

Foundations of XML Data Manipulation

Giorgio Ghelli

Course structure

- Data Model
- Query languages
- XPath
- Type systems, logics, tree automata
- Storing and querying

Structured data

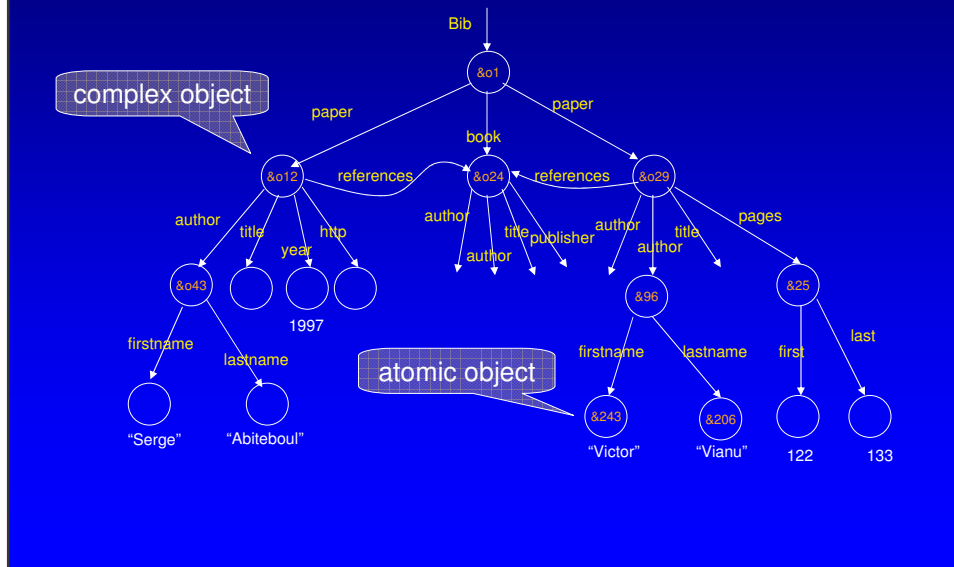
ID	Last Name	First Name	Title	Birth Date	Hire Date	City	Region
1	Davolio	Nancy	Ms.	08-dic-1968	01-mag-1992	Seattle	WA
2	Fuller	Andrew	Dr.	19-feb-1952	14-ago-1992	Tacoma	WA
3	Leverling	Janet	Ms.	30-ago-1963	01-apr-1992	Kirkland	WA
4	Peacock	Margaret	Mrs.	19-set-1958	03-mag-1993	Redmond	WA
5	Buchanan	Steven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

Order ID	Customer	Emp ID	Order Date	Required Date	Shipped Date
10248	Wilman Kala	1	04-lug-1996	01-ago-1996	16-lug-1996
10249	Tradição Hiperm.	6	05-lug-1996	16-ago-1996	10-lug-1996
10250	Hanari Carnes	3	08-lug-1996	05-ago-1996	12-lug-1996
10251	Victuailles en stock	3	08-lug-1996	05-ago-1996	15-lug-1996
10252	Suprêmes délices	2	09-lug-1996	06-ago-1996	11-lug-1996
10253	Hanari Carnes	3	10-lug-1996	24-lug-1996	
10254	Chop-suey Chinese	2	11-lug-1996	08-ago-1996	23-lug-1996

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. ...

Semistructured data



A syntax for SSD

```

expr ::= value | oid value | oid
value ::= atomic | { label : expr , ... , label : expr }

{ Bib: &o1 { paper: &o12 {...},
             book: &o24 {...},
             paper: &o29
               { author: &o52 "Abiteboul",
                 author: &o96 { firstname: &o243 "Victor",
                               lastname: &o206 "Vianu"},
                 title: &o93 "Regular path queries",
                 references: &o24,
                 page: &o25 { first: &o64 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'

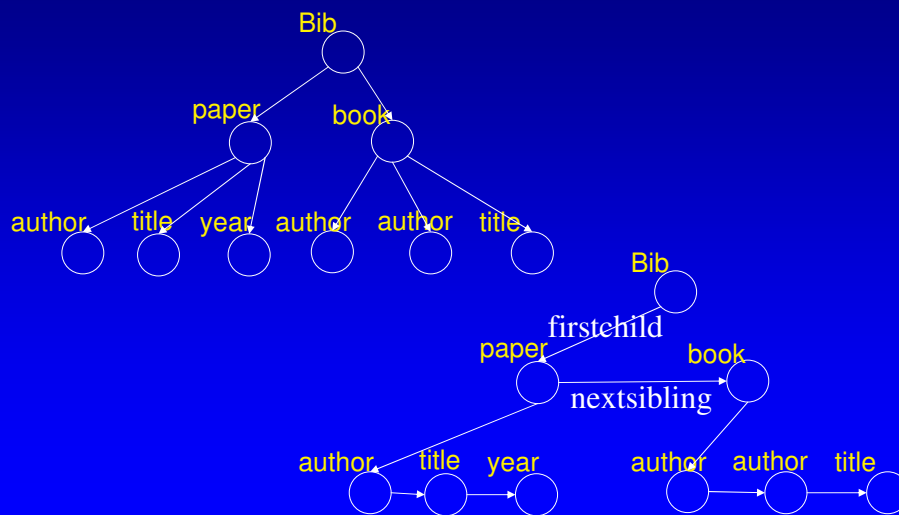
TQL data model

- 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
 - $item ::= \langle label \rangle value \langle /label \rangle \mid leaf$
 - $value ::= item^*$
- 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

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

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

```
<doc><title>Sample databases included with Access</title>
<subtitle>Microsoft Access provides sample
databases.</subtitle>
<subtitle> <link ref= "./NT.mdb">Northwind Traders
database </link> </subtitle>
<body>
<para author= "JDM" font="times">The Northwind
database contains the sales data for a company called
<emph>Northwind Traders</emph>, 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 $\langle \text{tree}, \text{path in the tree} \rangle$
 - $\{a:\{b: w\}\}/b \neq \{b: w\}$
- Node identity: a store is a forest-structured graph $\langle N, E \rangle$, 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)