XML-to-SQL Query Translation Literature

The State of the Art and Open Problems

Focus of Published Solutions

<table>
<thead>
<tr>
<th>Focus of Published Solutions</th>
<th>Simple Queries (path expressions)</th>
<th>Complex Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML Publishing</td>
<td>Tree Schema</td>
<td>Recursive Schema</td>
</tr>
<tr>
<td></td>
<td>XP</td>
<td>lot</td>
</tr>
<tr>
<td></td>
<td>XS/SO</td>
<td>lot</td>
</tr>
<tr>
<td></td>
<td>XS/SB</td>
<td>lot</td>
</tr>
<tr>
<td></td>
<td>XP</td>
<td>lot</td>
</tr>
<tr>
<td></td>
<td>XS/SO</td>
<td>some</td>
</tr>
<tr>
<td></td>
<td>XS/SB</td>
<td>none</td>
</tr>
</tbody>
</table>

Surveyed Techniques

<table>
<thead>
<tr>
<th>TECHNIQUE</th>
<th>SCENARIO</th>
<th>REPRESENTATION</th>
<th>CLASS OF XML QUERIES HANDLED</th>
<th>CLASS OF XML QUERIES SUPPORTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>XP/GAV</td>
<td></td>
<td>Bounded Intervals</td>
<td>all SS, QT XS/SB</td>
<td>Dynamic Intervals [7]</td>
</tr>
<tr>
<td>XP/LAV</td>
<td></td>
<td>Bounded Intervals</td>
<td>all SS, QT XS/SB</td>
<td>Dynamic Intervals [7]</td>
</tr>
<tr>
<td>XS/SO</td>
<td></td>
<td>Path expressions</td>
<td>all SS, QT XS/SB</td>
<td>Query Translation [35]</td>
</tr>
<tr>
<td>XS/SB</td>
<td></td>
<td>Path expressions</td>
<td>all SS, QT XS/SB</td>
<td>XML Storage, schema-oblivious</td>
</tr>
<tr>
<td>QT</td>
<td></td>
<td>Path expressions</td>
<td>all SS, QT XS/SB</td>
<td>Query Translation [35]</td>
</tr>
<tr>
<td>SS</td>
<td></td>
<td>Storage scheme</td>
<td>all SS, QT XS/SB</td>
<td>View Definition [7]</td>
</tr>
</tbody>
</table>

XML Publishing

Tasks
- Defining an XML view of relational data.
- Materializing the XML view.
- Evaluating an XML query

Xperanto

- The view definition languages permit definition of tree XML views over the relational data.
- XML view is materialized by pushing down a single "outer union" query into the relational engine.
- The XQuery query is converted to an XQGM representation and composed with the view definition.

Xperanto-continued

XML Publishing:

Global as view vs. local as view

View Definition

Source s1 specification... Source s2 specification... Source sn specification...
Rolex

- the view definition languages permit definition of tree XML views over the relational data
- a view composition algorithm for composing an XSLT stylesheet with an XML view definition to produce a new XML view
- world view is changed so that a relational system provides a virtual DOM interface to the application.

Oracle XML DB

- an annotated XSD XML schema is used to define the XML view
- supports recursive XML views
- implements the majority of the operators incorporated into the forthcoming SQL/XML standard

Microsoft SQL Server 2000/SQLXML

- an annotated XSD XML schema is used to define the XML view
- support for a limited number of depths of recursion using the max-depth annotation.
- supports the evaluation of XPath queries over the annotated XML Schema.
- The XPath query together with the annotated schema is translated into a FOR XML explicit query

IBM DB2 XML Extender

- a Document Access Definition (DAD) is used to define a non-recursive XML view.
- user-defined functions (UDFs) are provided to store and retrieve XML documents in XML columns, as well as to extract XML element or attribute values.

Agora

- uses the local-as-view approach (LAV), where the local source's schema are described as views over the global schema.
- describe a generic, virtual relational schema closely modeling the generic structure of an XML document.
- relational schema is then defined as views over this generic, virtual schema.
Agora

- Algorithm is two main steps:
  - translating XQuery expressions into SQL on the generic view
  - translating generic SQL into specific SQL

Open Problems

- Of the systems presented, only Oracle XML DB supports recursive views at all, and does not support the descendant (/) access.
- Most of the approaches do not support XQuery, or if they do, it’s not complete
- performance analysis and the advantages of LAV vs GAV have not been explored

Recursive XML View Schema and Linear Recursion in SQL

- The few papers that touched upon recursion claimed that the linear recursion supported by SQL was not sufficient for the non-linear recursion in DTDs
- the authors provide a counterexample that re-opens the question

Recursive XML View Schema and Linear Recursion in SQL

- non-linear recursive DTD: part -> pname, part*
  - underlying relational schema has two relations, Part and Subpart with the column: (partid, pname) and (partid, subpartid)
  - WITH RECURSIVE AllParts(partid, pname, rtolpath) as (select partid, pname, ''
    from Part(partid, pname)
    union all
    select P.partid, P.pname, rtolpath + A.partid
    from AllPartsA, Subpart S, Part P
    where S.partid = A.partid and S.subpartid = P.partid)
  - select from AllParts
- the core SQL query executes the following linear-recursive Datalog program. AllParts(partid, pname) <- Part(partid, pname)
  AllParts(subpartid, subpname) <- AllParts(partid, pname) Subpart(partid, subpartid)

Schema-Oblivious XML Storage

- goal is to find a relational schema that works for storing XML documents independent of the presence or absence of a schema.
- Relational schema design: which generic relational schema for XML should be used?
- Query translation algorithms: given a decision for the relational schema, how do we translate from XML queries to SQL queries.

Schema Oblivious Storage

- STORED: generates a mapping given a semi-structured database instance. happily stores non-conforming documents into overflow tables
- Edge Approach: XML documents are viewed as graphs, with each edge represented as a tuple in a single table.
- path based approach: all elements of a path are stored in a single relation
- IBM and Oracle: entire XML document is stored as a CLOB data type, bypassing SQL.

Simple Path Expression Queries

- interval-based: evaluating simple path expressions entails performing a range join for each step of the path expression.
- id-based solutions: each parent-child( ) step translates into an equijoin
- path-based solutions: the path id can be used to avoid performing one join per step of the path expression.
Complex XQuery queries

- All approaches save one do not address queries more complex than relatively simple path expressions.
- The paper that does address complex query statements requires the addition of operators to SQL and modification of the relational engine.
- So this area is still open

Schema-based Storage

approaches to storing XML in relational systems that make use of a schema for the XML data in order to choose a good relational schema.
- given an XML schema (or DTD), how should we choose a good relational schema and XML-to-relational mapping.
- having chosen an XML-to-relational mapping, how should we translate XML queries into SQL.