

## Web Publishing Frameworks – the XML Automation

Lect. Razvan Daniel ZOTA PhD.  
Academy of Economic Studies Bucharest, Romania

**Abstract:** *Internet technologies like Java and XML are developing continuously. Over the next few years, almost every major application will be either web-based, or at a minimum will have a web front-end. One of the hottest topics of the recent years is the XML application that seems to have generated the largest amount of excitement in the XML and Java communities: the web publishing framework. Even if the problem of generating presentation from content is probably over-hyped when compared to the value of the portable data that XML provides, using XML for presentation styling is still important, and this importance increases when looking at web-based applications.*

*One of the major applications of XML is about displaying an XML document using an XSL stylesheet. We will cover in this article some developments in the field using representative examples for a better understanding.*

**Keywords:** *XML, SGML, HTML, XSL, Java servlet, Web publishing, WML.*

### XML Introduction

Today a buzzword in the software industry, the eXtensible Markup Language (XML) is designed to complement, not replace, HTML. XML is a way to describe structured data, but unlike an HTML file, an XML file does not hold any information about *displaying* data. It simply contains the data in a highly organized and descriptive form.

The ancestor for both HTML and XML is the Standard Generalized Markup Language (SGML). This extended language is used to create complex systems (or applications) of data and includes rules on how to describe and display data. HTML is an application of SGML; on the other hand, XML is a subset of SGML designed to provide a simple, web-optimized way to represent data that is easier to work with than full-blown SGML.

XML conception is web-oriented and brings some things not found in SGML. Browsers can use XML combined with HTML for data display and presentation.

In order to support the XML developing over the Web, another two specifications have been created:

The Extensible Linking Language (XLL) specification provides a hyperlinking scheme that not only supports basic HTML

hyperlinking but also extended linking, which adds functionality such as letting a single link point to multiple resources.

The eXtensible Stylesheet Language (XSL) specification gives support for an XML style sheet mechanism, a feature also not found in straight SGML. Stylesheet permit authors to create templates of styles or combination of styles, and apply them to elements in an XML document when it is displayed. In conclusion, "XML provides 80 percent of the features and functionality of SGML with 20 percent of its complexity" [1].

An important feature of XML is the one that it can act as a centralized data-interchanging format, which can be accessed by a variety of computer systems. The interchange of information between different systems can be easily made using XML as an intermediary format. This enables data exchange among systems that usually are incompatible.

We will present a basic example of an XML document in comparison with an HTML document. Let us consider a plain text document which represents an entry in an addressbook: we have first name, last name, the name of the street, city, zip code, phone number and email address.

It is much simpler to view such a plain text document (see listing 1) instead of an HTML document (listing 2), which represents the equivalent information. Viewed through a browser, the HTML document would look great, but let's look what it has done to the information itself: the data has been obscured by the formatting tags. So, if you want to process the information, the plain version of the document looks more attractive. But can we tell whether Ilie is the first name or the last? Also, can we tell whether the number at the end is a regular phone no, a cell phone number, a fax number or something else? This brings us to the concept of *information about information* – data that

describes a piece of information and gives meaning to it. And if this meta-information can be given in such a way that a computer can process it, even better. This leads us to the basic understanding of the XML – it is about describing the data.

XML does not represent a language in itself. You can look at it as a set of specifications based on which other languages can be defined – in fact, you have the liberty of defining your own language based on XML rules. You can create your own tags to provide a better understanding of the information. We can represent the information we saw above using an XML syntax as in listing 3.

```
Adrian Ilie
Luterana Bucharest 7001
210.00.01
ilie@yahoo.com
```

**Listing 1**

```
<HTML><BODY><H1><U><B>Adrian
Ilie</B></U></H1><BR>
<I>Luterana Bucharest 7001</I><BR>
<H5>210.00.01</H5><BR>
<B><U>ilie@yahoo.com</U></B><P></BODY></HTML>
```

**Listing 2**

```
<?xml version="1.0" ?>
<AddressBook>
  <Person>
    <Name>
      <First>Adrian</First>
      <Last>Ilie</Last>
    </Name>
    <Address>
      <Street>Luterana</Street>
      <City>Bucharest</City>
      <ZipCode>7001</ZipCode>
    </Address>
    <Phone>210.00.01</Phone>
    <Email>ilie@yahoo.com</Email>
  </Person>
</AddressBook>
```

**Listing 3**

### XML Application for Web Publishing

The Web is nothing more than another media comparable to print, radio or TV. It has unique characteristics, but its reach is the same to other popular media.

As proof, there are some initiatives by existing publishers to offer their content online (visit [www.informix.com](http://www.informix.com) or [www.earthweb.com](http://www.earthweb.com)) or, furthermore, a growing number of companies (not necessarily publishers) use their web sites to distribute information, article, reports, etc.

In spite of these developments, this young media is continuously changing; there is a so called "web fashion" which changes every six months. According to W3C (World Wide Web Consortium), non-desktop browsers might reach 75% of all surfers by 2002. These include mobile phones, PDAs and WebTV. And most of these devices won't use HTML. For example, mobile phones use a special language, **Wireless Markup Language (WML)**, incompatible with HTML.

So, what is the solution? Should content providers limit themselves to either HTML or WAP? Should they be prepared to support even more formats? Developing original content (articles, books, reports and so on) is expensive. To cut off the costs, content owners want to distribute their content as widely as possible. Ideally, it should not matter whether the reader uses a PC, a mobile phone, a PDA or some other device.

In this context, XML can help address this challenge. The XML solution is to separate authoring from publishing. The author of the pages writes the document in XML, ignoring presentation. Instead of this, the author uses an XML vocabulary that focuses on the organization of the document: title, sections, paragraphs, references, etc.

Publishing the document then simply requires converting the document into HTML, WML or another popular format. And this thing can be done because the original XML document is structure rich.

The operative word is automated. For medium and large sites, it is more cost effective to automate publishing. To apply these principles we will focus on three main elements:

- documents in XML;
- XSLT stylesheets which implement the conversion (XML to HTML or another well-known format)
- a servlet that is responsible for applying the stylesheet.

### XSLT

XSL – the XML Stylesheet Language helps us to publish XML documents. More specifically, we will use XSLT (XSL Transformations). XSLT is a scripting language optimized for conversion between XML documents. In that respect it differs from early stylesheet languages, such as CSS (Cascading Style Sheets) or word processor stylesheets.

XSLT transforms the XML document into another XML document. It goes much further than simple presentation instructions. In fact, XSLT can completely reorganize a document and, for example, add a table of content or delete a section.

How does it work? We need to transform an XML document into a format that contains display information, such as HTML or WML. A browser or another viewer can render the transformed document on screen or on paper. We will focus to XML-HTML transformations. Let us consider the XML document with the same structure like in listing 3. We added some other information in the document; see listing 4. How do we convert it to HTML?

The last step involves the servlet that takes the XML document from listing 4 and publishes it in HTML.

I have considered the example from Benoît Marchal [2]; a servlet made in Java. Applying the stylesheet is the responsibility of a method in the java program, which uses Xalan 1.0, the Apache XSLT processor.

To run the servlet, in Windows you need to install JDK 1.1.8 or higher, adding the java\bin directory in your PATH variable and setting CLASSPATH to C:\<JavaDir>\classes; c:\<JavaDir>\lib\classes.zip, where <JavaDir> is the directory where Java is installed. After running the servlet, call in

a browser <http://localhost:8080/publish> and you will receive the XML transformed to HTML.

Listing 5 represent the XSL document used to transform the XML document, and listing 6 is the HTML document automatic obtained through the transformation.

```
<?xml version="1.0" ?>
<AddressBook>
  <Person>
    <Name>
      <First>Adrian</First>
      <Last>Ilie</Last>
    </Name>
    <Address>
      <Street>Luterana</Street>
      <City>Bucharest</City>
      <ZipCode>7001</ZipCode>
    </Address>
    <Phone>210.00.01</Phone>
    <Email>ilie@yahoo.com</Email>
  </Person>
  <Person>
    <Name>
      <First>Dan</First>
      <Last>Macovei</Last>
    </Name>
    <Address>
      <Street>Iancului</Street>
      <City>Bucharest</City>
      <ZipCode>7002</ZipCode>
    </Address>
    <Phone>210.00.02</Phone>
    <Email>macovei@yahoo.com</Email>
  </Person>
  <Person>
    <Name>
      <First>Diana</First>
      <Last>Lupescu</Last>
    </Name>
    <Address>
      <Street>Florilor</Street>
      <City>Cluj-Napoca</City>
      <ZipCode>7002</ZipCode>
    </Address>
    <Phone>210.00.02</Phone>
    <Email>diana@yahoo.com</Email>
  </Person>
```

```

<Person>
  <Name>
    <First>Adriana</First>
    <Last>Cacoveanu</Last>
  </Name>
  <Address>
    <Street>Paris</Street>
    <City>Bucharest</City>
    <ZipCode>7003</ZipCode>
  </Address>
  <Phone>210.00.03</Phone>
  <Email>adi@yahoo.com</Email>
</Person>
<Person>
  <Name>
    <First>Paul</First>
    <Last>Georgescu</Last>
  </Name>
  <Address>
    <Street>Lunga</Street>
    <City>Brasov</City>
    <ZipCode>7004</ZipCode>
  </Address>
  <Phone>210.00.04</Phone>
  <Email>paulg@yahoo.com</Email>
</Person>
<Person>
  <Name>
    <First>Manuela</First>
    <Last>Berindei</Last>
  </Name>
  <Address>
    <Street>Dacia</Street>
    <City>Bucharest</City>
    <ZipCode>7005</ZipCode>
  </Address>
  <Phone>210.00.05</Phone>
  <Email>manuela@yahoo.com</Email>
</Person>
</AddressBook>

```

**Listing 4**

```

<?xml version="1.0"?>

<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/TR/REC-html40"
  version="1.0">

<xsl:output method="html"/>

```

```

<xsl:template match="/AddressBook">
<HTML><HEAD><TITLE>XML / XSL Publishing</TITLE></HEAD>
<BODY><H1>All persons in the addressbook:</H1>
  <TABLE BORDER="3">

<THEAD><TD><B>FirstName</B></TD><TD><B>LastName</B></TD><TD><B>
>Street</B></TD><TD><B>City</B></TD><TD><B>Phone</B></TD><TD><B>
Email</B></TD></THEAD>

  <xsl:for-each select="Person">
    <TR>
      <TD><xsl:value-of select="Name/First"/></TD>
      <TD><xsl:value-of select="Name/Last"/></TD>
      <TD><xsl:value-of select="Address/Street"/></TD>
      <TD><xsl:value-of select="Address/City"/></TD>
      <TD><xsl:value-of select="Phone"/></TD>
      <TD><xsl:value-of select="Email"/></TD>
    </TR>
  </xsl:for-each>
</TABLE>

</BODY></HTML>
</xsl:template>

</xsl:stylesheet>

```

### Listing 5

```

<HTML xmlns="http://www.w3.org/TR/REC-html40">
<HEAD>
  <TITLE>XML / XSL Publishing</TITLE>
</HEAD>
<BODY>
  <H1>All persons in the addressbook:</H1>
  <TABLE BORDER="3">
    <THEAD>

<TD><B>FirstName</B></TD><TD><B>LastName</B></TD><TD><B>Street</B></
TD><TD><B>City</B></TD><TD><B>Phone</B></TD><TD><B>Email</B></TD>
    </THEAD>
    <TR>

<TD>Adrian</TD><TD>Ilie</TD><TD>Luterana</TD><TD>Bucharest</TD><TD>21
0.00.01</TD><TD>ilie@yahoo.com</TD>
    </TR>
    <TR>

<TD>Dan</TD><TD>Macovei</TD><TD>Iancului</TD><TD>Bucharest</TD><TD>2
10.00.02</TD><TD>macovei@yahoo.com</TD>
    </TR>

```

```

<TR>
  <TD>Diana</TD><TD>Lupescu</TD><TD>Florilor</TD><TD>Cluj-
Napoca</TD><TD>210.00.02</TD><TD>diana@yahoo.com</TD>
</TR>
<TR>
  <TD>Adriana</TD><TD>Cacoveanu</TD><TD>Paris</TD><TD>Bucharest</TD><TD>
210.00.03</TD><TD>adi@yahoo.com</TD>
</TR>
<TR>
  <TD>Paul</TD><TD>Georgescu</TD><TD>Lunga</TD><TD>Brasov</TD><TD>210
.00.04</TD><TD>paulg@yahoo.com</TD>
</TR>
<TR>
  <TD>Manuela</TD><TD>Berindei</TD><TD>Dacia</TD><TD>Bucharest</TD><TD>
210.00.05</TD><TD>manuela@yahoo.com</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

### Listing 6

#### Conclusions

Data interchange is an area in which XML can and will play a very prominent role. As opposed to a database, in which data may be organized and stored in a proprietary format, an XML file is an ASCII representation of the data, a representation that has the advantage of a good structural organization yet one can be easily transmitted. Considering the world of e-commerce, the exchange of information is done recently through the use of XML data. Fortunately, XML provides this solution to a common way of publishing the information. In the respect of exchanging data, the company involved would write a program that reads from this standard format (XML) and store in its

own database. This action grants them all (companies) complete independence of the others' databases.

#### References

- [1] Dynamic HTML in Action – second ed. – Eric M. Shurman and William J. Pardi, Microsoft Press, 1999
- [2] Applied XML Solutions – the Authoritative Solution – Benoît Marchal, SAMS, 2000
- [3] Developing e-Commerce Sites – An Integrated Approach – Vivek Sharma, Rajiv Sharma, Addison Wesley, 2000
- [4] Java and XML – Brett McLaughlin, O'Reilly, 2000
- [5] <http://www.w3.org/TR/2000/CR-xsl-20001121/> - XSL Reference