Managing XML Documents with ToX

Alberto Mendelzon
University of Toronto

Project participants

- PIs: Leonid Libkin, Alberto Mendelzon
- BUL KM Lab: Patricia Rodriguez-Gianolli.
- IBM CAS: John Keenleyside, Kelly Lyons
- Post Doc: Laurent Mignet
- Ph.D. students: Marcelo Arenas, Denilson Barbosa, Attila Barta, Ariel Fuxman, Flavio Rizzolo
- M.Sc. Students: Jeff Deng, Karima Echihabi, Cecilia Pilar, Yidan Rogers, Angus Stewart, Ramona Truta, Edward Xia

Motivation

Requirements

- Manage XML data and meta-data
- Un/semi/structured
- Collect related docs
- Query within/across docs
- Multiple views
- Multiple back-ends

Alternatives

- Relational/OO dbms
- XML dbms
- File system + ad-hoc mechanism
- SGML tools

The ToX Project

- ToX: a repository manager for XML data
- Documents stored in a file system, a relational database, or remotely on the Web
- Metadata and collection management
- XQuery support within/across documents
- Client applications:
  - The EXecutive Information Portal (EXIP)
  - Bringing Evidence to the Point of Care (EPoCare)

```
<mls_entry>
    <house_description>
        This single family home built in 1965 has 3 bedr(s), 2 bath(s). Rooms
        include family room, master bedroom.
    </house_description>
    <agent_name company=REMAX> Bonnie Morris </agent_name>
    <office>623-334-2900</office>
    <residential_listing>
        <list_price>$310,900</list_price>
        <location>Phoenix, AZ 85051</location>
        <mls_#>1464957</mls_#>
        <bedrooms>3</bedrooms>
        <baths>2</baths>
        <style>Ranch</style>
        <type>Single Family</type>
    </residential_listing>
    …
</mls_entry>
```

element

attribute

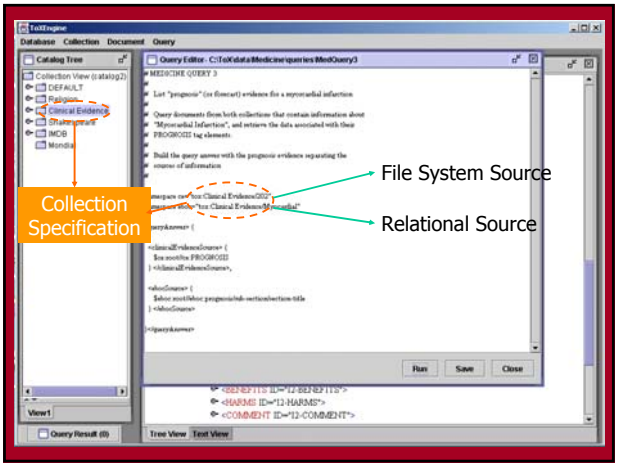
Semistructured Data

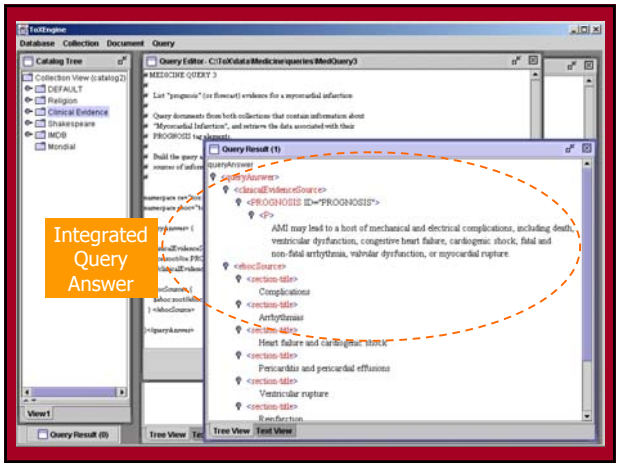
```
<north- america>
 <states>
  <state id = "s1"/>
   <sname> California
   </sname>
   <capital idref="c1"/>
   <governor> Gray Davis
   </governor>
 </state>
 ...
</states>
```

```
<provinces>
 <province id = "p1">
   <pname> Ontario </pname>
   <capital idref="c2" />
   <premier> Ernie Eaves
   </premier>
 </province>
...
</ provinces>
```

Semistructured data (cont.)

```
<cities>                                  ...
  <city id = "c1">                         </ cities>
    <cname>Sacramento</cname>
    <state- of idref = "s1"/>             ...
  </city>                                 </ north- america>
  <city id = "c2"/>
    <cname> Toronto </cname>
    <pop> 2.5M </pop>
    <province- of idref = "p1"/>
</ city>
```



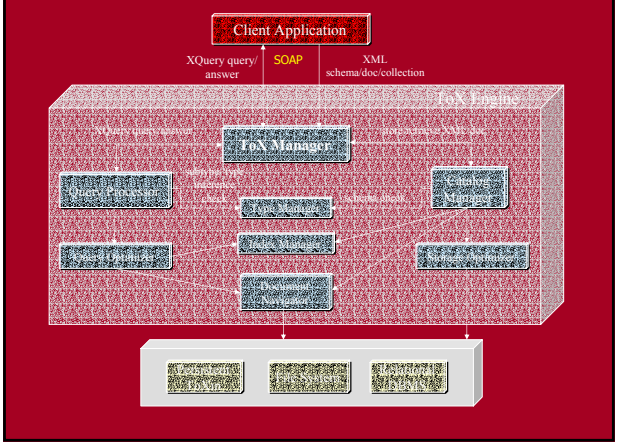


Document Type Definitions

```
<!ELEMENT catalog(book+,review+)>
<!ELEMENT book (title,author+,price)>
<!ATTLIST book isbn ID #REQUIRED>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT price (#PCDATA)>
<!ELEMENT review (#PCDATA)>
<!ATTLIST review isbn IDREF #REQUIRED>
```

Outline

- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type manager
- Related Tools
  - Data generation
  - Normalization
- Applications
- Research Challenges



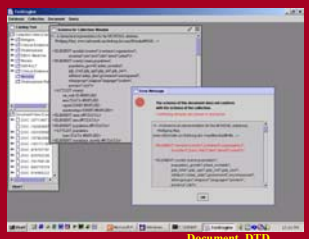
- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type managerDocument analysis
  - Data generation
  - Normalization
- Applications
- Research Challenges

Catalog Manager

- Document registration
- Collection management
- Storage and indexing choice

Document Registration

Document	*Collection*	*Conformance check*
DTD	No DTD	Check document conformance to its own DTD
No DTD	DTD	Check document conformance to collection's DTD
DTD	DTD	Check document's DTD is a subtype of collection's DTD.



Collection DTD

```
<!ELEMENT mondial (country*,
(continent |organization)*, mountain*,
sea*, river*, lake* ,desert*, island*)>
```

Document DTD

```
<!ELEMENT mondial (country*,
(continent |organization)*, mountain*,
(sea |  river)*, lake* ,desert*, island*)>
```

Automating Document Registration

- Automatically recommend collections for a document.

Col1 DTD	`<!ELEMENT country (name,population, population_growth?,infant_mortality?,government?, encompassed*,religion*,language?,border*,province*,city*)>`
Col2 DTD	`<!ELEMENT country (name,population, population_growth?,infant_mortality?,government?, encompassed*,religion*,language+,border*,province*,city*)>`
Col3 DTD	`<!ELEMENT country (name, population, language*)>`

Document DTD

```
<!ELEMENT country (name, population, language+)>
```

Future Work

Collection-independent queries

- Automatically select relevant collections and documents for a query.
- Need for type inference.

Col1	`<!ELEMENT country (name,population, population_growth?,infant_mortality?,government?, encompassed*,religion*,language?,border*,province*,city*)>`
Col2	`<!ELEMENT country (name,population, population_growth?,infant_mortality?,government?, encompassed*,religion*,language,border*,province*,city*)>`
Col3	`<!ELEMENT country (name, language+)>`

Query

```
for ($c in $root//country)
where ($c//language = "French" and $c//language = "English")
return $c/name
```

Schema Mapping

DTD of document

```
<!ELEMENT mondial (country*,(continent | organization)*,
        mountain*, sea*, river*, lake*, desert*, island*)>
```

DTD of recommended collection

```
<!ELEMENT geography (country*,continent*,geographic_feature*)>
```

- The user submits a document with a DTD.
- The system ***automatically*** recommends a collection with a ***similar*** DTD.
- The user gives correspondences between the DTDs.
- The system ***automatically*** builds a query that transforms the document.
- The transformed document is stored in the recommended collection.

- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type manager
- Related Tools
  - Data generation
  - Normalization
- Applications
- Research Challenges

ToX query language

```
#  PROGNOSIS FOR MYOCARDIAL INFARCTION
#
#  Extract PROGNOSIS from all documents in the two
#  MYOCARDIAL INFARCTION collections

let $ce:root := document("tox:Clinical Evidence Relational/0202") return
  let $eboc:root := document("tox:EBOC Medicine/Myocardial Infarction") return

<queryAnswer> {

  <clinicalEvidenceSource> {
    $ce:root//PROGNOSIS
  } </clinicalEvidenceSource>,

  <ebocSource> {
    $eboc:root//prognosis
  } </ebocSource>

}</queryAnswer>
```

Query Processor

- XQuery Core + some extensions
- Inter/intra document queries
- Location-transparent
- Queries compiled to navigational API
- XML output

- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type manager
- Related Tools
  - Data generation
  - Normalization
- Applications
- Research Challenges

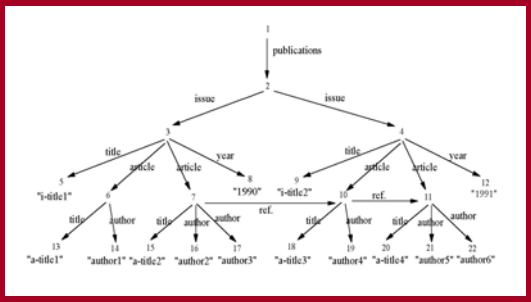
ToX Indexing

- ToXin: Structure + Value index
  - Goldman & Widom, Data Guides
  - Suciu & Milo, T-indexes
  - Kemper & Moerkotte, Access Support Relations
  - Cooper et al, Index Fabric
- Full Text Index
  - Navarro & Baeza-Yates, Proximal Nodes
  - Consens & Milo, Region Algebra
  - Jagadish et al, String Indexing

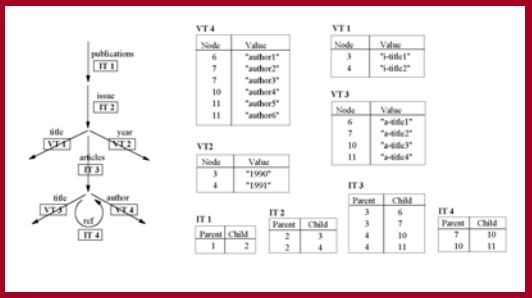
ToXin
(Rizzolo & M., WebDb'01)

- Main memory/relational XML index
- Supports forward and backward navigation from any node ("twig" queries)
- Path index: summarizes paths in document
- Value indices: support predicates on values

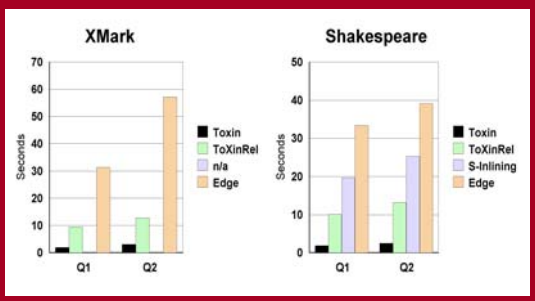
ToXin: XML Example



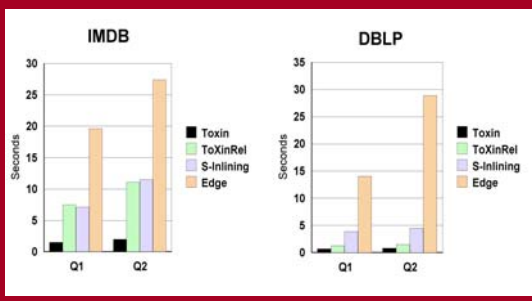
ToXin: Index Example



ToXin: Performance

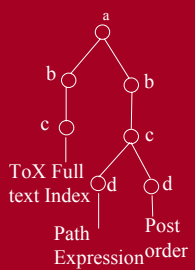


ToXin: Performance (cont.)



ToX Full Text Index

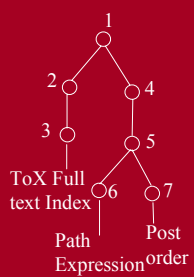
- map string to XPath expression.



- ToX => /a/b/c
- text => /a/b/c
- Expression => /a/b/c/d
- order => /a/b/c/d

ToX Full Text Index

- map string to node pre-order numbering



- ToX => 3
- text => 3
- Expression => 6
- order => 7

Using the indexes

- Query
  /Book[/Year="2002"]//Author[@name contains "Lacroix"]
- Use text index to find nodes that contain "Lacroix", structural index to find /Book[/Year="2002"] nodes, then join

- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type manager
- Related Tools
  - Data generation
  - Normalization
- Applications
- Research Challenges

Storage Mapper

- XML-to-relational mapping
- SQL implementation of navigation API
- Materialized path views for optimization

Relational DB - Approaches

Grammar Independent

Edge, Florescu & Kossmann
- Stores all edges in a single relation
- Maintains parent-child relationships

Binary, Florescu & Kossmann
- Stores all edges with the same label in the same relation

XRel, Yoshihawa et al
- Stores nodes based on its types (4 different relations)
- Label Path are kept
- Maintains inclusion relationships

XParent, Jiang et al
- Edge oriented approach
- Label Paths are kept
- Maintains parent-child relationships

Relational DB - Approaches

Grammar Dependent

Monet, Schmidt et al
- Explores an XML document instance to derive a relational schema
- Maintains associations of the same type in the same relation

STORED, Deutsch et al
- Extracts the most common structures in an XML document
- Stores uncommon structure in an overflow database
- Data Mining algorithm

Shared Approaches, Shanmugasundaram et al
- Analyzes information in DTDs to derive a relational schema

LegoDB, Bohannon et al
- Cost-based framework
- *p-Schema* = XML Schema + statistics
- Algebraic transformations over *p-Schema*
- Exploits the relational optimizer to derive an optimal schema

ToX Relational Mapping Scheme

- Grammar Independent
- Edge, Leaf, and Attribute relations
- Maintains order information
- DTD constraints checked on update
- Interval encoding of ancestor-descendant [Li and Moon VLDB'01]

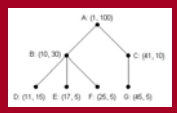
Interval Encoding

- Associates a pair of values (order, size) to each node in the XML graph, such that:

Condition:

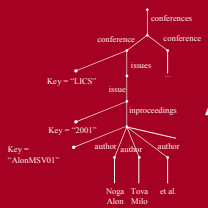
x is ancestor of y ⇔

order(x) < order(y) ≤ order(x) + size(x)



B is ancestor of F { 10 < 25 ; 25 ≤ 10 +30

Mapping an XML Document



Attribute Table

Source	Attr-Name	Char-Value	Attr-Type
3	Key	LICS	ID
5	Key	2001	ID
6	Key	AlonMSV01	ID
...	...	...	...

Edge Table

Source	Target	Name	Ordinal	Preorder	Size
1	2	Conferences	1	1	195
2	3	Conference	1	2	151
3	4	Issues	1	3	150
4	5	Issue	1	4	149
5	6	Inproceedings	1	5	148
6	7	author	1	6	9
...	...	...	...	...	...

Leaf Table

Source	Target	Type-Value	Char-Value	Int-Value	Ordinal	Preorder	Size
7	8	CHAR	Noga Alon		1	7	1
...	...	...	...	...	...	...	...

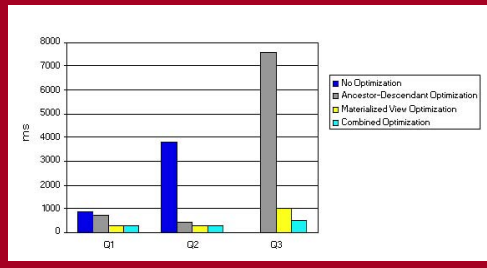
Materialized View Reuse

- Regular expression views
- Materialized views chosen from query workload
- Extended Calvanese et al query rewriting algorithm

Performance Evaluation

Data set:

✓ 6 MB, 150K tuples, highly nested



Incremental validation of XML documents

- Revalidating documents after updates
- Updates: insertions and deletions of subtrees
  - Append(x,y) insert x as last child of y
    InsertBefore(x,y) insert x as immediate left sibling of y
    - First validate x, then re-validate the document
  - Delete(x):
    - re-validate the document

DTDs as extended CFGs

- Rules of the form $e \rightarrow r$, r a 1-unambiguous regular expression
  - Membership in r can be checked by a linear size DFA A_r
  - DTDs are LL(1) grammars
- Incremental validation
  - Store the computation of A_r (O(n) space)
  - Partially re-do computation on update (O(n) worst case)

Validating Restricted DTD's

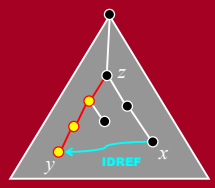
- 1,2-Conflict-free DTD's: O(log n) time, O(n) space
- Simple DTD's: O(log n) time, O(1) space

Updates with ID/IDREF(S)

- Keep reference counter for each node
- Insertions:
  - Check for duplicate IDs and "dangling" IDREF(S)
  - Update the reference counters
    - Both for the subtree being inserted and the document as a whole
    - Finding ancestors requires a recursive query (O(log n) iterations), but XML documents are shallow
- Deletions:
  - Check and update the reference counters (requires constant time)

Deleting nodes with ID/IDREF(S)

- Consider an ID/IDREF reference from *x* to *y*, and let *z* be their closest common ancestor
  - No ancestor of *y* (up to *z*) can be deleted
- Handling several references
  - Use a reference counter for each node (requires O(n) space)
  - Allow deletions only if the counter is set to 0



- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type manager
- Related Tools
  - Data generation
  - Normalization
- Applications
- Research Challenges

Type Manager

- Type system for XML: DTD's, XML Schema, XQuery
- Type checking: is T1 a subtype of T2?
- Type inferencing: if D is of type T, what's the type of Q(D)?
- Type lookup: is T1 a subtype of any type known to the system?

Related Work

- Subsumption for XML Types, Kuper & Simeon, ICDT 2001
- Typechecking for XML Transformers, Milo, Suciu & Vianu, PODS 2000
- XDUCE, Hosoya et al, ICFP 2000
- Term indexing for PROLOG, Ramesh et al, JLP 1995

Type Index Example

DTD of Collection 1	DTD of Collection 2
<!ELEMENT author (name, email*)> <!ELEMENT name (#PCDATA)> <!ELEMENT email (#PCDATA)>	<!ELEMENT author (name, email | tel)> <!ELEMENT name (#PCDATA)> <!ELEMENT email (#PCDATA)> <!ELEMENT tel (#PCDATA)>

- Type t_1: <!ELEMENT author (name,email*)>
- Type t_2 : <!ELEMENT author (name,email | tel)>
- Type Index Construction:
  - Converting DTD types into a binary tree representation (with first level unrolling of Kleene stars)
  - Merging type trees into the index trie

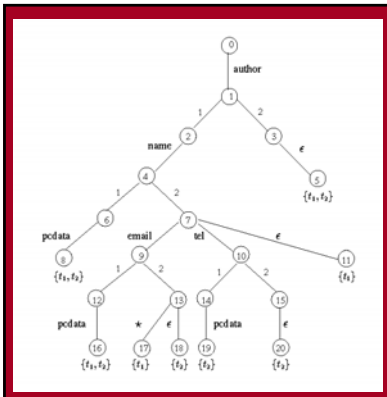


Figure 1: Index Trie for types t_1 and t_2

ε is empty type and * is any type

DTD of Input Document

<!ELEMENT author (name, tel)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT tel (#PCDATA)>

Retrieval algorithm

Collection 2 selected as the only relevant collection for the input document

Application

- Document Registration
  - Checking that the DTD of a document is a subtype of the DTD of a collection
  - Efficiently selecting relevant collections for a particular document
- Queries:
  - Running a query on selected documents or collections

- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type manager
- Related Tools
  - Data generation
  - Normalization
- Applications
- Research Challenges

ToXgene
(Barbosa et al, WebDb'02)

- XML synthetic data generator
- Multiple probability distributions
- Correlated sets of documents
- XML-Schema based

ToXgene - Motivation

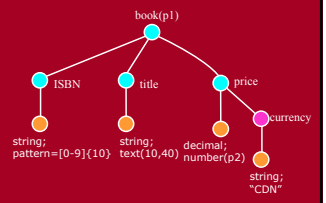
- Real data: hard to get, confidential, messy
- Need synthetic data sets for testing and benchmarking
- Need data sets of widely varying properties without writing/adapting hard-coded data generators

ToXgene in a Nutshell

- Declarative:
  - Templates are annotated XML Schema specs.
- Generates correlated (joinable) documents
- Supports integrity constraints
- Controlled randomness
- Supports reuse of real data

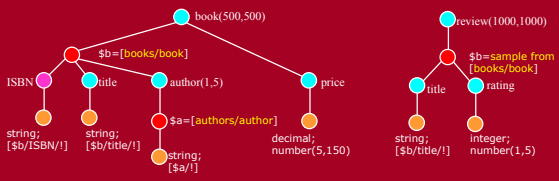
ToXgene templates

```
<element name="book" tox-distribution="p1">
 <ComplexType>
  <element name="ISBN" type="isbn_type">
  <element name="title">
   <SimpleType>
    <restriction base="string">
     <minLength="10"/><maxLength="40"/>
     <tox-string type="text">
    </restriction>
   <SimpleType>
  <element name="price">
   <ComplexType mixed="true">
    <attribute name="currency" type="string">
     <tox-expr value="'CDN'"/>
    </attribute>
   <tox-number tox-distribution="p2"
      format="0.00"/>
   </ComplexType>
  </element>
 </ComplexType>
</element>
```

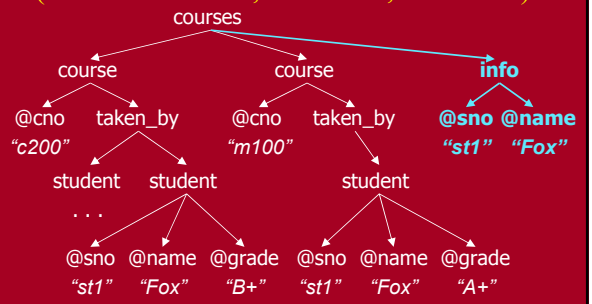


Element Sharing by Querying

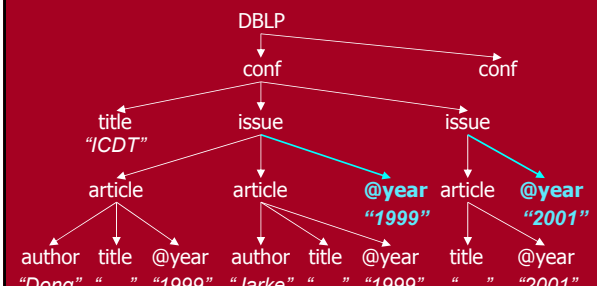
- Store the shared content into lists
  - Define different document templates querying the same lists



Normalization of XML Documents (Arenas & Libkin, PODS'02, PODS'03)



A "Non-relational" Example



Results

- XNF: XML Normal Form
- XML normalization algorithm
- Functional dependency implication
- For "simple" DTDs:
  - The implication problem for FDs is solvable in quadratic time.
  - Testing if a specification is in XNF can be done in cubic time.

- Motivation
- Architecture
  - Catalog manager
  - Query processor
  - Index manager
  - Storage optimizer
  - Type manager
- Related Tools
  - Data generation
  - Normalization
- Applications
- Research Challenges

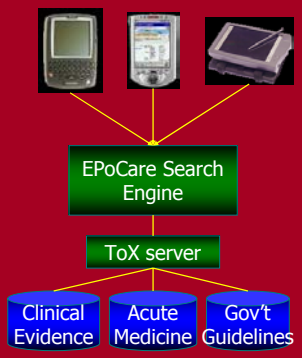
Evidence-Based Medicine

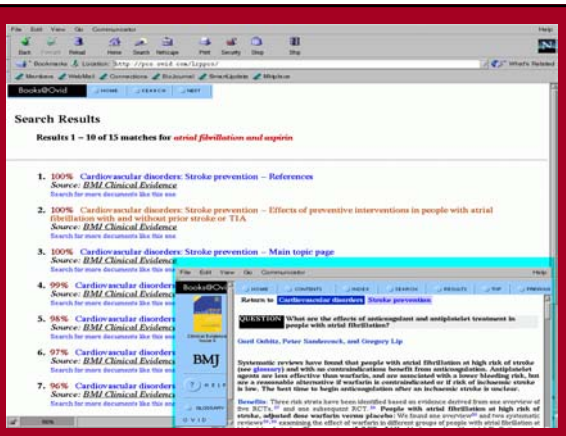
"Some experts estimate that only 20 percent of medical practices are based on rigorous research evidence. The rest are based on what has been published in books repeatedly without independent testing – or what doctors have always said should work. In other words, it's a kind of folklore."

NYT, December 2001

The EPoCare Project

- What kinds of information are most useful for clinicians?
- What is the most effective way of querying evidence-based resources?

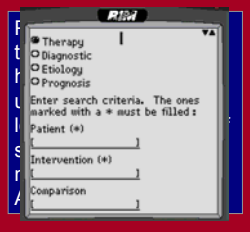


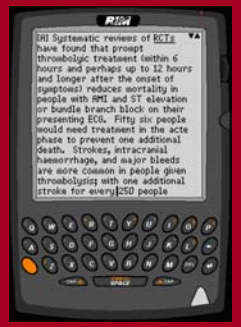


Focusing Queries

- Query by example interface
- User first specifies whether the query is about therapy, diagnosis, etiology, or prognosis
- PICO format

Delivering Information at the Point of Care





ToX for EPoCare

- Data sources in XML format
- Integration of evidence based resources
  - Answers to a query come from multiple sources
  - Sources need to be identified
- Mixed keyword and structural queries

Research Challenges

- Heterogeneous query optimization
- Adaptive query processing
- Distributed catalogs
- Approximate type matching
- Document analysis
  - Mining data for ToXgene templates
  - Storage mapping selection
  - Intelligent indexing

More information: www.cs.toronto.edu/tox

Thank you.

What we did

- Typechecking for schemas (DTDs)
  - Semantic definition: DTD *d1* is a subtype of DTD *d2* iff the set of all documents conforming to *d1* is included in the set of documents conforming to *d2*.
  - Indexing mechanism: Checking subtyping between an input DTD and a database of DTDs in an efficient way. Index acts as a filter in the presence of kleen stars

ToXin: Characteristics

- Edges with the same labels in distinct label paths are stored in distinct tables
- There is no overlapping among the Value Tables, the Instance Tables and the Index Graph
- Size is linear (worst case) w.r.t. the size of the XML graph

ToXin: Queries

- XMark
  - Q1: Closed_auctions/closed_auction[//date="%1998"] /annotation//listitem
  - Q2: open_auctions/open_auction[//start="%1998"] /annotation//listitem
- Shakespeare
  - Q1: file/play[title="%The%"]/act//speech/line
  - Q2: file/play[title="%The%"]/act//speaker

ToXin: Queries (cont)

- IMDB
  - Q1: movies/movie[genre="Drama%"]/title
  - Q2: movies/movie[year="198%"]//cast_member
- DBLP
  - Q1: conference[title="t1"]//issue/inproceedings/title
  - Q2: conference//issue[proceedings/year="y1"]/inproceedings/title

XML Index: ToX

- Help to express a Query (Epocare)
- Structural Index and Full Text Index are similar: associate a string a value (path-expression or value).

XML Index: Survey

- Current works resolved a query like /a/b[contains "foo"] by resolving first the path expression and next the condition.
- We want to extend the work of Consens-Milo to express an Algebras for XML Queries based on a cost function.

References

- [Choi02] B. Choi. What are real DTDs like? Proceedings of WebDB 2002.

ToX Relational References

[AB02] M. Arenas and D. Barbosa. YAM: Yet another (XML/Relational) mapping. 2002. Unpublished. University of Toronto.

[BFRS02] P. Bohannon, J. Freire, et al. From XML Schema to relations: A cost based approach to XML storage. In Proceedings of the 18th International Conference of Data Engineering, 2002.

[CGLV99] D. Calvanese, G. De Giacomo, et al. Rewriting of regular expressions and regular path queries. In Proceedings of the 18th ACM SIGACT Symposium of Principles of Database Systems, 1999.

[DFS99] A. Deutsch, M. Fernandez, et al. Storing Semistructured data with STORED. In Proceedings of ACM SIGMOD International Conference on Management of Data, 1999.

[FK99] D. Florescu and D. Kossmann. A performance evaluation of alternative mapping schemes for storing XML data in a relational database. Technical Report, INRIA, 1999.

[JLWY02] H.Jiang, H. Lu, et al. Xparent: An efficient RDBMS based XML database system. 2002.

[KM92] A. Kemper and G. Moerkotte. Access Support Relations: An indexing method for object bases. Information Systems, 1992.

[LM01] Q. Li and B. Moon. Indexing and querying XML data for regular path expressions. In VLDB Journal, 2001.

[SKWW00] A. Schimidt, M. Kersten, et al. Efficient relational storage and retrieval of XML documents. In WebDB (Informal proceedings), 2000.

[STZ+99] J. Shanmugasundaram, K. Tufte, et al. Relational databases for querying XML documents: Limitations and Oportunities. In VLDB Journal, 1999.

[SYU99, YASU01] M. Yoshihawa, O. Amagasa, et al. Xrel: A path based approach to storage and retrieval of XML documents using relational databases. In ACM Transactions On Internet Technology, 2001.

Index Manager

- ToXin index structure (Rizzolo & Mendelzon, WebDB'01)
- Path index: summarizes paths in document
- Value indices: support predicates on values
- ToX backend can be file system, relational db, or ToXin

XML Index: Survey

- Regions Queries: answer "string under a path expression"
  - Jagadish, Koudas, Srivastava [SIGMOD'00]
    - Multidimensional-string
  - Consens, Milo [PODS'95]
    - Algebras for Queries Text Regions (SGML)

XML Index: Survey

- Resolve Path Expression: associate for each label (string) an unique position number.
  - Li and Moon [VLDB'01] + Xyleme + ...:
    - post-order; pre-order; level
  - Milo [SODA 2001; PODS 2002]:
    - Dynamic and compact labelling

Our Goal

DTD:

courses ⇒ course*, **info***
course ⇒ taken_by
course ⇒ @cno
taken_by ⇒ student*
student ⇒ @sno, ~~@name~~, @grade
info ⇒ @sno, @name

Integrity Constraints:

~~@sno is the key of info~~
same @sno value must have the same name

Problems to Address

- Language for expressing functional dependencies
  - Absolute and relative constraints
- Normal form for XML documents (XNF)
- Algorithm for normalizing XML documents
  - Implication problem for functional dependencies

XML Functional Dependencies

- (Absolute) Two **students** with the same **@sno** value must have the same **@name**:

  courses.course.taken_by.student.@sno → courses.course.taken_by.student.@name

- (Relative) Every **student** can have at most one **@grade** in every **course**:

  { courses.course, courses.course.taken_by.student.@sno } → courses.course.taken_by.student.@grade

XNF: XML Normal Form

- XML specification: a DTD **D** and a set of functional dependencies **Σ**
- **(D, Σ)** is in **XNF** if:

  For each non-trivial FD **X → p.@l** implied by **(D, Σ)**, **X → p** is also implied by **(D, Σ)** .

Back to DBLP

- DBLP is not in XNF:

  DBLP.conf.issue → DBLP.conf.issue.article.@year
  is implied by (D,Σ)

  DBLP.conf.issue → DBLP.conf.issue.article
  is not implied by (D,Σ)

- Proposed solution is in XNF

Clinicians' Information Needs

- Daily need for valid information
- Traditional sources are inadequate
- Disparity between diagnostic skills and clinical judgement

Future Work

- Reducing false positives
- Materializing Index
- Supporting XML Schema
- Type Inference

Related work

- IBM XML Generator:
  - Template based (templates are DTDs)
  - Limited tools for specifying probability distributions and content of elements
- WebDB'2001:
  - Random structures and random content
- Xmark data generator:
  - Consistent ID/IDREFs
  - Hard-coded data generator

Challenges in Query Planning Paradigm

- The nodes in the query tree are no longer relations but views containing many sources on several sites.
- No longer sargable predicates - rather predicates based on regular path expressions.
- Executing selections/projections at scan time is no longer an option.
- Which data statistics to collect?

Evidence-Based Medicine

- "A revolution is erupting in the wards of practical medicine these days, one defined recently by The British Medical Journal as ''the conscientious, explicit and judicious use of current best evidence in making decisions about the care of individual patients.'' The revolution is called evidence-based medicine, or E.B.M., and many traditional treatments are being run through the machinery of the scientific method – and being found wanting."

  NYT, December 2001

XML and Databases

- Designing
- Storing
- Querying
- Mining
- Update/Transactions
- Security