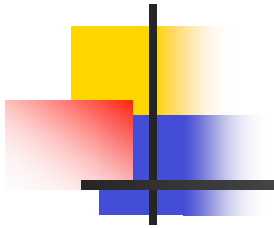
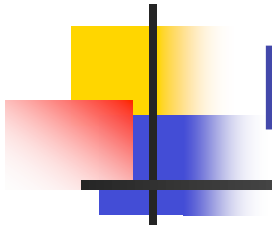


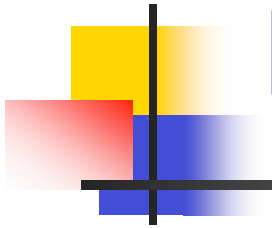
Designing XML Solutions

(Week 9 Lecture 2a)



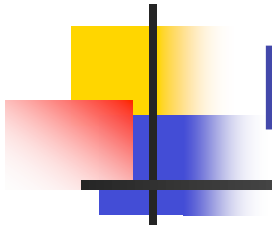
Learning Objectives

- Learn how the XML components we have been discussing can be put together in solutions.



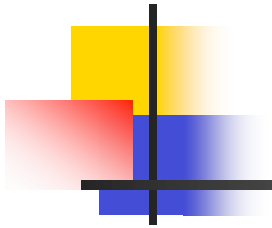
Learning Objectives

- 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.
  - It is important to know how to put the technologies together to provide a useful environment for data exchange and manipulation.



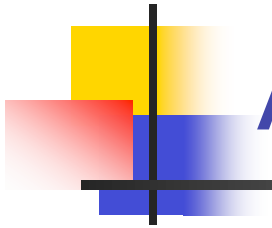
Lecture Outline

- Why and XML Solution?
- Components of an XML Solution.



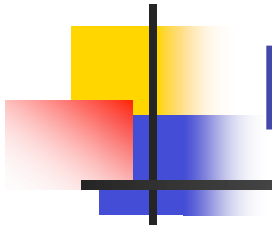
Designing XML Solutions

- 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.



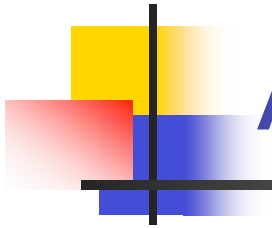
Advantages of an XML solution

- 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



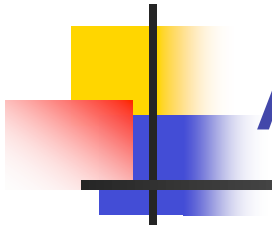
Public Standard

- 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.



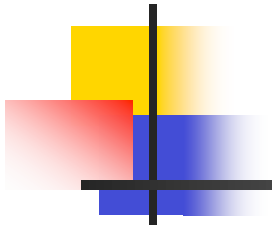
A Complete XML Solution

- 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



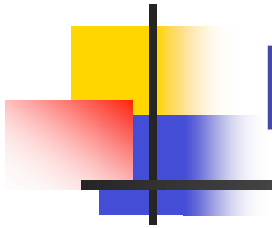
A Complete XML Solution

- 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

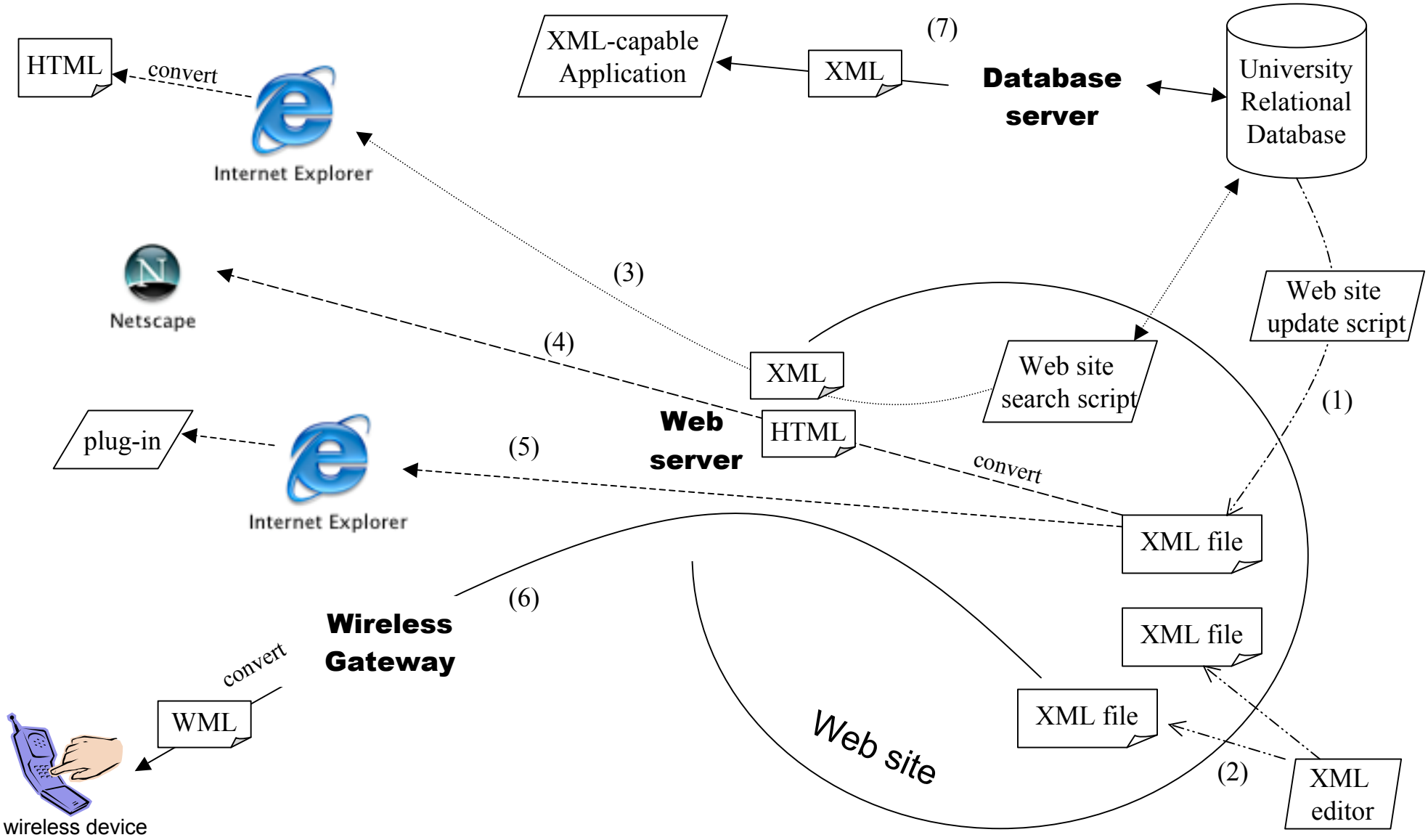
- The most obvious candidate for display tool is the web browser
  - Since so much data these days are transferred over the web (ie. requested and received by an HTTP client, sent by a HTTP server).
  - People are familiar with the basic interface and operations of standard web browsers.
  - It is easy 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).

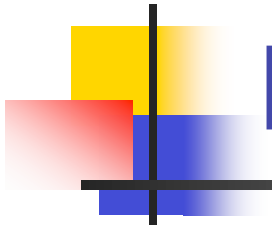


Browsers as Display Tools

- 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.

Example course XML from week 5 lecture 1:





Example

- From the previous diagram:

(1) A daily update script extracts course information from a relational database and generates XML files to put on a web site.

(2) A data-entry person uses an XML editor to create new XML files with course information.

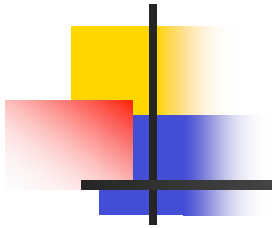
(3) A user surfing the web site uses the search facility to find some course information. The search script queries the database, gets back the results, and sends the results back to the browser in XML. Browser converts the XML to appropriate HTML and displays.

(4) A user using a web browser accesses some course information. The web server converts the XML document to HTML and sends HTML back to the browser. Browser displays HTML as normal.

(5) A user using a web browser with a special course-info plug-in accesses some course information. The web server sends the XML document directly to the browser. The browser uses the plug-in to display the course information.

(6) A user on a wireless device accesses some course information. The XML document is sent to the user's wireless gateway, converted to WML and then sent to the wireless device. Device displays WML as normal.

(7) An XML-enabled application connects to the database and retrieves some course information. The database server sends results back in XML.



Further Reading

- Read textbook chapter 5 for more design issues not covered in this lecture.