Session #5
Advanced Technical Writing
2005

Today in Class...
- Evaluation criteria for P1
- Meet with your editorial team, to prepare announcements/tips for authors
- Our readings on...er...reading
- Movie review markup show-n-tell
- XML wrap up (for now)

Editorial Teams
During session 6, you will review posted articles and offer the authors feedback related to your editorial role.

Use your meeting time today to discuss what you’ll be looking for, and to frame some tips for authors as they prepare their pieces for review. If you have lots of advice or if your advice has technical details that are important, write it up, post it, and send Bill H-D the link.

Show-n-Tell...and our readings for today
Show us your color coded instance of a movie review, walking us through the objects you have identified.

As you do this, tell us about what users’ interactions with these objects are likely to be and how these shaped your decisions.

Refer to Wright and Williams & Spyridakis when you can. (e.g. W&S describe basic relations among text elements that correspond to readers’ tasks, but they focus on one type of task and thus one type of element and its role in supporting that task)
Isn’t this all a bit too complicated?

Well...

Yes and no. At the syntax level, certainly. But at the conceptual level, the move to represent information semantically and then transform it for any number of display formats is a move to simplify things a bit...it’s the implementation that makes it tricky.

XML+DTD+XSL(T)= HTML

“Every IED concept report should be made up of a ‘heading’ and a ‘body’”

```
<xsl:template match="IED_Concept_Type_Report">
  <html>
    <head>
      <title><xsl:value-of select="/IED_Concept_Type_Report/Heading/Title" /></title>
    </head>
    <body>
      <xsl:apply-templates select="Heading" />
      <xsl:apply-templates select="Body" />
    </body>
  </html>
</xsl:template>
```
“The ‘heading’ should display the ‘title’, as defined in an XML tag of that name, and some info about ‘Authors’; we have some guidelines for ‘Authors’ below”

```xml
<xsl:template match="Heading">
  <h2><xsl:value-of select="Title" /></h2>
  <xsl:apply-templates select="Authors" />
</xsl:template>
```

“Create a table which will have the words “written by” in the top row and then look in the XML for any ‘Author’ objects and make a new row”

```xml
<xsl:template match="Authors">
  <table cellpadding="0" cellspacing="0" bgcolor="#DDDDFF" border="0" width="100%">
    <tr><th align="left">Written By</th></tr>
    <xsl:for-each select="Author">
      <tr><td align="center">First-Name</td><td align="center">Last-Name</td></tr>
    </xsl:for-each>
  </table>
</xsl:template>
```

“Make a cell to contain whatever is inside the ‘First-Name’ and ‘Last-Name’ tags; we’re done with the ‘Author’ table now.”

```xml
<td align="center">
  <xsl:value-of select="First-Name" />
  <xsl:value-of select="Last-Name" />
</td>
</tr>
</xsl:for-each>
</table>
</xsl:template>
```

A pretty good example...

Often re-published, the examples below show both CSS and XSL styling:

- [http://www.w3schools.com/xml/xml_display.asp](http://www.w3schools.com/xml/xml_display.asp)
- [http://www.w3schools.com/xml/xml_xsl.asp](http://www.w3schools.com/xml/xml_xsl.asp)

Whether they work for you or not will depend on your browser...
Create an information model (you can use some variant of DTD syntax) for “Family Film Reviews” that would allow you to transform an XML-formatted review like the one you have into two distinct "views":

One for parents
One for kids

Then...sketch the views, labeling the objects that would make them up. Post them for session 7 (not next time). You can consider this a dry-run for your next major project. You can go whole hog if you want...and do DTD>XML>XSLT>XHTML...or go lo-tech.

Post your P1 article for in-class review session
Post your movie review information model for session 7 (Tuesday 2/1)