Colorful XML

Divesh Srivastava
AT&T Labs-Research
http://www.research.att.com/~divesh/
Thesis

- XML originally designed for document markup
  - Elements with attributes, ordered nested sub-elements

- XML now also used for heterogeneous data modeling
  - Flexibility: repeated elements, optional elements, hierarchy
  - Data model underlies XPath, XQuery, XSchema

- Issue: XML data model supports a single hierarchy
  - Schema design: update anomalies vs concise queries

- Solution: Use multiple colored trees (MCT)!
Road Map

- Motivation: weaknesses in XML data modeling
- MCT (colorful XML) data model
- MCT schema design: desiderata, ER → MCT
- Experiments
Motivation: TPC-W ER Diagram
Motivation: Shallow XML Trees

- **Bad**: reliance on value-based joins, not XPath axes
  - **Q1**: Orders by customers with US addresses
    
    for $a$ in //order, $b$ in //customer, $c$ in address, $d$ in //country[@name='US']
    where $c$/in/@country_idref = $d$/@id and $b$/has/@address_idref = $c$/@id and $b$/@id = $a$/make/@customer_idref
    return $a$

- **Good**: no redundancy, no update anomalies
Motivation: Medium XML Trees

- Good: concise expression of some queries using XPath
  - Q₁: Orders by customers with US addresses
    
    ```xml
    for $a$ in //country[@name='US']/order return $a$
    ```

- Good: no redundancy, no update anomalies
Motivation: Medium XML Trees

- **Bad:** verbose expression of other queries using XPath
  - \(Q_2\): Orders by customers with US ship addresses
    
    ```xml
    for $a$ in //order, $b$ in //country[@name='US']/address
    where $b/@id = $a/shipping/ship_address_idref
    return $a
    ```

- **Good:** no redundancy, no update anomalies
Motivation: Deep XML Trees

- Good: concise expression of all queries using XPath
  - Q₁: Orders by customers with US addresses
    \[
    \text{for } $a \text{ in } //\text{customer}[\text{has} //\text{country}[@\text{name} = 'US'] //\text{order} \text{ return } $a
    \]
  - Q₂: Orders by customers with US ship addresses
    \[
    \text{for } $a \text{ in } //\text{order}[\text{shipping} //\text{country}[@\text{name} = 'US'] \text{ return } $a
    \]
- Bad: lots of redundancy, potential update anomalies
Motivation: Problem Statement

- Opposing goals in XML schema design
  - Update anomaly avoidance
  - Query expression ease, query evaluation efficiency

- Objective: given ER diagram, design XML-like schema where
  - Update anomalies can be avoided
  - All associations in the ER diagram can be expressed using structural predicates, permitting efficient evaluation

- Solution: multiple colored trees
Road Map

- Motivation: weaknesses in XML data modeling
- MCT (colorful XML) data model
- MCT schema design: desiderata, ER → MCT
- Experiments
MCT: Data Model

- Model: finite set of colors, each an (ordered) tree structure
  - Conservative extension of the XML data model

- Query: XPath/XQuery syntax extended with color specifications
Q₃: Orders by customers with US ship and bill addresses

- `for $a in /{red} country[@name = 'US']//{red} order [{pink} ancestor::country [@name = 'US'] return $a`

Illustrates colored XPath matching, color crossing
MCT: Intuition

- Application data may admit multiple hierarchical organizations
  - Orders by customer country/address, or by shipping country/address, or by billing country/address

- XML data model forces us to choose “one” hierarchy
  - Other hierarchical organizations encoded as values
  - Not very intuitive for complex data

- Multiple colors → support multiple hierarchies simultaneously
  - Akin to multiple dimension hierarchies in data warehouses
MCT: Philosophy

- What are colors?
  - Tuple = single color twig
  - Relation = single MCT color

- Colors are not views
  - View: dependency with base data, update ambiguities
  - MCT: colors are independent, no update ambiguities

- Multiple colored trees are not arbitrary graphs
  - Trees are simpler than arbitrary graphs
  - Key difference with OO and semi-structured models
Road Map

- Motivation: weaknesses in XML data modeling
- MCT (colorful XML) data model
- MCT schema design: desiderata, ER → MCT
- Experiments
Schema Design: Query Goals

- Definitions
  - Simple association: an association between 2 nodes in the ER diagram that is 1:n
  - Association: any connected sub-graph of the transitive closure of the ER diagram

- Desiderata: ease of query expression, efficiency of evaluation
  - Direct recoverability (DR): simple associations can be specified as a single ancestor-descendant XPath axis step
  - Association recoverability (AR): every association should be recoverable using structural navigation
Schema Design: Update Goals

- Desiderata: avoidance of update anomalies
  - Node normal form (NN): if no instance of a node in the ER diagram is present more than once in each color
  - Edge normal form (EN): if no edge in ER diagram is present in more than one color
XML Schema Design: Tradeoffs

- NN satisfied, AR not satisfied
- NN not satisfied, AR satisfied
ER Diagram → Single Color MCT

- Start from ER diagram
  - Give directions to edges based on cardinality constraints
  - Traverse graph covering all nodes, edges to obtain forest
ER Diagram → Single Color MCT

- Theorem: NN and AR are both satisfied iff
  - ER graph is a forest
  - ER graph does not contain k-ary (k > 2), m:n relationships
  - No entity in ER graph is on the “many” side of more than one 1:n relationship

- ER diagrams that satisfy above theorem are very limited!
ER Diagram → Single Color MCT

- NN satisfied, AR not satisfied
- NN not satisfied, AR satisfied
ER Diagram → MCT

- Edge Normal Form Schema: satisfy NN, AR, EN, minimal colors
  - Choose new schema node as candidate root for a color
  - Traverse graph until no edge can be added to maintain NN
  - Don’t traverse same edge in multiple colors → satisfy EN
Direct Recoverable (DR) Schema: satisfy NN, AR, DR
- Traverse graph until no edge can be added to maintain NN
- Add new colors until all simple (1:n) associations covered
Minimal Color Maximal Recoverable (MCMR): satisfy NN, AR
- Start from EN schema, add nodes, edges, satisfying NN
- Intuition: color minimality of EN, more DR than EN
Road Map

- Motivation: weaknesses in XML data modeling
- MCT (colorful XML) data model
- MCT schema design: desiderata, ER → MCT
- Experiments
Experiments: Goals and Strategies

- Quantify efficiency in query/update processing
  - Actual query evaluation with TPC-W benchmark
  - Running on TIMBER on top of SHORE
  - 7 schemas, each with 13 queries + 3 updates

- Demonstrate ease of query specification
  - Analysis of query metrics from collection of ER diagrams
  - Emulating XMark to generate query workload
  - 6 schemas, each with 20 queries + 8 updates from UpdateX
TPC-W: Storage

- Schema design alternatives evaluated
  - DEEP (AR, not NN), SHALLOW (NN, not AR)
  - EN (NN, AR, EN), DR (NN, AR, DR), MCMR (NN, AR)
  - AF (nested SHALLOW), UNDR (DR, some unnormalization)
TPC-W: Query Performance

- Schema design alternatives evaluated
  - DEEP (AR, not NN), SHALLOW (NN, not AR)
  - EN (NN, AR, EN), DR (NN, AR, DR), MCMR (NN, AR)
  - AF (nested SHALLOW), UNDR (DR, some unnormalization)
TPC-W: Query Characteristics

- Number of value joins + color crossings
- Value join > color crossing >> structural join
TPC-W: Query Characteristics

- Number of duplicate eliminations, duplicate updates, group bys
- Queries and updates expensive due to redundancy
ER Collection: Query Characteristics

- Number of structural joins

![Bar chart showing geometric mean for ER Collection queries with bars for different data sets and query types.]
ER Collection: Query Characteristics

- Number of value joins + color crossings
  - Value join > color crossing >> structural join
ER Collection: Query Characteristics

- Number of duplicate eliminations, duplicate updates, group bys
- Queries and updates expensive due to redundancy
Experimental Evaluation: Summary

- DEEP: best if no redundancy, worst if considerable redundancy

- SHALLOW and AF: bad on average because of lack of AR, DR
  - Good for updates because of no redundancy

- EN and MCMR: no redundancy, better than SHALLOW, AF

- DR: fastest on average with slightly more space than MCMR

- Recommendation: MCMR
Conclusions

- Demonstrated inadequacy of XML for data modeling
  - Hierarchies are important

- MCT: conservative extension of XML, best of both worlds
  - Update anomaly avoidance
  - Ease of query expression, efficiency of query evaluation
  - Practical: implemented in TIMBER

- Open: how to derive good MCT schema from an XML schema?
What Role can XML Play?

- Physical model: TIMBER, …
  - Actually stored representations, modified

- Logical model: MCT
  - Query and update abstractions

- Exchange model: XML
  - Good for serialization
  - Cost-based optimal serialization result in SIGMOD’04 paper
Acknowledgements

Colleagues
- H.V. Jagadish, Laks V.S. Lakshmanan, Monica Scannapieco, Nuwee Wiwatwattana

Papers