University of Washington
Department of Computer Science & Engineering

Managing XML and Semistructured Data

Lecture 2: XML

Prof. Dan Suciu

Spring 2001

In this lecture

- XML syntax
- XML Query data model
- Comparison of XML with semistructured data

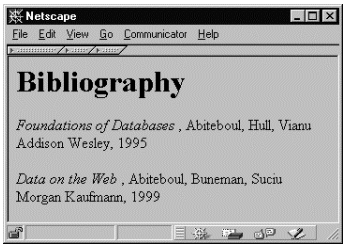
Papers:

- XML, Java, and the future of the Web by Jon Bosak, Sun Microsystems.
- W3C XML Query Data Model Mary Fernandez, Jonathan Robie.

XML

- a W3C standard to complement HTML
- origins: structured text SGML
- motivation:
  - HTML describes presentation
  - XML describes content
- HTML4.0∈ XML ⊂ SGML
- http://www.w3.org/TR/2000/REC-xml-20001006 (version 2, 10/2000)

From HTML to XML



HTML describes the presentation

HTML

```
<h1> Bibliography </h1>
<p> <i> Foundations of Databases </i>
      Abiteboul, Hull, Vianu
      <br> Addison Wesley, 1995
<p> <i> Data on the Web </i>
      Abiteoul, Buneman, Suciu
      <br> Morgan Kaufmann, 1999
```

XML

```
<bibliography>
    <book>   <title> Foundations… </title>
             <author> Abiteboul </author>
             <author> Hull </author>
             <author> Vianu </author>
             <publisher> Addison Wesley </publisher>
             <year> 1995 </year>
    </book>
    …
</bibliography>
```

XML describes the content

XML Terminology

- tags: book, title, author, …
- start tag: <book>, end tag: </book>
- elements: <book>…<book>,<author>…</author>
- elements are nested
- empty element: <red></red> abbrv. <red/>
- an XML document: single *root element*

well formed XML document: if it has matching tags

More XML: Attributes

```
<book price = "55" currency = "USD">
  <title> Foundations of Databases </title>
  <author> Abiteboul </author>
  …
  <year> 1995 </year>
</book>
```

attributes are alternative ways to represent data

More XML: Oids and References

```
<person id="o555">  <name> Jane </name> </person>

<person id="o456">  <name> Mary </name>
                    <children idref="o123 o555"/>
</person>

<person id="o123" mother="o456"><name>John</name>
</person>
```

oids and references in XML are just syntax

More XML: CDATA Section

- Syntax: <![CDATA[.....any text here...]]>
- Example:

```
<example>
  <![CDATA[ some text here </notAtag> <>]]>
</example>
```

More XML: Entity References

- Syntax: &entityname;
- Example:
  <element> this is less than < </element>
- Some entities:

<	<
>	>
&	&
'	'
"	"
&	Unicode char

More XML: Processing Instructions

- Syntax: <?target argument?>
- Example:
  <product> <name> Alarm Clock </name>
  <?ringBell 20?>
  <price> 19.99 </price>
  </product>
- What do they mean ?

More XML: Comments

- Syntax <!-- .... Comment text... -->
- Yes, they are part of the data model !!!

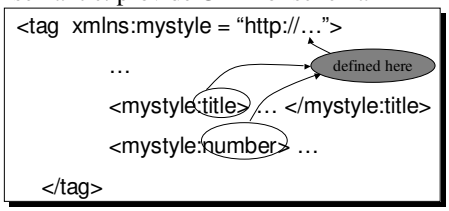
XML Namespaces

- http://www.w3.org/TR/REC-xml-names (1/99)
- name ::= [prefix:]localpart

```
<book xmlns:isbn="www.isbn-org.org/def">
    <title> … </title>
    <number> 15 </number>
    <isbn:number> …. </isbn:number>
</book>
```

XML Namespaces

- syntactic: <number> , <isbn:number>
- semantic: provide URL for schema



XML Data Model

Several competing models:

- Document Object Model (DOM):
  - http://www.w3.org/TR/2001/WD-DOM-Level-3-CMLS-20010209/ (2/2001)
  - class hierarchy (**node**, **element**, **attribute**,…)
  - objects have behavior
  - defines API to inspect/modify the document
- XSL data model
- Infoset
  - PSV (post schema validation)
- XML Query data model (next)

XML Query Data Model

- http://www.w3.org/TR/query-datamodel/ 2/2001
- Describes XML as a tree, specialized nodes
- Uses a functional-style notation (think ML)

XML Query Data Model

- Node ::= DocNode |
  ElemNode |
  ValueNode |
  AttrNode |
  NSNode |
  PINode |
  CommentNode |
  InfoItemNode |
  RefNode

XML Query Data Model

Element node (simplified definition):

- elemNode : (QNameValue,
  {AttrNode },
  [ElemNode | ValueNode])
  → ElemNode

- QNameValue = means "a tag name"
- {...} = means "set of..."
- [...] = means "list of ..."

XML Query Data Model

- Reads: "give me a tag, a set of attributes, a list of elements/values, and I will return an element"

XML Query Data Model

Example

```
<book price = "55"
      currency = "USD">
  <title> Foundations … </title>
  <author> Abiteboul </author>
  <author> Hull </author>
  <author> Vianu </author>
  <year> 1995 </year>
</book>
```

```
book1= elemNode(book,
  {price2, currency3},
  [title4,
   author5,
   author6,
   author7,
   year8])
price2 = attrNode(…)   /* next */
currency3 = attrNode(…)
title4 = elemNode(title, string9)
…
```

XML Query Data Model

Attribute node:

- attrNode : (QNameValue, ValueNode)
  → AttrNode

XML Query Data Model

Example

```
<book price = "55"
      currency = "USD">
  <title> Foundations … </title>
  <author> Abiteboul </author>
  <author> Hull </author>
  <author> Vianu </author>
  <year> 1995 </year>
</book>
```

```
price2 = attrNode(price,string10)
string10 = valueNode(…)  /* next */
currency3 = attrNode(currency,
                     string11)
string11 = valueNode(…)
```

XML Query Data Model

Value node:

- ValueNode = StringValue |
  BoolValue |
  FloatValue …

- stringValue : string → StringValue
- boolValue : boolean → BoolValue
- floatValue : float → FloatValue

XML Query Data Model

Example

```
<book price = "55"
      currency = "USD">
  <title> Foundations … </title>
  <author> Abiteboul </author>
  <author> Hull </author>
  <author> Vianu </author>
  <year> 1995 </year>
</book>
```

```
price2 = attrNode(price,string10)
string10 = valueNode(stringValue("55"))
currency3 = attrNode(currency, string11)
string11 = valueNode(stringValue("USD"))
title4 = elemNode(title, string9)
string9 =
valueNode(stringValue("Foundations…"))
```

XLink

- Generalizes HTML's href
- Many types: simple, extended, locator, ...
  - Discuss only simple links

```
<person xmlns:xlink="http:///.w3.org/1999/xlink"
        xlink:type="simple"                          } required attributes
        xlink:href="http://a.b.c/myhomepage.html"
        xlink:title="The Homepage"                   } optional attributes
        xlink:show="replace"
        xlink:actuate="onRequest">
  .....
</person>
```

XLink

- **show** attribute can be
  - "new"
  - "replace"
  - "embed"
  - "other"
- **actuate** attribute can be
  - "onLoad"
  - "onRequest"
  - "other"
  - "none"

XLink

- **href** attribute:
  - a URI or
  - an Xpointer (next)

XPointer

- An extension of XPath (next week)
- Usage:
  - **href=**"www.a.b.c/document.xml#xpointerExpr"
- An xpointer expression points to:
  - A point
  - A range

XPointer

- Pointing to a point (=XML element or character)
  - Full form: e.g. #xpointer(id("3652"))
  - Bar name: e.g. #3652
  - Child sequence: e.g. #xpointer(/1/3/2/5),
    #xpointer(/bib/book[3])
- Pointing to a range: e.g. #xpointer(id(3652 to 44))
- Most interesting examples use XPath

XML v.s. Semistructured Data

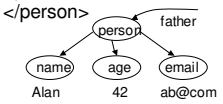
- both described best by a graph
- both are schema-less, self-describing

Similarities and Differences

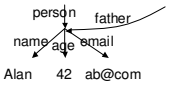
```
<person id="o123">
  <name> Alan </name>
  <age> 42 </age>
  <email> ab@com </email>
</person>
```

```
{ person: &o123
    { name: "Alan",
      age: 42,
      email: "ab@com" }
}
```

<person father="o123"> …
</person>



{ person: { father: &o123 …}
}



similar on trees, different on graphs

More Differences

- XML is ordered, ssd is not
- XML can mix text and elements:

```
<talk> Making Java easier to type and easier to type
      <speaker> Phil Wadler </speaker>
</talk>
```

- XML has lots of other stuff: entities, processing instructions, comments

Very important:
these differences make XML data management harder

Summary of Data Models

- semistructured data, XML
- data is self-describing, irregular
- schema embedded with the data