Symmetrically Exploiting XML

Shuohao Zhang and Curtis Dyreson

School of E.E. and Computer Science
Washington State University
Pullman, Washington, USA

The 15th International World Wide Web Conference
May 2006
Edinburgh, Scotland
1970’s Database Controversy

- Hierarchical model vs. relational model
- Codd: symmetric exploitation of data

- part/project works on some, but not all

- Path expressions are asymmetric
- Currently, all XML query languages use path expressions
Querying Data with Path Expressions

- Task
  - Find books by E. F. Codd
- XQuery
  - `return doc("author.xml")//author[name= 'E. F. Codd']/book`
Same task
- Find books by E. F. Codd

Need different XQuery
- \( \text{return } \text{doc("book.xml")//book[author/name='E. F. Codd']} \)
Goal

• Make same query work on different structures

• Useful when there is
  ■ lack of schema knowledge
  ■ heterogeneous data
  ■ irregular data
  ■ schema evolution

• Factor off problem of different label sets, others are working on it
Existing Axes are Directional
Proposal: A Non-directional Axis

Symmetrically Exploiting XML: Zhang, Dyreson
Proposal: A Non-directional Axis
Proposal: A Non-directional Axis

Symmetrically Exploiting XML: Zhang, Dyreson
The Closest Axis

- **Syntax**
  - `closest::`
  - `->name` is abbreviation for `closest::name`

- **Semantics**
  - a function that takes a context node and returns a sequence of `closest` nodes
Closest Axis of the First Title

- \textit{closest::*}
  - Returns a list of five nodes
- \textit{closest::price}
  - Returns the first price node
When the First Book Lacks a Price

- Node selection restricted by *minimal type distance*
  - The minimal distance between a title and a price is 2
- *closest::price*
  - Returns an empty list
Type Distance is Crucial

- *closest::name* for each book?

- Root-to-node path type
  - *author/name*
  - *author/book/publisher/name*
Querying with the Closest Axes

Same query --
\[\text{return } \text{doc("any.xml")->author[->name='E. F. Codd']->book} \]

Symmetrically Exploiting XML: Zhang, Dyreson
Querying with Directional Axes

Query#1 -- \texttt{return doc("author.xml")//author[name='E. F. Codd']/book}

Result#1

Query#2 -- ……

Result#2

Result#3

Query#3 -- \texttt{return doc("book.xml")//book[author/name='E. F. Codd']}
In-memory Implementation

- Naïve approach
  - Compute *Closest* for every node
  - Time complexity is $O(sn^2)$
    - $s$: number of labels in the signature
    - $n$: number of nodes

- Converting to a path expression

Find the closest price for title
- Non-directional expression
  - `closest::price`
- Directional (path) expression
  - `parent::*/child::price`
Experiment

- Compare directional vs. nondirectional
  
  for $b$ in doc("bib.xml")//title/closest::publisher
  return $b

  for $b$ in doc("bib.xml")//title//..//publisher
  return $b

- Implemented closest in eXist (an XML DBMS)
Persistent Implementation

- Take advantage of type indexes
- LCA-join
  - Every *Closest* pair related via an LCA
  - Idea is to merge lists of types

- $O(sn)$
Related Work

- Data integration
  - TSIMMIS
  - YAT
    - Christophides, Cluet, Simèon (*SIGMOD Record* June 2000)
  - Silkroute
    - Fernandez, Tan, Suciu (*WWW* 2000)

- LCA-related techniques
  - Schmidt, Kersten, Windhouwer (*ICDE* 2001)
  - Cohen, Mamou, Kanza, Sagiv (*VLDB* 2003)
  - Li, Yu, Jagadish (*VLDB* 2004)
Related Research Projects

• XML Restructuring
  - Zhang, Dyreson (IIWeb 2006)

• XML Compaction
  - Zhang, Dyreson, Dang (DASFAA 2006)

• Common theme – symmetric exploitation!
Conclusion

• Current XQuery depends on path expressions

• A path expression is directional (asymmetric)
  - May break down if structure changes

• The closest axis is non-directional (symmetric)
  - Simple in syntax
    ◆ Can be easily integrated in XQuery
  - Can be implemented efficiently
    ◆ In-memory
    ◆ Persistent
Thank You!