

eXtensible Markup Language

An introduction in XML and parsing XML

XML

- Overview
- XML Components
- Document Type Definition (DTD)
- Attributes and Tags
- An XML schema

Overview

- XML is a set of related technologies
 - SGML
 - HTML
 - SAX
 - DOM
 - ...
- <http://www.w3c.org/XML>

Overview

- XML is a meta-language
 - describes the structure and content of a document
- XML does not specify the grammar of a document
 - eg. Set of tags - only have meaning to a specific language processor
 - eg. Correct use of tags

Applications

- Anywhere where data needs to be stored/retrieved
 - configuration files
 - data exchange
 - electronic business (ebXML)
 - messaging – Simple Object Access Protocol (SOAP)
 - Chemical Markup Language (CML)
 - ...

XML and HTML

- XML separates content from presentation
 - HTML specifies the presentation
- HTML defines a set of legal tags and grammar
 - eg. <h2>XML and HTML</h2>
- XML and HTML are based on SGML
 - Standard Generalized Markup Language

XML and HTML

- XML allows any set of tags to be used
 - M = meta
- XML describes data
- HTML displays data

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.4.1" class="java.beans.XMLDecoder">
  <object class="java.util.HashMap">
    <void method="put">
      <string>image</string>
      <string>images/house.gif</string>
    </void>
  </object>
</java>
```

XML encoding and decoding avoids the problems with serialisation

Components

- Prologue
 - Defines the XML version, entity definitions and DOCTYPE
eg.
`<?xml version="1.0" encoding="UTF-8"?>`
 - version - of XML
 - encoding - character set used
 - standalone identifies if an external Document Type Declaration (DTD) is used

Document Type Declaration (DTD)

- May be declared internally or externally:
 - externally:
`<!DOCTYPE java PUBLIC
"http://java.sun.com/DTDs/java1.4.1.dtd">`
 - internally:
`<DOCTYPE java [
<!ELEMENT java "version=" (#PCDATA)
"class=" (#PCDATA) (object)>
...`

Why use a DTD?

- Application independent way of sharing data
- Industries or trading parties can agree on a standard for interchanging data
- Verification that data received from trading parties is valid.

Body Components

- Components of the document
 - Tags
 - Case sensitive
 - `(<letter> | “_”)(<letter> | <digit> | “-” | “.”)*`
 - Every tag has an end tag `<put> ... </put>`
 - Nesting
 - Can have attributes

```
<java version="1.4.1" class="java.beans.XMLDecoder">  
...  
</java>
```

Body Components

- Components of the document
 - Entities refer to a (text) data item
 - start with & end with ;
 - eg. predefined entities: < & etc.
 - <!ENTITY COPYRIGHT “2003 UofA”>
 - referenced by: ©RIGHT;

Body Components

- Components of the document
 - Processing instructions
 - language processor specific
 - generic: `<?processor-instruction?>`
 - example: `<?style href="plain.xml"?>`
 - Comments
 - `<!-- This is a comment -->`

Well-Formed/Valid

- An XML document is well formed if it is syntactically correct
- An XML document is valid if
 - it is well-formed
 - its structure conforms to that described by its associated Document Type Definition

Document Type Definition

- Defines the structure of the document
 - Set of valid tags
 - Constraints on attribute values
 - Nesting of tags
 - Number of occurrence of tags
 - Entity definitions
- Akin to language grammar definition

Document Type Definition

- DTD is not expressed in XML
 - ugly!
 - Weak data typing
 - PCDATA – Parsed character data
 - CDATA – any character data
 - enumerations
 - ID, IDREF, NMTOKEN, NMTOKENS, ENTITY, ENTITIES, NOTATION
- XML Schema will replace DTDs

RCL target language DTD

```
<!ELEMENT RCL-TARGET (SYMBOL-TABLE, DATA-DEFS,  
  CODE-SEG)>  
<!ELEMENT SYMBOL-TABLE (SYMBOL)*>  
<!ELEMENT DATA-DEFS (SYMBOL)*>  
<!ELEMENT CODE-SEG (LABEL | INSTR)*>  
<!ELEMENT SYMBOL (#PCDATA)>  
<!ATTLIST SYMBOL Name ID #REQUIRED>  
<!ELEMENT INSTR (#PCDATA)>  
<!ATTLIST INSTR Op ID #REQUIRED>  
<!ELEMENT LABEL (#PCDATA)>  
<!ATTLIST LABEL Name ID #REQUIRED>
```

RCL target language sample

```
<?xml version='1.0'?>  
<!DOCTYPE RCL-TARGET SYSTEM  
  "http://berliner.cs.adelaide.edu.au/ccp/RCL-TARGET.dtd">  
<RCL-TARGET>  
  <SYMBOL-TABLE>  
    <SYMBOL Name="edge"> </SYMBOL>  
  </SYMBOL-TABLE>  
  <DATA-DEFS>  
    <SYMBOL Name="lengthSide"> </SYMBOL>  
    <SYMBOL Name="rotateSteps"> </SYMBOL>  
    <SYMBOL Name="side"> </SYMBOL>  
  </DATA-DEFS>  
  ...  
</RCL-TARGET>
```

RCL target language sample

```
<RCL-TARGET>  
...  
<CODE-SEG>  
  <LABEL Name="circle"> </LABEL>  
    <INSTR Op="INC"> 0 3 </INSTR>  
    <INSTR Op="LCB"> 0 127 </INSTR>  
    <INSTR Op="STO"> 0 0 </INSTR>  
    <INSTR Op="LCB"> 0 75 </INSTR>  
    <INSTR Op="STO"> 0 1 </INSTR>  
    <INSTR Op="LCB"> 0 1 </INSTR>  
    <INSTR Op="STO"> 0 2 </INSTR>  
  ...  
</CODE-SEG>  
</RCL-TARGET>
```

RCL target language sample

```
<RCL-TARGET>
...
<LABEL Name="loop"> </LABEL>
  <INSTR Op="LDV"> 0 2 </INSTR>
  <INSTR Op="LCB"> 0 16 </INSTR>
  <INSTR Op="OPR"> 0 15 </INSTR>
  <INSTR Op="JIF"> 0 done </INSTR>
  <INSTR Op="LDV"> 0 2 </INSTR>
  <INSTR Op="LCB"> 0 1 </INSTR>
  <INSTR Op="OPR"> 0 3 </INSTR>
  <INSTR Op="STO"> 0 2 </INSTR>
  <INSTR Op="MST"> 1 0 </INSTR>
  <INSTR Op="LDV"> 0 0 </INSTR>
  <INSTR Op="CAL"> 1 edge </INSTR>
  <INSTR Op="JMP"> 0 loop </INSTR>
...
</RCL-TARGET>
```

RCL target language sample

```
<RCL-TARGET>
...
<LABEL Name="done"> </LABEL>
  <INSTR Op="JMP"> 0 0 </INSTR>
<LABEL Name="edge"> </LABEL>
  <INSTR Op="LDV"> 0 0 </INSTR>
  <INSTR Op="OPR"> 0 23 </INSTR>
  <INSTR Op="OPR"> 0 34 </INSTR>
  <INSTR Op="LDV"> 1 1 </INSTR>
  <INSTR Op="OPR"> 0 23 </INSTR>
  <INSTR Op="OPR"> 0 2 </INSTR>
  <INSTR Op="OPR"> 0 34 </INSTR>
  <INSTR Op="OPR"> 0 0 </INSTR>
</CODE-SEG>
</RCL-TARGET>
```

XML Schema

- W3C recommendation 2001
- Standard and user defined data types
- Supports better type checking
- A schema is an XML document

- The future....

XML Parser

- Included in Microsoft Internet Explorer 5.0+
- Many parsers are available for many languages
 - including Java
 - Parser produces standardised AST called a Document Object Model (DOM)

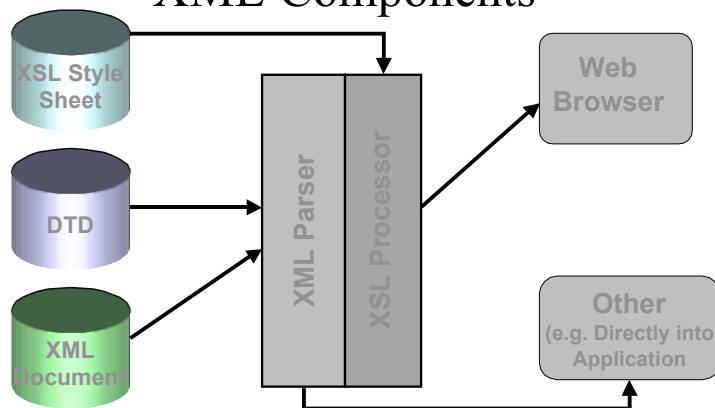
Document Object Model (DOM)

- Defines how a document can be accessed
- Represents a tree view of the XML document
- Node interface accesses elements in the XML tree

eXtensible Stylesheet Language (XSL)

- Method for transforming XML documents
- Method for formatting XML documents
- Browser or server based

XML Components



Using XML to Exchange Data

