More XML

Extensible Markup Language (XML) has developed in a major way in the last few years. It has become one of the most important means of exchanging information of all kinds on the Internet. And there are now a number of different specialized versions of XML. These range from markup for chemistry documents to a quick way to distribute news releases.

Specialized versions are accompanied by document type definitions (DTDs) and/or schema. Both provide some information about the tags used in the field, but schema, being more recent, are more extensive. However, we will begin with a look at DTDs.

Document Type Definitions

A DTD for an xml file is a list of elements (tags) used in the file, together with some information about how they are defined. The document must have a single root node. This is followed by the children of the root and either their children or data type. In a DTD there are only two data types, PCDATA (parsed character data) or CDATA, unparsed character data. Most of the examples will use parsed character data.

A DTD also indicates how many times an element can occur in the file. The default is once. But most files use the same tag names a number of times. The notation used is similar for that used in regular expressions.
- * means zero or more occurrences.
- + means one or more occurrences.
- ? means zero or one occurrence.

A DTD also allows for choice. A vertical bar (|) is used to indicate one element or another.

A DTD follows for a grocery store application.

grocery.dtd

```
<!--  A document type definition for grocery.xml.  -->
<!ELEMENT grocery (heading+, fruit*, vegetables*, bakery*)>

<!--  The elements that have children.  -->
<!ELEMENT heading (name, id, quantity, price)>
<!ELEMENT fruit (name, id, quantity, price)>
<!ELEMENT vegetables (name, id, quantity, price)>
<!ELEMENT bakery (name, id, quantity, price)>

<!--  Definition of the data types.  -->
<!ELEMENT name (#PCDATA)>
<!ELEMENT id (#PCDATA)>
<!ELEMENT quantity (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

From this DTD you can see that the root element is <grocery>. This element has four different kinds of children. There can be zero or one heading. The heading can be used in a table that displays the file. The DTD also indicates that there can be zero or more fruit, vegetables, and bakery elements. But it also mandates that all fruit elements come first, vegetable elements next, and bakery elements last.

A file that satisfies all these requirements follows:

```xml
<?xml version="1.0"?>
<!DOCTYPE  grocery SYSTEM "grocery.dtd">
<!--
      An xml file that shows names, ids, quantities,
      and prices of fruit, vegetables, and bakery items.
-->
<grocery>
      <heading>
            <name>Name</name>
            <id>ID</id>
            <quantity>Quantity</quantity>
            <price>Price</price>
      </heading>
      <fruit>
            <name>apples</name>
            <id>A123</id>
            <quantity>25</quantity>
            <price>1.25</price>
      </fruit>
      <fruit>
            <name>pears</name>
            <id>P234</id>
            <quantity>50</quantity>
            <price>2.55</price>
      </fruit>
      <vegetables>
            <name>beans</name>
            <id>B345</id>
            <quantity>10</quantity>
            <price>.85</price>
      </vegetables>
      <vegetables>
            <name>corn</name>
            <id>C456</id>
            <quantity>60</quantity>
            <price>.50</price>
      </vegetables>
      <bakery>
            <name>bread</name>
            <id>B567</id>
            <quantity>15</quantity>
            <price>2.30</price>
      </bakery>
      <bakery>
            <name>cake</name>
            <id>C678</id>
            <quantity>4</quantity>
            <price>4.25</price>
      </bakery>
</grocery>
```

If an xml file satisfies the requirements of a DTD, it is said to be *valid*. The previous file has been validated using a program made available on the W3Schools website. This website, created and maintained by the Refsnes Data Company of Norway, is a web consulting firm. The tutorials on their site are an easy way to learn more about web development.

Displaying XML

Many browsers will display an xml file as a tree. A portion of the file as displayed by Mozilla's Foxfire browser appears below.

```
- <!--
     An xml file that shows names, ids, quantities,
   and prices of fruit, vegetables, and bakery items.
  -->
- <grocery>
  - <heading>
      <name>Name</name>
      <id>ID</id>
      <quantity>Quantity</quantity>
      <price>Price</price>
    </heading>
  - <fruit>
      <name>apples</name>
      <id>A123</id>
      <quantity>25</quantity>
      <price>1.25</price>
    </fruit>
  - <fruit>
      <name>pears</name>
      <id>P234</id>
      <quantity>50</quantity>
      <price>2.55</price>
    </fruit>
  - <vegetables>
      <name>beans</name>
      <id>B345</id>
      <quantity>10</quantity>
      <price>.85</price>
    </vegetables>
```

The hyphens can be used to collapse branches in the tree. They are replaced by plus signs that can be clicked on to expand the tree again.

A CSS file can also be used to display the xml file in other ways. The following link must be added to the beginning of the xml file.
       <?xml-stylesheet type="text/css" href="grocery.css"?>
Some browsers will use this information to display the file while others will ignore it. Both Foxfire and Netscape version 7.2 understand the style sheet, while Internet Explorer version 6.0 does not.

The following style sheet will display the file in a table.

<u>grocery.css</u>

```
/* Style sheet for address application. */
grocery
{
      font-family: "Times New Roman", serif
      display: table;
      border-style: solid;
      border-width: thin;
      margin-left: 1.0cm;
      margin-top: 1.0cm;
}
heading, fruit, vegetables, bakery
{
      display: table-row;
}
name, id, quantity, price
{
      display: table-cell;
      border-style: solid;
      border-width: thin;
      padding: 0.3cm;
      text-align: center;
}
```

If this style sheet is applied, the display looks like the following in Foxfire.

Name	ID	Quantity	Price
apples	A123	25	1.25
pears	P234	50	2.55
beans	B345	10	.85
corn	C456	60	.50
bread	B567	15	2.30
cake	C678	4	4.25

This style sheet says that the root element, grocery, should be displayed as a table.
      display: table;
The other styles determine the font and table properties such as a solid, thin border and 1 cm margins.

The columns of the table are given by the next elements, heading, fruit, vegetables, and bakery. The style for these is display: table-row. This will display these elements as rows.

Finally the data elements, name, id, quantity, and price, will be displayed in the table cells.

    display: table-cell;

The cell styles must also have instructions as to how the borders should appear.

There are many other applicable styles. W3Schools has an extensive list in their tutorial on CSS.

Attribute Lists in DTDs

XML tags may include attributes. These are name-value pairs such as width = "300". We have seen these in applet and image tags. They can be used in XML and are required in some places. An example might be the following xml file that contains information about students in a course. Each exam grade has a *weight* attribute to indicate how it should factor into the course grade.

```
<?xml version="1.0"?>
<!DOCTYPE roster SYSTEM "roster.dtd">
<?xml-stylesheet type="text/css" href="roster.css"?>

<roster>
    <heading>
        <Name>Name</Name>
        <Midterm>Midterm</Midterm>
        <Final>Final</Final>
    </heading>
    <student>
        <name>
            <first>Alice</first>
            <last>Lee</last>
        </name>
        <midterm weight1 = "50">85</midterm>
        <final weight2 = "50">92</final>
    </student>
    <student>
        <name>
            <first>Barbara</first>
            <last>Smith</last>
        </name>
        <midterm weight1 = "50">78</midterm>
        <final weight2 = "50">84</final>
    </student>
    <student>
        <name>
            <first>Cathy</first>
            <last>Jones</last>
        </name>
        <midterm weight1 = "50">82</midterm>
        <final weight2 = "50">87</final>
    </student>
</roster>
```

Attributes should also be listed in the DTD for the file. They are defined in an attribute list given by an ATTLIST definition.

roster.dtd

```
<!--  A document type definition for roster.xml.  -->
<!ELEMENT roster (heading, student+)>
<!ELEMENT heading (Name, Midterm, Final)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT Midterm (#PCDATA)>
<!ELEMENT Final (#PCDATA)>

<!--  Each student must have midterm and final grades..  -->
<!ELEMENT student (name, midterm, final)>
<!ELEMENT name (first, last)>
<!ELEMENT first (#PCDATA)>
<!ELEMENT last (#PCDATA)>
<!ELEMENT midterm (#PCDATA)>
<!ELEMENT final (#PCDATA)>

<!--  The midterm and final attributes consist of CDATA with a default value of "0".  -->
<!ATTLIST midterm weight1 CDATA "0">
<!ATTLIST final weight2 CDATA "0">
```

While there are only two datatypes for elements, PCDATA and CDATA, there are quite a few for attribute lists. To learn about the others types, see the references.

The style sheet for roster.xml is very similar to the one for grocery.xml.   Applying it displays the data in the following way.

Name	Midterm	Final
Alice Lee	85	92
Barbara Smith	78	84
Cathy Jones	82	87

References

1. Elliotte Rusty Harold, *Processing XML with Java,* chapter 1, Addison Wesley, 2002.
2. Elliotte Rusty Harold and Scott Means, *XML Programming*, chapters 4 and12, O'Reilly & Associates, Inc., 2004.
3. W3Schools Online Web Tutorials, http://www.w3schools.com.