Foundations of XML Data Manipulation

Giorgio Ghelli

Course structure

• Data Model
• Query languages
• XPath
• Type systems, logics, tree automata
• Storing and querying
Structured data

<table>
<thead>
<tr>
<th>ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>Title</th>
<th>Birth Date</th>
<th>Hire Date</th>
<th>City</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Davolio</td>
<td>Nancy</td>
<td>Ms.</td>
<td>08-dic-1968</td>
<td>01-mag-1992</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>Dr.</td>
<td>19-feb-1952</td>
<td>14-ago-1992</td>
<td>Tacoma</td>
<td>WA</td>
</tr>
<tr>
<td>3</td>
<td>Leverling</td>
<td>Janet</td>
<td>Ms.</td>
<td>30-ago-1963</td>
<td>01-apr-1992</td>
<td>Kirkland</td>
<td>WA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>Mrs.</td>
<td>19-set-1958</td>
<td>03-mag-1993</td>
<td>Redmond</td>
<td>WA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>Mr.</td>
<td>04-mar-1955</td>
<td>17-ott-1993</td>
<td>London</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Suyama</td>
<td>Michael</td>
<td>Mr.</td>
<td>02-lug-1963</td>
<td>17-ott-1993</td>
<td>London</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>King</td>
<td>Robert</td>
<td>Mr.</td>
<td>29-mag-1960</td>
<td>02-gen-1994</td>
<td>London</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order ID</th>
<th>Customer</th>
<th>Emp ID</th>
<th>Order Date</th>
<th>Required Date</th>
<th>Shipped Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>10248</td>
<td>Wilman Kala</td>
<td>1</td>
<td>04-lug-1996</td>
<td>01-ago-1996</td>
<td>16-lug-1996</td>
</tr>
<tr>
<td>10249</td>
<td>Tradição Hiperm.</td>
<td>6</td>
<td>05-lug-1996</td>
<td>16-ago-1996</td>
<td>10-lug-1996</td>
</tr>
<tr>
<td>10250</td>
<td>Hanari Carnes</td>
<td>3</td>
<td>08-lug-1996</td>
<td>05-ago-1996</td>
<td>12-lug-1996</td>
</tr>
<tr>
<td>10253</td>
<td>Hanari Carnes</td>
<td>3</td>
<td>10-lug-1996</td>
<td>24-lug-1996</td>
<td></td>
</tr>
</tbody>
</table>

Unstructured data

- **Sample databases included with Access**
  - Microsoft Access provides sample databases that you can use while you're learning Access.
  - **Northwind Traders sample database**
  - The Northwind database and Access project (available from the Sample Databases command on the Help menu) contains the sales data for a fictitious company called Northwind Traders, which imports and exports specialty foods from around the world. By viewing the database objects included in the Northwind database. …
A syntax for SSD

\[ expr ::= value \mid oid value \mid oid \]
\[ value ::= atomic \mid \{ label : expr, \ldots, label : expr \} \]

\{ Bib: \&01 \{ paper: \&012 \{ …\},
  book: \&024 \{ …\},
  paper: \&029
    \{ author: \&052 "Abiteboul",
      author: \&096 \{ firstname: \&0243 "Victor",
        lastname: \&0206 “Vianu”),
      title: \&093 “Regular path queries”,
      references: \&024,
      page: \&025 \{ first: \&064 122, last: … \} \}
    \}
\} \]
Why SSD

• The origin:
  – Data integration
  – Documents
  – Scientific databases

• The interest:
  – Cannot be ignored
  – WWW and bioinformatics

The Data Model

• The information behind the syntax, i.e.:
  when two pieces of data really differ

• Some alternatives:
  – OEM: SSD as graphs modulo bisimulation
  – XML: ordered trees with node identity (and with pointers)
  – TQL: unordered trees
OEM with bisimulation

• Edge-labelled version
• Bisimulation: generalizes the notion of set equality to labelled graphs:
  – \{a: v, b: w\} = \{b: w, a: v\}
  – \{a: v, a: v, b: w\} = \{a: v, b: w, b: w\}
• Exists \( R \subseteq G \times G' \) such that:
  • \( n R m \) and \( n,l,n' \) in \( G \) \( \Rightarrow \) exists \( m,l,m' \) in \( G' \) with \( n' R m' \) and conversely
  • \( n R m \) and \( n \) leaf in \( G \leftrightarrow m \) leaf in \( G' \)

TQL data model

• Edge-labeled trees defined as multisets of label-tree pairs:
  – \( f ::= \emptyset \mid a[f] \mid f \mid f \)
  – \( f ::= \{\} \mid \{a:f\} \mid f \cup f \)
• Hence:
  – \( \{a: v, b: w\} = \{b: w, a: v\} \)
  – \( \{a: v, a: v, b: w\} \neq \{a: v, b: w, b: w\} \)
• The same syntax can be interpreted as node-labeled forests
Ordered children (as in XDM)

- Node-labeled ordered trees of elements
  - $item ::= <label> value </label> | leaf$
  - $value ::= item^*$

- Hence:
  - $\{a: v, b: w\} \neq \{b: w, a: v\}$
  - $\{a: v, a: v, b: w\} \neq \{a: v, b: w, b: w\}$

Binary representation
XML

• Simplification of SGML
• Designed to substitute HTML
• The standard for data exchange and web-services invocation
• Some W3C related standards:
  – XPath/XQuery
  – XML Infoset and XDM
  – XSLT
  – DTD, XSD
  – Many other things
XML for data exchange

```xml
<trader ID="T12">
  <name>Wilman Kala</name>
  <address><country>...</country>...</address>
  <orders>
    <order OID="O121">
      <date>1/3/2005</date>
      <item>...</item>
    </order>
    <order OID="O122">
      <item>...</item>
    </order>
  </orders>
</trader>

<trader ID="T13">
  <name>Hanari Cames</name>
  <address><city>...</city>...</address>
  <orders>
    <order OID="T131">
      <date>3/3/2005</date>
      <item>...</item>
    </order>
  </orders>
</trader>
```

XML as it was designed

```xml
<doc>
<title>Sample databases included with Access</title>
<subtitle>Microsoft Access provides sample databases.</subtitle>
<subtitle><link ref="./NT.mdb">Northwind Traders</link> database</subtitle>
<body>
  <para author="JDM" font="times">The Northwind database contains the sales data for a company called Northwind Traders, which imports and exports specialty foods from around the world. By viewing the <link ref="./NT.mdb">database objects</link> included in the Northwind database.</para>
  …</body>
</doc>
```
**XDM**

- A value is a sequence of nodes
- Parent axis: a node is a pair <tree, path in the tree>
  - \{a: {b: w}}/b \neq \{b: w\}
- Node identity: a store is a forest-structured graph <N,E>, and a node is an element of N
  - \{a: v\} \neq \{a: v\}

**Moreover**

- Six other types of nodes
- Unordered attributes
- ID – IDREFs to encode pointers
- Namespaces
- Type annotations
Conclusions

• We now know what SSD is
• Questions:
  – How do we describe its structure?
  – How do we manipulate it?
  – How do we store it?

Suggested readings

• Some papers are in the “query folder”
• klarlundSchweintick: general introduction to XML, DTD, XSD, XPath, XQuery, XSLT.
• AbiQuaMcH97: OEM and Lorel
• BunDavHil96: UnQL data model
• ColGheAl06: MicroXQuery data model
• Hidders: XML data model
• CarGhe03: TQL data model
• www.w3.org/TR/xpath-datamodel: XQuery/XPath data model (XDM)