Introduction to XML

What Is Markup?

- *Information added to a text to make its structure comprehensible*
- *Pre-computer markup (punctuational and presentational)*
  - *Word divisions*
  - *Punctuation*
  - *Copy-editor and typesetters marks*
  - *Formatting conventions*

The Friendly letter

- *This shows something about what third graders learn about reading and writing*
- *That documents are alike in key ways*
- *That they have parts, with names*
- *That those parts are (usually) distinctively displayed*

Computer markup

- *Any kind of codes added to a document*
  - *Typesetting (presentational markup)*
    - *MS Word, TeX, LaTeX, etc.*
  - *Declarative markup*
    - *HTML (sometimes)*
    - *XML*

What do we mean by declarative?

- *Names and structure*
- *Finer level of detail (most human-legible signals are overloaded)*
- *Independent of presentation (abstract)*
- *People often call this “semantic”*

XML

- *The Extensible Markup Language*
- *XML is a standard, interoperable way to represent documents for flexible processing*
  - *Multi-format delivery*
  - *Schema-aware information retrieval*
  - *Transformation and dynamic data customization*
  - *Archival: standardized, self-describing*

The two worlds of XML

- *Markup of documents: the original*
  - *This perspective is our focus here*
  - *Document representation was the primary problem XML was created to solve*
- *Data exchange and protocol design*
  - *XML turned out to fill important gaps*
  - *Relational databases needed a way to share records and multi-table data*
  - *Protocol designers wanted a way to encapsulate structured data*

The two worlds united

- ***Documents and "semi-structured" data share features***
  - ***Hierarchical structure***
  - ***String content***
  - ***Variations in structure***
- ***Their applications also share needs***
  - ***Need for a lingua franca, independent of APIs***
  - ***Ability to cope with international characters***
  - ***"Fit" with WWW and HTTP.***

XML is more general

- *Tags label arbitrary information units*
  - *More suited to multiple purposes*
  - *“Looking right” is needed but not enough*
- *Supports custom information structures*
  - *If you have “price” or “procedure”, you can make a tag for it, and validate its usage*
- *More “teeth” to enforce consistent syntax*
  - *Works hard to avoid semi-interoperable docs*

Better rendering than HTML

- *Fully internationalized*
  - *Also better for visually-impaired users*
- *Supports multiple renderings*
  - *Customize to the user, time, situation, device*
  - *Separates formatting from structure*
  - *And processing other than rendering*
- *Large documents don’t break it*
  - *Easy to trade off server/client work*
  - *No searches that fail because big doc was split*
- *XHTML is XML-conforming flavor of HTML*
  - *Clean existing HTML is already close...*

XML treats documents like databases

- *XML brings benefits of DBs to documents*
  - *Schema to model information directly*
  - *Formal validation, locking, versioning, rollback...*
- *But*
  - *Not all traditional database concepts map cleanly, because documents are fundamentally different in some ways*

What is structure

- *To Relational Database theorists, structure is:*
  - *Tables with fixed sets of non-repeating named fields, that have little internal structure*
  - *E-R diagrams with fixed number of nodes*
- *Structured documents are different:*
  - *The order of Sections, Paragraphs, etc. matters (a lot)*
  - *Many hierarchical layers (which text crosses)*
  - *Text/graphic data mixes with aggregate objects*
  - *Optional or repeatable sub-parts abound*
  - *Interaction with natural language phenomena*
- *These are very different requirements*

What’s the difference?

- ***Without structure***
  - *Data conversion is far more expensive*
  - *Multi-platform and/or multi-media delivery require re-authoring and hand-work*
  - *Paper production is inconsistent*
  - *Late format changes are far more risky*
  - *Retrieval is prone to many false hits*
- *“Pay me now, or pay me later”*

XML design principles

- *Straightforwardly usable over the Internet*
- *Support for a wide variety of applications*
- *Compatible with SGML*
- *Make writing XML programs easy*
- *Avoid optional features*
- *Human-readable (if not terse) markup*
- *Formal and concise design*
- *Design produced quickly*

Opportunities with XML

- *Scalability and openness of Web solutions*
- *“Rich clients” for complex information*
  - *Dynamic user views*
- *XML as interprocess communication protocol for “data” (as opposed to “text”)*
- *eCommerce integration*
- *New methods of creation*
  - *Schema combination/composition*
  - *Free-form, schema-less data development*

Web usage

- *XML works with familiar Web paradigms*
  - *Locations are expressed as URIs*
  - *High interoperability because of few options*
  - *Easily implementable and usable*
  - *Robust against network failures*
  - *Avoids serving schemas every time with documents*
    - *(but can do better validation anyway, when needed)*

Some additional XML details

- *Well-formedness*
- *Error handling*
- *Case sensitivity*
- *HTML compatibility*

Well-formedness

- *Document has a single root element, and*
- *Elements nest properly*
  - *Try* <B>foo<I>bar</B>baz</I> *in your browser!*
- *Entities are whole subtrees (not </P><P>)*
- *No tag omission (close what you open)*
- *Attributes must be quoted*
- *< and & must always be escaped in some way*
- *A document can be well-formed (and parsable) whether or not it fits a given schema*

Partial and missing DTDs

- *DTDs (schemas) are needed for validation*
- *DTD processing adds a burden*
- *Because of Well-formedness,*
  - *DTDs are not needed just to parse*
  - *Even subtrees can be parsed in isolation*
    - *One exception: Default attributes*
- *Very handy for development/experimentation*

Error handling

- *“Draconian error handling”*
  - *Major errors cause processor to stop passing data in the “normal way”*
- *Fatal errors:*
  - *Ill-formed document*
  - *Certain entity references in incorrect places*
  - *Misplaced character-encoding declarations*
- *This helps save huge $ on error-recovery*
  - *Hopefully, the $ will go to better features instead*

Case sensitivity

- *HTML is*
  - *Case-insensitive for tag names:* **<P> = <p>**
  - *Case-sensitive for entity names:* **&LT; ≠ &lt;**
- *XML is case-sensitive for both!*
  - *Unicode standard advises against case-folding*
  - *Folding is not well-defined for all languages*
    - *Turkish has two lower-case i's, only one upper*
    - *In languages with no accented caps, can't reverse*
    - *Error-prone for programmers*
- *XHTML uses lower case*

Summary

- *XML has:*
  - *Representational power and extensibility*
    - *Custom tags, order constraints, etc.*
  - *Validation and consistency (several ways)*
  - *Much of HTML's simplicity for users/implementors*
- *XML trashes:*
  - *SGML's syntax/feature complexity*
  - *SGML's high startup costs*
  - *HTML's inflexibility*
  - *ASCII legacy*

Well Formed XML

- ***Markup***
  - *Prolog & Document Type Declaration*
  - *Elements*
  - *Attributes*
- ***Content***
  - *Entities*
  - *Encoded (Unicode) characters*

Prolog & Document Type Definition

- *XML documents should begin with an XML Declaration which specifies version*
  - *Optionally may also include:*
    - *Encoding (recommended)*
    - *Stand-alone declaration*
- *Document Type Definition is typically next*

- **<?xml version="1.0" encoding='UTF-8' standalone='no' ?>**
- **<!DOCTYPE root SYSTEM "myDocs.dtd" >**

Elements

- ***Elements are markup that enclose content***
- **<element_name>…</element_name>**
  or <element_name />
- ***Content models***
- *Parsed Character Data Only*
- *Child Elements Only*
- *Mixed*
- *Empty*
- **<author>Cole, T</author>**

Elements

- **Root**
  - *There is exactly one element, called the root, or document element, no part of which appears in the content of any other element.*
- **`<book>This is a book</book>`**

-
  ```
     <list>
    <item>Item 1</item>
    <item>Item 2</item>
    <item>Item 3</item>
  </list>
  ```
- Proper nesting of tag is required

Permitted Characters

- *Element names can contain letters, digits, hyphens, underscores, colons, or full stops*

-
  ```
  <permittedNames>
       <name/>
       <xsl:copy-of/>
       <A_long_element_name/>
       <A.name.separated.with.full.stops/>
       <a123323123-231-231/>
       <_12/>
  </permittedNames>
  ```

Forbidden Names

-
  ```
  <forbiddenNames>
       <A;name/>
       <last@name>
       <@#$%^()%+?=/>
       <A*2/>
       <1ex/>
  </forbiddenNames>
  ```
- **Names can not start with xml**

  -
    ```
    <forbiddenNames>
         <xmlTag/>
         <XMLTag/>
         <XmLTag/>
         <xMlTag/>
         <xmLTag/>
    </forbiddenNames>
    ```

Attributes

- *Associate a name-value pair with an element*
- **<tag name1="value1" name2='value2'>...</tag>**
  - *Can be used to embellish content...*
  - *or to associate added content to an element*
- *Attributes beginning XML are reserved*
  - **<author order='1'>**Cole, T**</author>**
    - **<author name='Habing, T' />**

```
<elements-with-attributes>
     <el _ok = "yes" />
     <one attr = "a value"/>
     <several first="1" second = '2'
third= "333"/>
     <apos_quote case1="John's" case2
='He said: "Hello, world!"  '/>
</elements-with-attributes>
```

Entities

- *Placeholders for internal or external content*
    - *Placeholder for a single character…*
    - *or string of text…*
    - *or external content (images, audio, etc.)*
- *Implementation specifics may vary*

- `<!ENTITY copyright "&#xA9;" >`
- `&copyright; is replaced by ©`
- `<!ENTITY pic SYSTEM "mugshot.gif" NDATA gif >`
- `&pic; is replaced by graphic image`

Special Characters

- *Characters < and & cannot be used in text as they are used in markup*

```
<example>
     <isLower>
          23 &lt; 46
     </isLower>
     <ampersand>
          Willey &amp; sons
     </ampersand>
</example>
```

Assignment #1

- *Write a report on the various way XML documents can be processed and parsed in JAVA (there are different API available to do this)*
  - *BS Project Groups*
  - *Deadline: 20 May, 2010 1500hrs*
  - *Online Submission*