Introduction

- Dramatic increase of web data (HTML, XML)
- XML gains popularity as a candidate
- Needs to manage XML data arise
- Two approaches
  - Building customized DB
  - Using RDB as underlying engine
- **XPRESS** (XML Processing and Relaxation in rElational Storge System) project @ UCLA/CSD
XPRESS: Architecture

<table>
<thead>
<tr>
<th>Query</th>
<th>Database</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>MQL</td>
<td>XML doc</td>
<td></td>
</tr>
<tr>
<td>SQL</td>
<td>RDB</td>
<td>TAH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>query processing query relaxation</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

October 11 ER 2000

XPRESS: Pros & Cons

- **Pros**
  - Present market is dominated by RDB; impractical to abandon RDB to adopt XML
  - Mature RDB techniques (OLAP, Data Warehousing, etc) can be reused
  - Support various types of input/output

- **Cons**
  - High conversion costs btw. XML and Relational models
  - Incorrect or incomplete conversion
Related Works

- STORED @ AT&T Labs [SIGMOD98]
- Inlining @ U. Wisconsin [VLDB99]
- INRIA [IEEE Eng. Bulletin 00]

- XML-DBMS, IBM DB2 XML Extender, Informix Web DataBlade, Oracle 8i, Sybase Adaptive Server Enterprise 12.0, …

- Existing mapping methods
  - Structure-oriented; ignoring constraints
  - Require programming/human intervention

Background: XML & DTD

- Not a single, predefined markup language (e.g. HTML): it's a meta-language by W3C
  - Start and end tags (<name>...</name>)

- Two XML schema languages from W3C
  - DTD (Document Type Definition), XML-Schema

- DTD: formal description about the *structures and constraints* of the XML document
  - XML vs. DTD =~ SQL vs. DDL
  - Element vs. attribute (DTD) =~ table vs. column (RDB) =~ class vs. attribute (OO)
Document Type Definition (DTD)

- **Element** (ordered) & **attribute** (unordered)

```
<!ELEMENT elm-name elm-type>
<!ATTLIST elm-name att-name att-type att-option>
```

- **Element type** `<elm-type>`
  - String type (#PCDATA)
  - 0 or 1 (?), 0 or more (*), 1 or many (+)
  - Sequential (,), choice (|)

- **Attribute type** `<att-type>`
  - String type (CDATA)
  - identifier (ID), foreign key (IDREF, IDREFS)

- **Attribute option** `<att-option>`
  - #IMPLIED, #REQUIRED

Example

- A **paper** element has a unique **id**, one **title**, one or many **authors**, and zero or many referenced **papers**:

```
<!ELEMENT paper (title, author+)>
<!ATTLIST paper pid ID #REQUIRED
   ref IDREFS #IMPLIED>

<paper pid="p10" ref="p1 p3">
  <title>…</title>
  <author>A</author><author>B</author>
</paper>
```
XML to Relational Mapping

Query | Database | Result
--- | --- | ---
XML Query | XML doc | Query processing
SQL | RDB | Result

Difficulties

- No 1-to-1 mapping
- Set \((a^*, (b+|c?))\), Recursion

```xml
<list name="A"><item>1</item><item>2</item> </list>
<list name="B"><item>3</item><item>4</item> </list>
```

- Fragmentation & inlining

```xml
<!ELEMENT A (B|C)>
1. Fragmented: 3 tables A, B, C
2. Inlined: 1 table D
```

<table>
<thead>
<tr>
<th>list</th>
<th>name</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1,2</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>3,4</td>
<td></td>
</tr>
</tbody>
</table>
CPI Algorithm

- Uses structure-oriented translation algorithm (e.g., hybrid inlining algorithm [VLDB 99]) as basis
- Preserves constraints during the translation
- Convert DTD to a digraph with annotated edge types
- Identify top nodes: 1) source, 2) child of *,+ node, 3) recursive node with indegree>1, ...
- Map top nodes $T$ to table $t$ (to avoid non-1NF); map leaf nodes reachable from $T$ to column $c$ of $t$ via inlining unless $T$ is another top node
- New columns for bookkeeping
  - $fk_{\text{key}}$, $parent_{\text{elm}}$, $root_{\text{elm}}$, ordinal, ...

Constraints in DTD

- Domain constraint
  ```xml
  <!ATTLIST author gender (M|F) #REQUIRED
  married (Y|N) #IMPLIED>
  ```

- Cardinality constraint
  ```xml
  <!ELEMENT book (title,author+,ref*,price?)>
  ```
  - 1-to-{0,1}: at most 1 (price)
  - 1-to-{1}: must be 1 (title)
  - 1-to-{0,...}: any occurrence (ref)
  - 1-to-{1,...}: at least 1 (author)
Constraints in DTD (cont.)

- Inclusion Dependency (ID)

```
<!ELEMENT person  (name,(email|phone)?)>
<!ATTLIST person  id  ID  #REQUIRED>
<!ELEMENT contact EMPTY>
<!ATTLIST contact  aid IDREF  #REQUIRED>
<!ELEMENT editor  (person*)>
<!ATTLIST editor  eid IDREFS  #IMPLIED>
```

aid \(\subseteq\) id, eid \(\subseteq\) id

Constraints in DTD (cont.)

- Equality-Generating Dependency (EGD)
  - Values in one columns require values in other columns be equal
  - In XML, EGD is disguised as “Singleton” property
  - When an element instance \(x\) of type \(X\) satisfies the singleton property towards its sub-element instances \(y_1\) and \(y_2\) of type \(Y\), \(y_1\) and \(y_2\) must be equal
  - 1-to-\{0,1\} and 1-to-\{1\} cardinality cases

\[X \rightarrow X.Y\]
Constraints in DTD (cont.)

- Tuple-Generating Dependency (TGD)
  - Require some tuples of a certain form be present
  - In XML, TGD is disguised as "Not-Nullness" property
  - Child property ($P \rightarrow C$): Every element of type $P$ must have at least one child element of type $C$
    - 1-to-{$1$} and 1-to-{$1,...$} cardinality cases
  - Parent property ($C \rightarrow P$): Every element of type $C$ must have a parent element of type $P$
    - Only can be enforced by semantic knowledge since any proper element can be a root w/o parent

---

Conference.dtd

```xml
<!ELEMENT conf (title, date, editor?, paper*)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT date EMPTY>
<!ATTLIST date year CDATA #REQUIRED mon CDATA #REQUIRED day CDATA #IMPLIED>
<!ELEMENT editor (person*)>
<!ATTLIST editor eids IDREFS #IMPLIED>
<!ELEMENT paper (title, contact?, author, cite?)>
<!ATTLIST paper id ID #REQUIRED>
<!ELEMENT contact EMPTY>
<!ATTLIST contact aid IDREF #REQUIRED>
<!ELEMENT author (person*)>
<!ATTLIST author id ID #REQUIRED>
<!ELEMENT person (name, (email|phone)?)>
<!ATTLIST person id ID #REQUIRED>
<!ELEMENT name EMPTY>
<!ATTLIST name fn CDATA #IMPLIED ln CDATA #REQUIRED>
<!ELEMENT email (#PCDATA)>
<!ELEMENT cite (paper*)>
<!ATTLIST cite id ID #REQUIRED format (ACM|IEEE) #IMPLIED>
```
<conf id="er99">
  <title>Int'l Conference on Conceptual Modeling (ER)</title>
  <date> <year>1999</year> <mon>May</mon> <day>20</day> </date>
  <editor eids="sheth bossy">
    <person id="klavans">
      <name fn="Judith" ln="Klavans"/><email>kla@columbia.edu</email>
    </person>
  </editor>
  <paper id="p1">
    <title>Indexing Model for Structured...</title>
    <contact aid="dao"/>
    <author><person id="dao"><name fn="Tuong" ln="Dao"/></person></author>
  </paper>
  <paper id="p2">
    <title>Logical Information Modeling of...</title>
    <contact aid="shah"/>
    <author>
      <person id="shah"><name fn="Kshitij" ln="Shah"/></person>
      <person id="sheth">
        <name fn="Amit" ln="Sheth"/><email>amit@cs.uga.edu</email>
      </person>
    </author>
  </paper>
  <cite id="c100" format="ACM">
    <paper id="p3">
      <title>Making Sense of Scientific Info...</title>
      <author>
        <person id="bossy">
          <name fn="Marcia" ln="Bossy"/><phone>391.4337</phone>
        </person>
      </author>
    </paper>
  </cite>
  <paper id="p7">
    <title>Constraints-preserving Transformation...</title>
    <contact aid="lee"/>
    <author>
      <person id="lee">
        <name fn="Dongwon" ln="Lee"/><email>dongwon@cs.ucla.edu</email>
      </person>
    </author>
    <cite id="c200" format="IEEE"/>
  </paper>
</conf>
Annotated DTD Graph

CPI Steps
CPI Steps (cont.)

CPI Steps (cont.)
CPI Steps (cont.)

- id, title cannot be NULL
- cite_id, contact_aid, cite format can be NULL
- fk_conf is a FK to conf
- fk_cite is included in cite_id (i.e., fk_cite ccite_id)
- id is a PK
- cite_id is UNIQUE

---

Paper Table after CPI

<table>
<thead>
<tr>
<th>id</th>
<th>root_elm</th>
<th>parent_elm</th>
<th>fk_conf</th>
<th>fk_cite</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>conf</td>
<td>conf</td>
<td>er99</td>
<td></td>
</tr>
<tr>
<td>p2</td>
<td>conf</td>
<td>conf</td>
<td>er99</td>
<td></td>
</tr>
<tr>
<td>p3</td>
<td>conf</td>
<td>cite</td>
<td></td>
<td>c100</td>
</tr>
<tr>
<td>p7</td>
<td>paper</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>title</th>
<th>contact_aid</th>
<th>cite_id</th>
<th>cite_format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing...</td>
<td>dao</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical...</td>
<td>shah</td>
<td>c100</td>
<td>ACM</td>
</tr>
<tr>
<td>Making...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Constraints...</td>
<td>lee</td>
<td>c200</td>
<td>IEEE</td>
</tr>
</tbody>
</table>
Schema after CPI

CREATE TABLE paper (  
  id              NUMBER NOT NULL,  
  title           VARCHAR(50) NOT NULL,  
  contact_aid     NUMBER,  
  cite_id         NUMBER,  
  cite_format     VARCHAR(50) CHECK (VALUE IN ("ACM", "IEEE")),  
  root_elm        VARCHAR(20) NOT NULL,  
  parent_elm      VARCHAR(20),  
  fk_cite         VARCHAR(20) CHECK (fk_cite IN (SELECT cite_id FROM paper)),  
  fk_conf         VARCHAR(20),  
  PRIMARY KEY (id),  
  UNIQUE (cite_id),  
  FOREIGN KEY (fk_conf) REFERENCES conf(id),  
  FOREIGN KEY (contact_aid) REFERENCES person(id)
);  

Experimentation

- DTD from OASIS site
  - 12 DTDs in different domain: play (Shakespeare), MusicML (music), Xbel (bookmark), tstmt (religious)...

- DBLP data
  - http://www.cobase.cs.ucla.edu/pub/dblp/
  - DB-related conferences (79,547), journals (60,963) & books (1,045) XML files (60MB)
  - Each XML file size < 5K bytes
  - Mapped to 10 tables, total 509,392 tuples
Summary

- CPI (Constraints-preserving Inlining) algorithm is presented
  - Work with structure-oriented mapping algorithms
  - Find and preserve constraints during the mapping

http://www.cobase.cs.ucla.edu/projects/xpress/