XML For Technical Writers

1/23/02

Mark Volkmann

Partner

Object Computing, Inc.

Talk Description

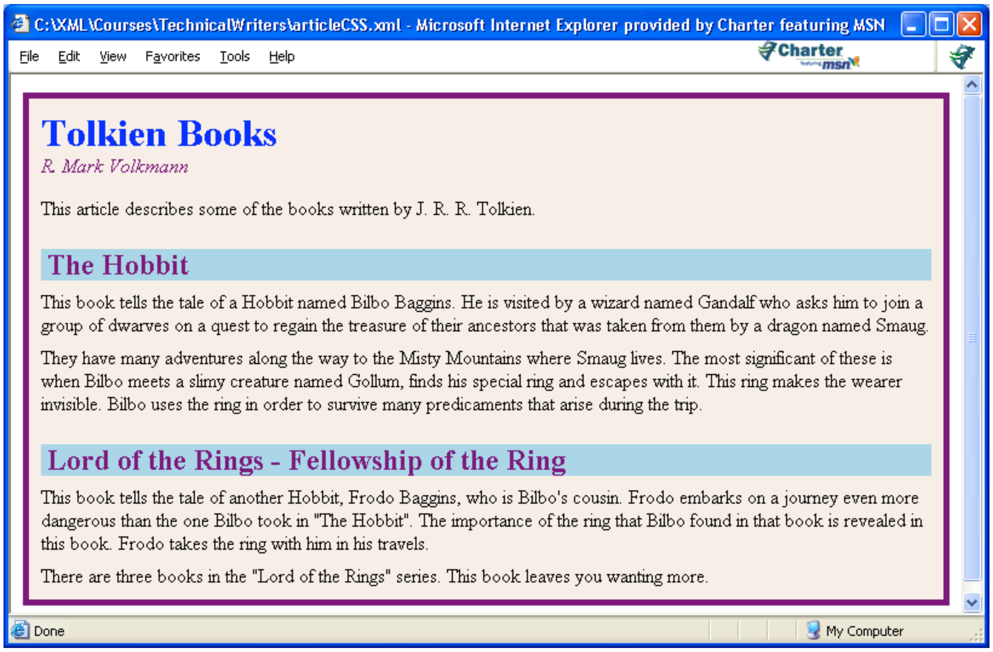
This talk will cover the areas of XML that are most applicable to technical writers.
Topics include XHTML, CSS, XSLT, XSL-FO and the tools Tidy, XML Spy and FOP.

XHTML is a form of HTML that aims to be more portable that HTML across a wide variety of web browsers. **Tidy** is a tool that automates turning HTML into XHTML.

XSLT is a language that can transform XML documents into alternate text formats including HTML. The combination of XML and XSLT is an improvement over HTML because it provides a clean separation between content and presentation, allowing multiple presentations of the same content. **XML Spy** is an XML editor that supports creation and testing of XSLT stylesheets.

XSL-FO is a language for specifying formatting characteristics in a media-independent manner. **FOP** is a tool that transforms XSL-FO documents into PDF and other presentation formats.

The Goal in HTML


C:\XML\Courses\TechnicalWriters\articleCSS.xml - Microsoft Internet Explorer provided by Charter featuring MSN
File Edit View Favorites Tools Help
Charter msn
Done
My Computer

Tolkien Books

R. Mark Volkmann

This article describes some of the books written by J. R. R. Tolkien.

The Hobbit

This book tells the tale of a Hobbit named Bilbo Baggins. He is visited by a wizard named Gandalf who asks him to join a group of dwarves on a quest to regain the treasure of their ancestors that was taken from them by a dragon named Smaug.

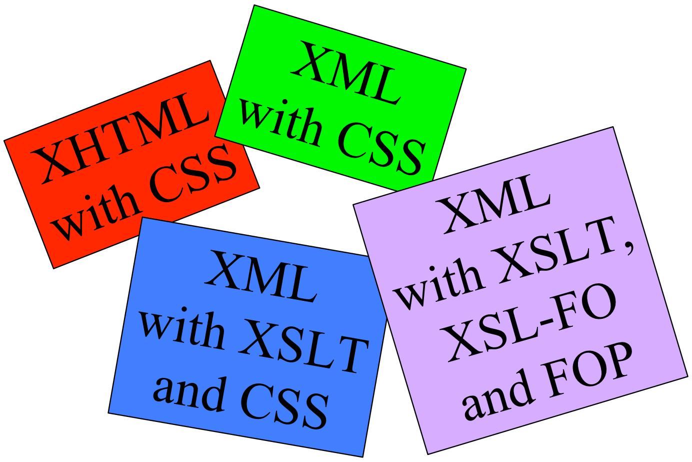
They have many adventures along the way to the Misty Mountains where Smaug lives. The most significant of these is when Bilbo meets a slimy creature named Gollum, finds his special ring and escapes with it. This ring makes the wearer invisible. Bilbo uses the ring in order to survive many predicaments that arise during the trip.

Lord of the Rings - Fellowship of the Ring

This book tells the tale of another Hobbit, Frodo Baggins, who is Bilbo's cousin. Frodo embarks on a journey even more dangerous than the one Bilbo took in "The Hobbit". The importance of the ring that Bilbo found in that book is revealed in this book. Frodo takes the ring with him in his travels.

There are three books in the "Lord of the Rings" series. This book leaves you wanting more.

Presentation Generation Variations We’ll Discuss



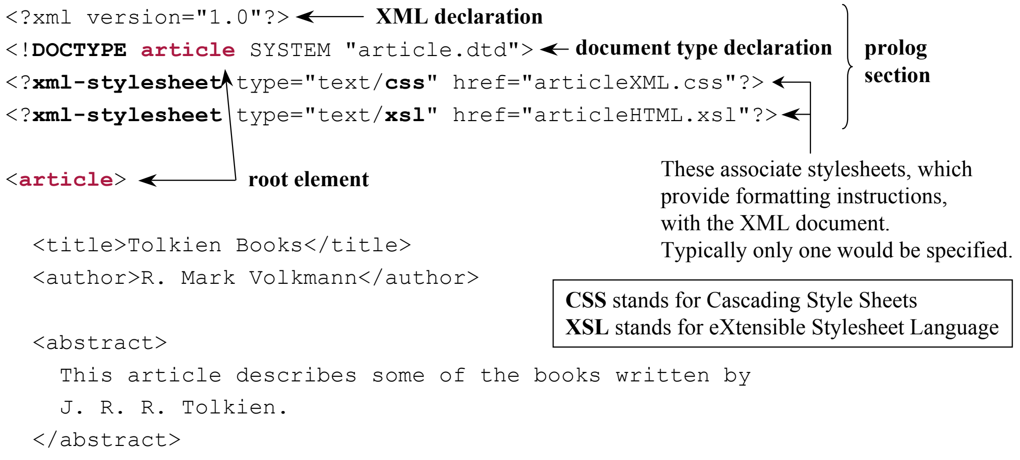
What Is XML?

- Marks up data or content with tags and attributes
  - see example on next page
- Benefits
  - separates data from formatting
  - very portable data format
    - can be created and parsed by most programming languages on most operating systems
  - human readable
  - easily transformed to different XML (using XSLT or DOM)
  - easily validated (using DTDs or XML Schemas)
  - well supported (by tools, programming APIs and some web browsers)
- Not very compact
  - compared to binary and positional text formats
  - however, it compresses well with tools like WinZip and BOX

box.sourceforge.net

Example XML

(see articleCSS.xml and articleXSLT.xml)



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

<article>

  <title>Tolkien Books</title>
  <author>R. Mark Volkmann</author>

  <abstract>
    This article describes some of the books written by
    J. R. R. Tolkien.
  </abstract>
```

These associate stylesheets, which provide formatting instructions, with the XML document. Typically only one would be specified.

CSS stands for Cascading Style Sheets
XSL stands for eXtensible Stylesheet Language

Example XML (Cont’d)

```
  <section>
    <title>The Hobbit</title>
    <paragraph>
      This book tells the tale of a Hobbit named Bilbo Baggins.
      He is visited by a wizard named Gandalf who asks him to
      join a group of dwarves on a quest to regain the treasure
      of their ancestors that was taken from them by a dragon
      named Smaug.
    </paragraph>
    <paragraph>
      They have many adventures along the way to the Misty Mountains
      where Smaug lives. The most significant of these is when Bilbo
      meets a slimy creature named Gollum, finds his special ring
      and escapes with it. This ring makes the wearer invisible.
      Bilbo uses the ring in order to survive many predicaments that
      arise during the trip.
    </paragraph>
  </section>
```

Example XML (Cont’d)

```
  <section>
    <title>Lord of the Rings - Fellowship of the Ring</title>
    <paragraph>
      This book tells the tale of another Hobbit, Frodo Baggins,
      who is Bilbo's cousin. Frodo embarks on a journey even more
      dangerous than the one Bilbo took in "The Hobbit".
      The importance of the ring that Bilbo found in that book
      is revealed in this book.
      Frodo takes the ring with him in his travels.
    </paragraph>
    <paragraph>
      There are three books in the "Lord of the Rings" series.
      This book leaves you wanting more.
    </paragraph>
  </section>

</article>
```

XML Validation

- Validates the structure and content of an XML document
- Primarily specified with
  Document Type Definitions (DTDs) and XML Schemas
  - XML Schema is more powerful, but isn’t as widely supported yet
- Ways of validating XML
  - IE and Netscape do not report validation errors
  - W3C Validator (`validator.w3.org`)
    - web page that validates XML files on a web server
      - doesn’t work with XML files that are only locally accessible
  - XML Spy (`www.xmlspy.com`)
    - a commercial XML/XSL editor
  - custom software using SAX or DOM parsers
    - for example, Xerces from Apache

Document Type Definition (DTD)

- Defines allowed elements and their attributes

  ```
  <author gender="male">
  ```

  - `author` is an element
  - `gender` is an attribute
- Defines the valid element nesting relationships

  ```
  <article>
    <title> . . . </title>
    <author> . . . </author>
    <abstract> . . . </abstract>
  </article>
  ```

- Defines the sequence in which elements must appear
  - `title` must appear before `author` element
- Defines the cardinality of elements
  - number of allowed occurrences

Example DTD

(see article.dtd)

```
<!ELEMENT article (title, author, abstract, section*)>
<!ELEMENT section (title, paragraph*)>

<!ELEMENT abstract (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT paragraph (#PCDATA)>
<!ELEMENT title (#PCDATA)>
```

Cardinality

? means 0 or 1

***** means 0 or more

+ means 1 or more

nothing means exactly one

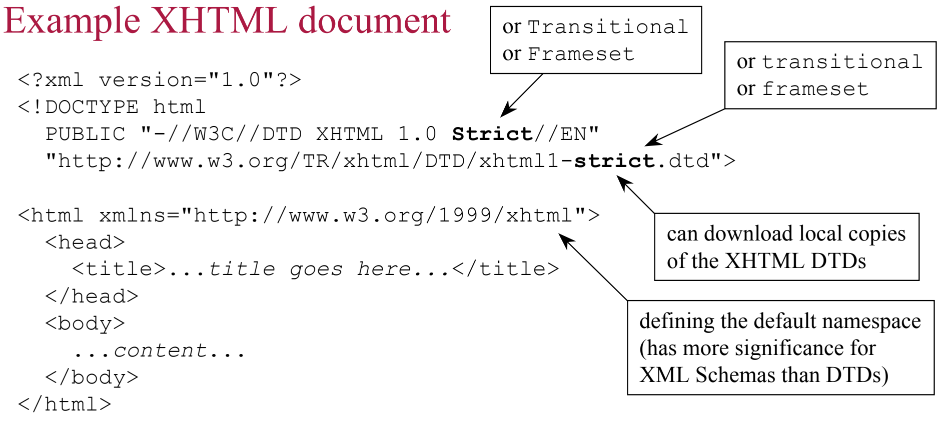
- **#PCDATA** is parsed character data
- can use HTML-style **comments**
- additional syntax is used to define allowed **attributes**, **entities** and **notations**

XHTML

- W3C set of DTDs describing HTML version 4.01
  - allows validation of HTML
  - **increases likelihood of correct rendering in multiple browsers**
- Three levels of conformance
  - strict
    - for XML documents that conform to HTML 4.01
    - all formatting must be specified using CSS
    - doesn’t support frames, deprecated tags and non-CSS formatting tags (such as font and center)
  - transitional (a.k.a. loose)
    - for XML documents that do not fully conform to HTML 4.01
    - supports deprecated tags and non-CSS formatting tags
    - doesn’t support frames
  - frameset
    - superset of transitional that adds frame support

XHTML (Cont'd)

- Example XHTML document



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

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>...title goes here...</title>
  </head>
  <body>
    ...content...
  </body>
</html>
```

or `Transitional`
or `Frameset`

or `transitional`
or `frameset`

can download local copies of the XHTML DTDs

defining the default namespace (has more significance for XML Schemas than DTDs)

XHTML is XML

- XHTML documents are XML documents
  which means they must be well-formed
- Introduces rules that are unfamilar to many HTML authors
  - tags are case-sensitive
    - all XHTML elements and attributes are lowercase
  - there must be a root element (`<html>`)
  - all tags must be closed
    - browsers may not support shorthand way of closing tags
    - for example, `<br/>` may confuse the browser but `<br />` will work
  - all attributes values must be enclosed in quotes, even numeric ones
  - all attributes must have a value
    - HTML attributes that don't need one, such as the `checked` attribute of the `input` tag which is used for checkboxes, use the attribute name as the value
      - for example, `<input type="checkbox" checked="checked" />`
      - fortunately there are few attributes like this in HTML (12)

XHTML is XML (Cont’d)

- Special characters
  - replace all occurrences of `&` with `&amp;`
  - replace all occurrences of < with `&lt;`
- Tag identifiers
  - most HTML tags support the `name` attribute which allows them to be referenced from JavaScript
  - XHTML deprecates this in favor of the `id` attribute which serves the same purpose
  - the `id` attribute is declared to be an attribute of type ID which forces all `id` values to be unique within a document

Converting HTML to XHTML

- Tidy
  - free download from www.w3.org/People/Raggett/tidy
  - can perform most of this conversion
  - closes unclosed tags
  - fixes mismatched end tags
  - fixes incorrect tag nesting
  - changes case of tags and elements to lowercase
  - adds missing quotes to attribute values
  - and more
- Steps to use
  - download
  - add directory to PATH or place in a directory already in PATH
  - `tidy` *`filename`*`.html >` *`filename`*`.xhtml`

Tidy Example

`tidy -i sample.html > sample.xhtml`

Example Input:

```
<HTML>
  <HEAD>
    <TITLE>My Web Page</TITLE>
  </HEAD>
  <BODY>
    <H1 align=center>
      Welcome To My Web Page!
    </H1>
    Here's a list of my interests.
    <UL>
      <LI>Java
      <LI>XML
      <LI>hockey
      <LI>tennis
      <LI>basketball
    </OL>
  </BODY>
</HTML>
```

Example Output:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
  <head>
    <meta name="generator"
     content="HTML Tidy, see www.w3.org">
    <title>My Web Page</title>
  </head>
  <body>
    <h1 align="center">Welcome To My Web Page!</h1>
    Here's a list of my interests.
    <ul>
      <li>Java</li>
      <li>XML</li>
      <li>hockey</li>
      <li>tennis</li>
      <li>basketball</li>
    </ul>
  </body>
</html>
```

Tidy does not add a default namespace declaration to the `html` tag!

the generated meta tag is not terminated!

Benefit of Stylesheets

- XML only describes content or data, not formatting or layout of data
- Formatting can be provided with separate CSS or XSL stylesheets
  - CSS currently can only output content in the XML and only in the order in which it appears
  - XSL is more flexible
  - CSS is useful in conjunction with XSL
- CSS can also be used with HTML to separate content from formatting
  - we’ll look at this option next

Using CSS with HTML or XHTML

- Remove all formatting from the HTML
- Create a CSS file containing formatting rules
  - composed of selectors and property lists
- Two ways to associate formatting with tags
  - use the class attribute
    - useful with HTML
  - match on tag names
    - useful with XML

Example XHTML

(see article.xhtml)

```
<?xml version="1.0"?>
<!DOCTYPE html
 PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"
>
<?xml-stylesheet type="text/css" href="articleHTML.css"?>

<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Tolkien Books</title>
  </head>
  <body>
    <div class="articleTitle">Tolkien Books</div>
    <div class="author">R. Mark Volkmann</div>
    <div class="abstract">
      This article describes some of the books written by
      J. R. R. Tolkien.
    </div>
```

associates a CSS stylesheet with the document

Note the use of the class attribute.

Example XHTML (Cont’d)

```
    <div class="sectionTitle">The Hobbit</div>
    <div class="paragraph">
      This book tells the tale of a Hobbit named Bilbo Baggins.
      He is visited by a wizard named Gandalf who asks him to
      join a group of dwarves on a quest to regain the treasure
      of their ancestors that was taken from them by a dragon
      named Smaug.
    </div>
    <div class="paragraph">
      They have many adventures along the way to the Misty Mountains
      where Smaug lives. The most significant of these is when Bilbo
      meets a slimy creature named Gollum, finds his special ring
      and escapes with it. This ring makes the wearer invisible.
      Bilbo uses the ring in order to survive many predicaments that
      arise during the trip.
    </div>
```

Example XHTML (Cont’d)

```
    <div class="sectionTitle">
      Lord of the Rings - Fellowship of the Ring
    </div>
    <div class="paragraph">
      This book tells the tale of another Hobbit, Frodo Baggins,
      who is Bilbo's cousin. Frodo embarks on a journey even more
      dangerous than the one Bilbo took in "The Hobbit".
      The importance of the ring that Bilbo found in that book
      is revealed in this book.
      Frodo takes the ring with him in his travels.
    </div>
    <div class="paragraph">
      There are three books in the "Lord of the Rings" series.
      This book leaves you wanting more.
    </div>
  </body>
</html>
```

CSS for Example XHTML

(see articleHTML.css)

```
body {
   background-color:      linen;
   border:                solid purple 5px;
   padding:               10px;
   width:                 100%;
}

head {
   display:               none;
}

.articleTitle {
   color:                 blue;
   display:               block;
   font-weight:           bold;
   font-size:             24pt;
}
```

Selectors that begin with a “dot” define classes.

CSS for Example XHTML (Cont’d)

```
.author {
   color:                 purple;
   display:               block;
   font-style:            italic;
   margin-bottom:         2ex;
}

.paragraph {
   display:               block;
   margin-top:            1ex;
}

.sectionTitle {
   background-color:      lightblue;
   color:                 purple;
   display:               block;
   font-size:             18pt;
   font-weight:           bold;
   margin-top:            2ex;
   padding:               0 5 0 5;
}
```

CSS for Example XML

(see articleXML.css)

```
article {
   background-color:      linen;
   border:                solid purple 5px;
   padding:               10px;
   width:                 100%;
}

article title {
   color:                 blue;
   display:               block;
   font-weight:           bold;
   font-size:             24pt;
}

author {
   color:                 purple;
   display:               block;
   font-style:            italic;
   margin-bottom:         2ex;
}
```

This produces the same output as on the previous page. Refer back to the XML on page 4.

CSS for XML (Cont'd)

```
paragraph {
   display:               block;
   margin-top:            1ex;
}

section title {
   background-color:      lightblue;
   color:                 purple;
   display:               block;
   font-size:             18pt;
   font-weight:           bold;
   margin-top:            2ex;
   padding:               0 5 0 5;
}
```

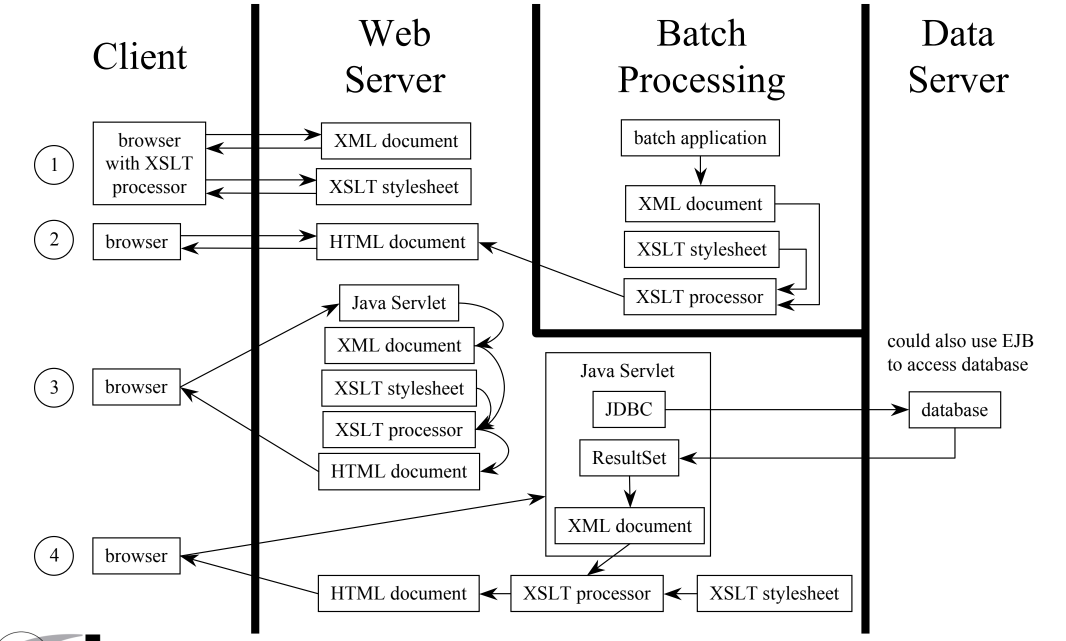
eXtensible Stylesheet Language (XSL)

- A more powerful alternative to CSS
- Transforms and formats XML
  - output can be new XML, HTML, XHTML or any text
- Two parts
  - tree transformation (XSLT)
    - can output content multiple times (useful for a table of contents)
    - can output content in a different order (specific or sorted order)
    - can add new data and structure (parent/child relationships)
    - can create a new version of a document that conforms to a different schema
    - XPath defines syntax for selecting nodes
  - formatting (XSL-FO)
    - creates a result tree whose nodes are XSL "formatting objects"
    - goal is to support creation of multiple media types from this
      - currently PDF is the primary media type supported
      - Apache FOP will be discussed later

XSLT Stylesheet Content

- Uses XML syntax, CSS doesn't
  - can edit, validate, and transform using standard XML tools
- Composed of templates that each contain
  - a pattern to be matched
  - the formatting objects to be output
- Two basic approaches
  - template-driven
    - relies primarily on procedural traversal of contents using `<xsl:if>`, `<xsl:for-each>` and `<xsl:choose>` constructs
    - typically results in fewer templates that are larger
  - data-driven
    - relies primarily on pattern matching
    - typically results in more templates that are smaller
    - usually the best approach; easier to understand and maintain

Many Ways To Apply XSLT



Client

Web Server

Batch Processing

Data Server

1

browser with XSLT processor

XML document

XSLT stylesheet

2

browser

HTML document

batch application

XML document

XSLT stylesheet

XSLT processor

3

browser

Java Servlet

XML document

XSLT stylesheet

XSLT processor

HTML document

could also use EJB to access database

Java Servlet

JDBC

database

ResultSet

XML document

4

browser

HTML document

XSLT processor

XSLT stylesheet

Data-Driven XSLT Example

(generates same output as earlier CSS example)

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <html>
      <head>
        <link rel="stylesheet" type="text/css" href="articleHTML.css"/>
        <title><xsl:value-of select="article/title"/></title>
      </head>
      <body>
        <xsl:apply-templates select="article"/>
      </body>
    </html>
  </xsl:template>
```

"/" matches document root

Data-Driven XSLT Example (Cont'd)

```
<xsl:template match="article">
  <div class="articleTitle"><xsl:value-of select="title"/></div>
  Abstract:
  <span class="abstract"><xsl:value-of select="abstract"/></span>
  <div class="author">by <xsl:value-of select="author"/></div>
  <xsl:apply-templates select="section"/>
</xsl:template>

<xsl:template match="section">
  <div class="sectionTitle">
    <xsl:value-of select="title"/>
    - Section <xsl:value-of select="position()"/>
  </div>
  <xsl:apply-templates select="paragraph"/>
</xsl:template>
```

order of output can be changed; text can be added

adds section number to each section title

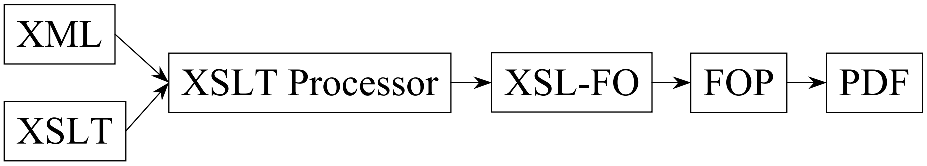
Data-Driven XSLT Example (Cont'd)

```
  <xsl:template match="paragraph">
    <div class="paragraph"><xsl:value-of select="."/></div>
  </xsl:template>

</xsl:stylesheet>
```

XSL-FO and FOP

- Currently, the most popular application supporting XSL-FO seems to be the Formatting Object Processor (FOP) from Apache
- FOP parses an XML document containing XSL formatting objects and produces a PDF file
- Ideal for producing printed output
- Can also be viewed in a web browser



XSL-FO Example

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0”
                xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
      <fo:layout-master-set>
        <fo:simple-page-master master-name="simple"
                               page-height="11in"
                               page-width="8.5in"
                               margin-top="0.5in"
                               margin-bottom="0.5in"
                               margin-left="1in"
                               margin-right="1in">
          <fo:region-body margin-top="0.5in"/> <!-- normal text -->
          <fo:region-before extent="0.5in"/> <!-- header -->
          <fo:region-after extent="0.5cm"/> <!-- footer -->
        </fo:simple-page-master>
      </fo:layout-master-set>
```

XSL-FO Example (Cont’d)

```
    <fo:page-sequence master-reference="simple">
      <fo:flow flow-name="xsl-region-body">
        <fo:block background-color="linen"
                  border-color="purple"
                  border-style="solid"
                  border-width="5px"
                  font-family="serif"
                  padding="10px">
          <xsl:apply-templates select="article"/>
        </fo:block>
      </fo:flow>
    </fo:page-sequence>
  </fo:root>
</xsl:template>
```

XSL-FO Example (Cont’d)

```
<xsl:template match="article">
  <fo:block font-size="24pt"
            font-weight="bold"
            line-height="24pt"
            color="blue">
    <xsl:value-of select="title"/>
  </fo:block>

  <fo:block>
    Abstract: <xsl:value-of select="abstract"/>
  </fo:block>

  <fo:block color="purple"
            font-style="italic"
            line-height="24pt">
    by <xsl:value-of select="author"/>
  </fo:block>

  <xsl:apply-templates select="section"/>
</xsl:template>
```

XSL-FO Example (Cont'd)

```
<xsl:template match="section">
  <fo:block background-color="lightblue"
            color="purple"
            font-size="18pt"
            font-weight="bold"
            padding-left="5px"
            padding-right="5px"
            space-before.optimum="10pt">
    <xsl:value-of select="title"/>
    - Section <xsl:value-of select="position()"/>
  </fo:block>

  <xsl:apply-templates select="paragraph"/>
</xsl:template>
```

XSL-FO Example (Cont'd)

```
  <xsl:template match="paragraph">
    <fo:block space-before.optimum="10pt">
      <xsl:value-of select="."/>
    </fo:block>
  </xsl:template>

</xsl:stylesheet>
```

FOP

- Open source XSL-FO processor from Apache
  – http://xml.apache.org/fop
- Supports the following output formats
  – PDF (used by Adobe Acrobat Reader)
  – PCL (Printer Control Language; used by some HP printers)
  – Postscript
  – Scalable Vector Graphics (SVG)
  – XML (just for testing)
  – print (goes directly to a specified printer)
  – AWT (Java GUI)
  – Marker Interchange Format (MIF; used by Adobe Framemaker)
  – Rich Text Format (RTF; not supported yet, but coming soon)

FOP Command-line Usage

- To run FOP, enter a command matching the following on a single line

```
– fop [-fo|-xml] infile
      [-xsl file]
      [-awt|-pdf|-mif|-pcl|-svg|-print]
      outfile
```

- Example

```
– fop -xml article.xml
      -xsl articleFO.xsl
      -pdf article.pdf
```

XML Spy

- Live demo