7 XML Web-Site Architectures

- How (and why) to apply XML techniques in the implementation of Web sites?

7.1 XML, Databases, and Multi-tiered Systems
- Overview of Web-based applications built using XML & database technology and multi-tiered architectures
- Later:
  7.2 Apache Cocoon publishing framework

Advantages of XML based solutions?

- Creating and delivering final document representations (HTML, PDF, ...) more straightforward than using XML
- Why then XML-based web sites?
  - control of uniform style (via style sheets)
  - processing of semantic content (XML) on the client
  - utilising XML processing capabilities on the server
  - tailoring different versions of the same content
    - for different browsers, different user groups, different uses (browsing, printing, audio, ...)

XML and Databases

- How to store and to manage XML documents?
  Depends on application requirements:
  - file system
    - simple and cheap
    - often sufficient, especially if collection small and static
  - database management systems
    - provide efficient access to large data collections
    - more appropriate when documents dynamically modified, especially if simultaneously by multiple users
    - provide consistency and integrity preserving "ACID" transactions

Database Transactions

- Transaction
  - semantically meaningful unit of update operations
  - e.g., money transfer btw two bank accounts
- ACID properties of transactions
  - Atomicity: either all changes or none are committed
  - Consistency: transactions maintain consistency of data
  - Isolation: transactions do not interfere with each other
  - Durability: committed updates are stored permanently

Types of XML databases

1. XML generating databases
   - database not aware of XML, which is just an output format for data extracted
     (using SQL/OQL/JSP/...)
2. XML document databases
   - fields for entire documents or document fragments
   - metadata fields to support management of these chunks
3. XML component databases
   - "true XML databases"
     - store document components as separate objects
     - schema for document objects corresponds to DTD
Commercial XML databases

- Steve Pepper's "Whirlwind guide" of SGML/XML tools (Dec. 99) mentioned
  - 10 document DB managers; 1 with XML support
    - SigmaLink of STEP GmbH
  - 6 component DB managers, 3 with XML support
    - Astoria, POET Content Mgmt Suite, TARGET 2000
    - strong development activity underway by major O-O and
      relational database vendors
- Prizes of high-end XML database systems around USD 40,000 - 100,000 (including configuration, installation and
  integration to business processes)

Commercial XML databases (2)

- Ronald Bourret, Nov. 2000:
  http://www.rpbourret.com/xml/XMLDatabaseProds.htm
  - 5 XML-enabled databases
    - XML generating/document databases
    - DBX XML Extender, MS SQL Server 2000, Oracle 8i & 9i, ...
  - 8 native XML databases
    - XML component databases (?)
  - 3 RD prototypes, rest commercial products (?)
  - Also middleware, XML servers, XML application
    servers, content management systems, persistent
    DOM implementations: classification often difficult

Three-tiered architectures

- Typical architecture of web-based data
  delivery/processing applications:
  - client tier (or layer)
    - standard web browser (+ possible additions)
  - middle tier / applications layer
    - "business logic" running on server
    - can be viewed or realised as number of separate
      tiers
    - "multitiered architectures"
  - data tier (or layer)
    - storage and maintenance of data

Why three-tiered architectures?

- HTTP client + server + data source is a
  convenient implementation environment
  - familiar browser user-interface
  - standard components (browsers & servers)
    - cheap (free)
    - available on various OS & HW platforms
  - minimal start-up pain for new users or customers

Where XML gets processed?

- Two major strategies:
  1. to deliver XML to client
    - and to use, say,
      SAX/DOM-based applets or DOM-based scripts for
      processing, and
      CSS, or XSLT->HTML(+CSS) for presentation
  2. to process XML on the server
    - and to deliver plain HTML (+CSS) to client
- Which one to choose?

Delivering XML to web clients

- Advantages:
  - single semantically marked-up XML document may
    correspond to multiple single-purpose HTML pages
  - reduces network traffic and moves processing load
    from server to client
- Disadvantages:
  - assumes XML capable browsers
  - application development for unknown client
    environments difficult
Delivering XML to web clients (2)

- Disadvantages (continued)
  - Exposure of data structures (XML) and “business logic” (applets or scripts) to customers
- XML processing on client may be a viable solution in controlled environments like intranets (clients uniform, users trusted)
- “XML on server, HTML to client” strategy gained more popularity recently

Processing XML on server

- Plain HTML delivered to clients
- Advantages:
  - Support of non-XML-capable browsers
  - Ability to address wider audiences
  - “Semantic firewall”
  - Control over processing environment (server)
  - Control over performance

Processing XML on server (2)

- Disadvantages:
  - May increase data traffic: a new HTML page for each different view on the document
  - Fails to utilise (often idle) processing cycles of client computers
  - Heavier network and server load
- Combinations of client/server processing also possible: Delivering
  - “Downgraded” XML to clients
  - XML to selected clients only