

**B336 Advanced Internet Computing**

**Designing XML Solutions  
&  
Example XML Applications**

# Learning Objectives

- Learn how the XML components we have been discussing can be put together in solutions.
- Learn from some example real-world applications of XML.

# Learning Objectives

- In the scheme of what we are doing in this unit:
  - We are studying how to use XML as an important set of Internet technologies to use as solutions in different areas (for example, the problem defined in Assignment 2).
  - There are a few key considerations we should always keep in mind when designing XML solutions.
  - 
  - We can learn a lot about how to effectively use XML technologies by looking at some examples of how they have been applied to other problems.

# Lecture Outline

- The components of an XML solution.
- Some examples of existing XML applications:
  - XML for streaming multimedia
  - XML for mathematical manipulation
  - XML for bioinformatics and genetic research
  - XHTML

# Designing XML Solutions

- The purpose behind the XML document is to describe data.
  - The data could be the normal declarative ones, like financial details, personnel details, etc.
    - » ie. ones of the form “this data has the value...”,
  - Or they could be commands and instructions
    - » eg. XSLT is a XML application
    - » Think about your favourite programming language in XML form, and what you can do with source-code manipulation if you have it in that form.

# Advantages of an XML solution

- Although anything you want to do using XML technologies, you can do using alternatives that do not involve XML, you do lose some advantages that XML offers:
  - Public standard
  - Wide acceptance
  - Ease of creating a new structure and description language
  - A common platform, which creates
    - » Large based of new developments
    - » Abundance of reusable components, and
    - » wide-spread support

# Public Standard

- Information in a network environment will always originate from different platforms (different applications, different OS, different hardware)
- Yet that information almost always needs to be usable on other platforms.
- For public information to remain easily exchangeable, it cannot afford to be restricted to one make or model or manufacturer, or to lose control of its data format to private hands.

# A Complete XML Solution

- To design an effective XML solution, we need to consider the following components:
  - Defining the language syntax for the documents, using DTD
    - » ...or XML Schema, if you prefer.
  - Ability to convert to other formats (eg. HTML)
    - » Using XSL and XSLT
  - Parsing and Processing tools
    - » to allow programmers to create effective software, which in turn...
    - » allow users to do productive tasks in the particular area XML is applied.
  - Display tools
    - » Since most data needs to be showed to human users at some stage.
    - » The only exception to needing display tools is when the data is mean for software-to-software, or machine-to-machine communication.

# Display Tools

- The most obvious candidate for display tool is the web browser
  - Since so much data these days are transferred over the web (ie. requested and received by an HTTP client, sent by a HTTP server).
  - People are familiar with the basic interface and operations of standard web browsers.
  - It is easy to make use of the web browser to do display by doing XML-to-HTML style sheet conversions (at the server-side or the client-side).

# Browsers as Display Tools

- If web browsers' HTML is not capable of doing the complex display a particular XML application requires, then there are a few options:
  - Build a Java applet
  - Build a plug-in for each standard browser
  - Convert to another display format (eg. GIF or JPG images) at the server-side before sending.
  - Build your own display tool.
- Many of today's major XML applications have their own display tools, and these display tools are also called "browsers" (but not "web browsers").

# Reference

- Read section 12 of the unit reader for more discussions about the issues I have outline above.
  - The information will be useful for your Assignment 2.

# Example XML Applications

- We will have a look at the following example XML applications:
  - Synchronized Multimedia Integration Language (SMIL)
    - » for streaming multimedia
  - Mathematical Markup Language (MathML)
    - » for manipulating mathematical expressions
  - XML in Bioinformatics and Genetic Research
  - XHTML
    - » **the XML version of HTML**

# SMIL

- Synchronized Multimedia Integration Language (SMIL) is a language defined by W3C
  - to “enable simple authoring of TV-like multimedia presentations... on the Web”.
  - The language contains mark-ups to define information about how the multimedia components (eg. images, sounds, videos) are to be synchronized.

# An Example SMIL Document

```
<?xml version="1.0">
<smil>
  <head>
    <meta name="title" content="A SMIL Document" />
    <region id="image_region"
      left="0" top="0" height="200" width="200" />
  </head>
  <body>
    <seq>
      
      <par>
        <audio src="track.wav" />
        
      </par>
    </seq>
  </body>
</smil>
```

Present  
in sequence

Present  
in parallel

# Creating SMIL Documents

- Some example authoring tools for creating SMIL documents:
  - Sausage Software's SMIL Composers
  - Allaire's HomeSite
  - RealNetworks' RealProducer G2 Authoring Kit
  - Adobe have also announced that they will support SMIL content creation in its GoLive web authoring tool in the future.
- These software generate SMIL documents by allowing users to use a WYSIWYG graphical interface to define a streaming multimedia presentation.

# SMIL Browsers

- Example popular multimedia players now have support for SMIL (ie. they can show the presentation as described by a SMIL document)
  - Apple Quicktime 4.1
  - RealNetworks' RealPlayer 8
  - Internet Explorer 5.5
- We also have browsers which are specifically created for SMIL presentations
  - X-Smiles
  - Grins

# Reference

- For a more complete look of SMIL technologies, start at W3C's SMIL page:
  - <http://www.w3.org/AudioVideo/>

# MathML

- MathML is intended to facilitate the use and re-use of mathematical and scientific content on the Web, and for other applications such as computer algebra systems, print typesetting, and voice synthesis.
- It can describe both
  - the presentation of mathematical notation for high-quality visual display, and
  - the actual mathematical content.

# Why MathML?

- HTML does not support display of mathematical expressions very well.
  - Besides the few ASCII symbols and operators we find on the keyboard
- With so much information moved onto the Web, the scientific community finds it very restrictive not being able to share basic mathematical descriptions.
  - They resorted to doing things like scanning a page with the mathematical expressions into images files and exporting the images.

# Example MathML

$$x^2 + 4x + 4$$

```
<apply>
  <plus/>
  <apply>
    <power/>
    <ci>x</ci>
    <cn>2</cn>
  </apply>
  <apply>
    <times/>
    <cn>4</cn>
    <ci>x</ci>
  </apply>
  <cn>4</cn>
</apply>
```

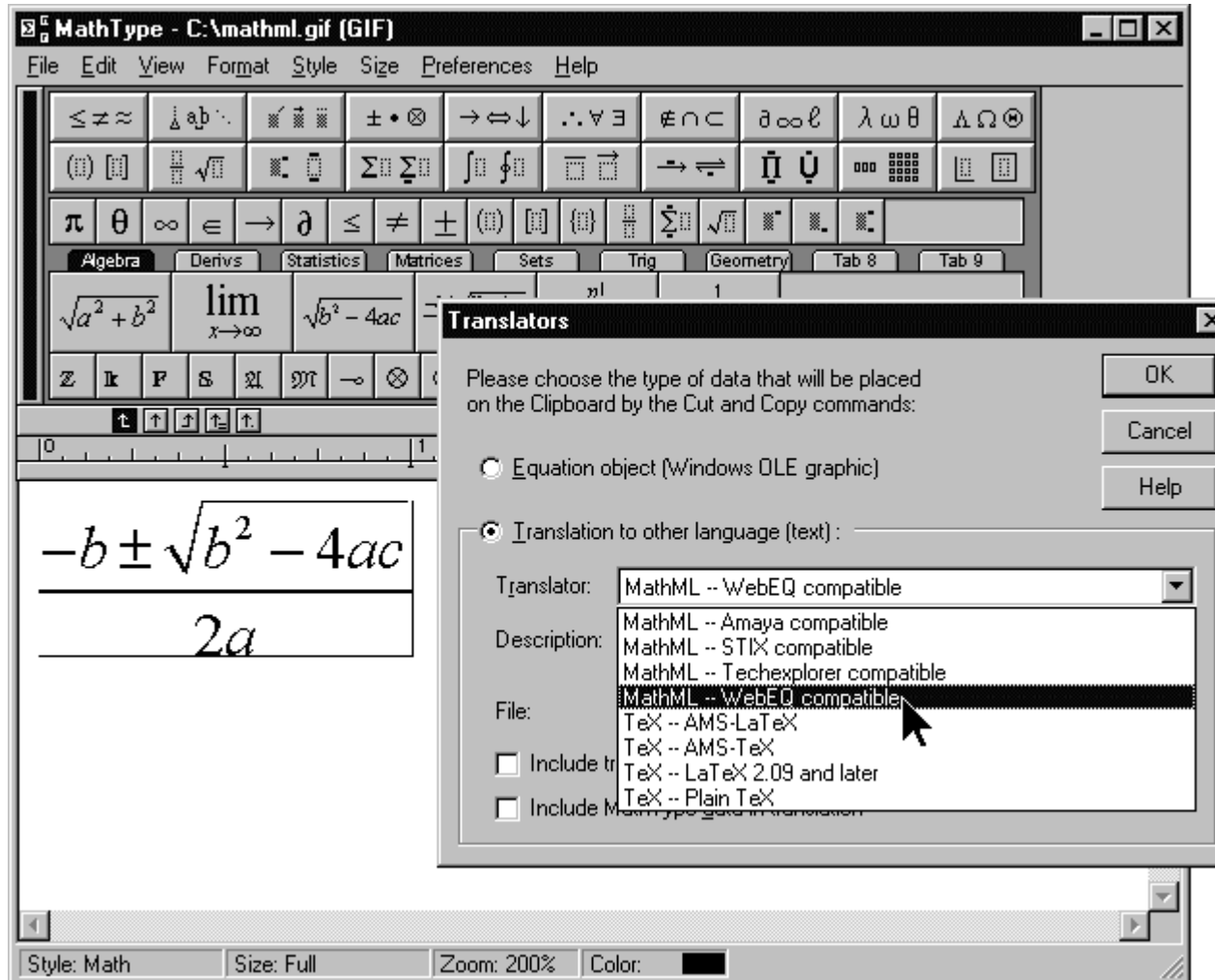
# Example MathML Software

- Amaya
  - W3C's web browser, displays MathML and includes an editor for MathML.
- Maple:
  - a popular computer algebra system
  - import, export and evaluate MathML content
  - in the future, support generation and rendering of MathML
- Mathematica:
  - another popular maths manipulation system
  - contains a visual typesetting and authoring tool which both renders and exports MathML.

# Example MathML Software

- **MathType:**
  - a stand-alone equation editor and authoring tool for MathML
- **LaTeX2HTML MathML package:**
  - LaTeX to MathML conversion
- **LiveMath:**
  - an interactive mathematics plug-in
- **MML:**
  - an Equation Editing applet for Windows

# Example Software



## The MathType package

For a clearer picture, see source:  
<http://www.w3.org/Math/implementations.html>

# MathML in Standard Web Browsers

- The ultimate aim of MathML is for it to be natively render in all standard web browsers using style sheets (CSS and XSL).

# Reference

- For an overview of MathML, see
  - MathML technologies
    - » <http://www.w3.org/Math/>
  - MathML Software, including screen shots
    - » <http://www.w3.org/Math/implementations.html>

# XML for Bioinformatics

- Bioinformatics is the application of computing to biology but is mostly used to analyze genomic and genetic data.
- Good data management and processing tools is now critical due to the amount of data coming out of DNA sequencing efforts such as the Human Genome Project.
- XML provides a good platform to formulate these data management solutions.

# Some Example XML Applications for Bioinformatics

- The following languages describe information in molecular biology and bioinformatics:
  - BIOpolymer Markup Language (BIOML)
    - » for the annotation of biopolymer sequence information.
    - » Browser: BioBrow – The BIOML Browser
    - » Ref: <http://www.bioml.com/BIOML/index.html>
  - Bioinformatic Sequence Markup Language (BSML)
    - » for the representation of molecular biological data
    - » developed by Visual Genomics
    - » BSML web browser
    - » Ref: <http://visualgenomics.com/>

# Example BSML Software

The screenshot displays the Genomic XML Viewer v1.0 interface, a web-based application for viewing genomic data. The interface is organized into several sections:

- Top Bar:** Includes the LabBank logo, current sequence and feature information, and a menu with options like Document, Sequence, Feature, and Navigation.
- Header:** Displays "Genomic XML Viewer™ v1.0 January 2001".
- Left Sidebar:** Contains navigation links for XML Data Standards, Software Updates, and About LabBank, along with a description of the software's capabilities.
- Main Content Area:** Features a "Monthly Focus" section with an article titled "XML View to biotechnology community..." and a circular diagram. Below this is a section for "BSML - The new standard in biosequence analysis" with a link for more information.
- Right Panel:** A "Help" section titled "Using the Genomic Viewer" with an "Introduction" subsection, explaining the software's purpose and how to use the toolbar.

# Bio.Pperl.Org

- A site and project dedicated to the development of PERL tools for molecular biology computing.
  - Maintains a bio.perl package
- The project is looking to incorporate XML technology into the BIO.PERL toolkit
  - Create new modules based on XML::Parser, XML::Parser.DOM etc which support bioinformatics data manipulation
- Ref: <http://bio.perl.org/Projects/XML>

# Reference

- For an overview of available XML technologies in bioinformatics
  - <http://www.cpb.uokhsc.edu/ojvr/xmlpaper.html>
    - » The paper uses the expression “Extended” Markup Language to refer to XML, which is wrong.

# XHTML

- The eXtensible HTML (XHTML) is W3C's recommendation for the latest version of HTML,
- XHTML 1.0 is a reformulation of HTML 4.01 in XML, and combines the strength of HTML 4 with the power of XML.

# XHTML Syntax

- The syntax for XHTML should be very familiar to you who know HTML. It just adds all the restrictions placed by XML. Eg
  - All opening tags must have a closing tag, or a “/” at the end of the opening tag.
  - No overlapping tags.
  - One root element (the <html> tag).
  - Etc.

# Why XHTML?

- The principal aim of XHTML is to provide richer and more manipulatable Web pages, so that it can be passed to on an ever increasing range of browser platforms including cell phones, televisions, cars, wallet sized wireless communicators, kiosks, and desktops.

# Reference

- For an overview of XHTML development
  - <http://www.w3.org/MarkUp/>

# What can we learn?

- From all these examples, we can see some common threads
  - The creation of a good standard description for the data in their problem domains
  - Having a good description mechanism allows people and software to easily exchange information
  - The common XML platform facilitates the creation and sharing of software created to increase the productivity of people in these fields.