Conceptual Modeling for XML Data
Tok Wang Ling
National University of Singapore

Outlines

• Why do we need conceptual modeling?
• What are the important semantic information to be captured?
• Uses of conceptual model for some XML research topics

Motivation: Why do we need to have a conceptual model to represent XML Data?

Motivation (cont.)

Q: What are the important semantic information and constraints cannot be captured by the DTD and Dataguide?

• What are the object classes? department, course, student?
• Attributes of object classes?
• Identifiers of object classes?
• What are the relationship types defined among object classes?
• e.g. Relationship types among department, course, student?
• What is “grade”? Object class? Attribute of student?
• Are there redundancies?

Semantic Information to be captured by an XML conceptual model

• Object class
  – attributes of object class
  – ordering on object class

• Relationship Type
  – Represent hierarchical structure
  – degree of n-ary relationship type
  – participation constraints of object classes in relationship type
  – attributes of relationship type
  – disjointive relationship type
  – recursive relationship type

Reference
Semantic Information to be captured by an XML conceptual model (cont.)

- Attribute
  - key attribute / identifier
  - composite attribute
  - disjunctive attribute
  - attributes with unknown structure
  - fixed and default values of attribute
  - derived attribute

- Functional dependencies and other constraints
- Inheritance hierarchy (class hierarchy)
- Semi-structured data instance representation

Q: What are the semantic information cannot be represented by Dataguide, DTD, XML Schema?

- Attribute or object class
- Degree of relationship type
- Attribute of object class or relationship type
- Class hierarchy
- Functional dependency
- ...

A solution: ORA-SS, an object-relationship attribute model for semi-structured data.

Figure 2: ORA-SS instance diagram

Figure 3: ORA-SS schema diagram

The data model of ORA-SS - Relationship Type

- attributes of relationship type
- degree of n-ary relationship type
- participation constraints of objects in relationship type
- disjunctive relationship type
- recursive relationship type

Figure 5: Representing binary relationship type

The data model of ORA-SS - Attribute

- key attribute
- composite attribute
- disjunctive attribute
- attribute with unknown structure
- fixed and default values of attribute
- derived attribute

Figure 7: Object classes with relationship type and attributes in an ORA-SS schema diagram
Uses of the Conceptual model for XML research

- Normal form XML schema
  - remove redundant data
  - resolve multiple inheritance conflicts
- Storage structure for XML databases
  - use Object Relational Model
- XML Views
  - derived information from references and class hierarchy
  - defining views
  - materialized view maintenance
  - view updates
- Integration of XML documents
- Evaluating XML queries on XML databases

Research Topics using ORA-SS Model

Normal Form XML Schema

- Schema may have a lot of redundant data
- Update anomalies
- Normal Form schema is needed

Storage Structure for XML Databases

- Main Rules
  - Each object class together with its attributes form a nested relation (object relation)
  - Each relationship type together with its attributes form a nested relation (relationship relation)
- Nested relations can be handled by Object Relational model, e.g. ORACLE 8i.

XML Views

- What information can be directly derived from references and class hierarchy

Referencing an object class in an ORA-SS schema diagram
Research Topics using ORA-SS Model

XML Views (cont.)
- Valid views of an ORA-SS schema
- Operations: selection, projection, join, up/down

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Part</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>price</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proj</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qty</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Supplier</th>
<th>Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>price</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>total_qty</td>
</tr>
</tbody>
</table>

View 1 View 2 View 3

Conclusion
- A good conceptual model is needed for XML database applications:
  - normal form schema
  - storage structure
  - view design and view updates
  - ...