [Accessed 21 November 2012].

W3Schools (1999-2013). *XSLT Introduction*. [ONLINE] Available at: http://www.w3schools.com/xsl/xsl_intro.asp. [Accessed 21 November 2012].

*Warren Anacoura (*2013). *Module Description*. [ONLINE] Available at: http://www.ljmu.ac.uk/cmp/95853.asp?searchcode=CMPCD3044. [Accessed 21 November 2012].

XMLBlueprint (2002 - 2013). *Well-formed and valid XML Documents*. [ONLINE] Available at: http://www.xmlblueprint.com/help/html/topic_118.htm. [Accessed 21 November 2012].

Liam R. E. Quin (2010). *Transformation - W3C*. [ONLINE] Available at: http://www.w3.org/standards/xml/transformation. [Accessed 21 November 2012].

MSDN - Microsoft (2013). *Walkthrough: Displaying an XML Document in a Web Forms Page using Transformations*. [ONLINE] Available at: http://msdn.microsoft.com/en-us/library/18az815d%28v=vs.71%29.aspx. [Accessed 21 November 2012].

Refsnes, Jan Egil (n.d.) (1999-2013). *XSLT Transformation*. [ONLINE] Available at: http://www.w3schools.com/xsl/xsl_transformation.asp. [Accessed 21 November 2012].

ShahabFatemi (2005). *How to Convert XML Files to HTML - CodeProject*. [ONLINE] Available at: http://www.codeproject.com/Articles/12047/How-to-Convert-XML-Files-to-HTML. [Accessed 21 November 2012].

Vladimir Geroimenko (2004). Terms of XML Technologies and the Semantic Web. UK: Springer London. Pp1-219.