Extreme Java
G22.3033-007

Session 3 - Sub-Topic 6
XML Information Retrieval

Dr. Jean-Claude Franchitti

New York University
Computer Science Department
Courant Institute of Mathematical Sciences

Agenda

■ Applications of XML to Database Technology
  ■ XML Query Languages
  ■ XPath
  ■ XML Queries
■ XQuery: A Query Language for XML
■ XML Query Engines
■ XML Object Persistence
■ Advanced XQuery Concepts
■ Presentation Oriented Publishing (POP)
  Frameworks
XML-Based Retrieval Development

- XML Software Development Methodology
  - Language + Stepwise Process + Tools

- XML Application Development Infrastructure
  - Metadata Management (e.g., XMI)
  - XML Query Engine (3rd party software)
  - XML Tools (e.g., XML Editors)

- XML Applications Involved in the Rendering Phase:
  - Application(s) of XML
  - XML-based applications/services (markup language mediators)
    - MOM, POP, Persistence service
    - Application Infrastructure Frameworks

XML Data Retrieval Patterns

- XML Data Retrieval Operations
  - Access
  - Query
  - Manipulate

- Multiple XML Data Sources Integration
- XML Message Filtering
- DBMS Data Views
- Database System Interfacing
Part I

Applications of XML to Database Technology

Towards XML Application Services

- Processing
  - DOM Extensions
  - Binding Extensions
  - Component Frameworks (reusable component models)
  - Model-Based Automation (MDA)
- Rendering
  - DOM 2.1.0, SAX 2.0, JAXP 1.1 & TraX, XSL-FO 1.0
  - Component Frameworks
- Querying
  - XQuery 1.0, XSLT 1.1/2.0, XPath 1.0/2.0
- Security (signatures encryption/decryption, etc.)
- Etc.
Retrieval Software Development

- Languages (XML-QL, YaTL, XQL, etc.)
  - Data Model + Operations + Syntax
- Process ("XUP")
- Frameworks
  - Custom Engine
    - e.g., XQEngine
  - Translation to SQL
    - e.g., DB2XML, Oracle’s XML I/F
  - Translation to OQL
- XML Query Infrastructure
  - XPath Processors: Saxon 6.1, Xalan-J 2.1.0
  - XQuery Processors
  - SQLX

XML Query History

- SGML Query Facilities
- Ad-hoc Approach to Query Languages
- 02/98: XQL Proposal
- 08/98: XML-QL Submission
- 12/98: W3C QL’98 Workshop
  - Candidate Requirements for XML Query
  - Database Desiderata for an XML Query Language
- 11/99: XPath Recommendation
W3C XML Query WG

- 07/99: WG Proposal
- 09/99: WG Official Inception
- Today:
  - 30 W3C Member Companies
  - 11 Meetings, 60+ Telcons
  - Heartbeat every Three Months
  - Proposed Recommendation(s)
- Goals:
  - XML Query Data Model for XML Documents
  - Query Operators for XML Query Data Model
  - Query Language Based on XML Query Operators

W3C’s Related Standards

- XML Query Specifications:
  - *XML Query Requirements* (02/16/01 - orig. 01/00)
  - *XML Query Use Cases* (06/08/01 - orig. 08/00)
  - *XQuery 1.0 and XPath 2.0 Data Model* (06/07/01 - orig. 05/00)
  - *XQuery 1.0 Formal Semantics* (06/07/01 - orig. 12/00)
  - *XQuery 1.0: An XML Query Language* (06/07/01)
  - *XML Syntax for XQuery 1.0 (XQueryX)* (06/07/01)
- XPath 2.0 Specifications
  - *XPath Requirements Version 2.0* (02/15/01)
  - *XQuery 1.0 and XPath 2.0 Data Model* (06/07/01)
Related XML Technologies

- XPath
- XSL
- XPointer
- XML Schema
- XML Infoset
- WAI
- Internationalization
- IETF DASL
  - Distributed Authoring Searching and Locating

Properties of RDBMS Queries

- Pattern + Filter + Construction clause
- Construction clause may have ordering subclauses
- Queries may perform joins across multiple input sets
- Queries may generate intermediate variables or path expressions
Mapping XML to a RDBMS

- SQL-like queries that return XML documents
  - e.g., Microsoft IIS + SQL Server
  - e.g., Oracle Database Server
- Broad spectrum of possible mappings
  - Hierarchical v.s. limited RDBMS tree structure

JDBC Refresher

- See section 6.2 of XML and Java textbook
  - Importing JDBC Package
  - Loading a JDBC Driver
  - Connecting to a Database
  - Submitting a Query
XML Embedded in SQL (SQLX)

- SQL Embedded in XML
  - See Section 6.3 of XML and Java textbook
  - Front-end to RDBMS that provides XML-based Input/Output
  - Translates XML query into sequence of JDBC calls, and converts the result to a DOM structure which is returned

Part II

XML Query
XML Query Requirements (Part I)

- General:
  - Declarative Language
  - Readable XML Syntax
  - Protocol Independence
  - Standard Error Conditions
  - Support for Future Updates

- Data Model
  - Based on XML Infosets
  - Namespace Aware
  - Support for XML Schema Data Types
  - Support Inter/Intra Document References

XML Query Requirements (Part II)

- Query Functionality:
  - Operators on All Data Types
  - Text Operators Across Element Boundaries
  - Hierarchies and Sequences
  - Combination of Data from Various Locations
  - Aggregation and Sorting
  - Combination of Operators (Queries as Operands)
  - Support NULL values
  - Preservation of Structure/Identity
  - Operations on Names/Schemas
  - Extensibility & Closure
XML Query Use Cases

- Approach
  - Description, DTD/Schema, Input, Queries, Results
- Existing Use Cases
  - XMP (examples)
  - TREE (queries that preserve hierarchy)
  - SEQ (queries based on sequence)
  - R (relational data access)
  - TEXT (text search)
  - NS (namespace-based queries)
  - PARTS (recursive parts queries)
  - REF (queries based on references)

XML Query Data Model

- Information Presented to a Query Processor
- Augmented Infoset:
  - XML Schema Data Types (PSVI)
  - Document Collections
  - References
- Node-Labeled Tree Constructor Model with Node Identity
- Infoset Mapping to Query Data Model is Defined as Part of the Specification
**XML Query Data Model**  
(continued)

- **Nodes**
  - Node = DocNode | ElemNode | AttrNode | ValueNode  
    | NSNode | PINode | CommentNode | InfoItemNode

- **XML Schema Primitive Types**
  - string, boolean, ID, IDREF, decimal, etc.

- **Collections**
  - list [T], set {T}, bag {T}, disjoint/union (T1 | T2),  
    tuple (T1, …, Tn)

- **References**
  - ref(T)

---

**XML Query Algebra**

- **Defines Static and Dynamic Semantics**
  - Static Semantics are Type Inference Rules
    - Relate Algebra Expressions to Types
  - Dynamic Semantics are Value Inference Rules
    - Relate Algebra Expressions to Values

- **Issues:**
  - Algebra Type System Alignment with XML Schema
  - Operators on Schema Simple Types not Defined
  - Lexical Representation of Schema Simple Types not Defined
Constructors

- **Construct Values in XML Query Data Model**

  attrNode : (Ref(QNameValue), Ref(ValueNode))
  -> AttrNode
  ValueNode = QNameValue | StringValue | DecimalValue | ...
  qnameValue : (uriReference | null, string,
  Ref(Def_QName))
  -> QNameValue
  decimalValue : (decimal, Ref(Def_decimal))
  -> DecimalValue

  <part price=10.50/>
  <xsd:attribute name="price" type=xsd:decimal/>
  attrNode(ref(qnameValue(null, “price”),
  Ref(Def_QName)),
  ref(decimalValue(10.50, Ref(Def_decimal))))

Assessors

- **Deconstruct Values in XML Query Data Model**

  name : AttrNode -> Ref(QNameValue)
  value : AttrNode -> Ref(ValueNode)
  type : AttrNode -> Ref(ElemNode)

  <xsd:attribute name="price" type=xsd:decimal/>
  <part price=10.50/>

  name(A1) = ref(qnameValue(null, “price”))
  value(A1) = ref(decimalValue(10.50, Ref(Def_decimal)))
  type(A1) =
  <!-- data model representation of simple type
  decimal -->
Part III

XML Query Languages

XQuery

- Functional Language
  - Query Represented as an Expression
- Expressions can be Nested without Restriction
- Input/Output of an XQuery are Instances of the XML Query Data Model
- Based on OQL, SQL, XML-QL, XPath
- Readable XML Syntax
XQuery Expressions

- Path Expressions
- Element Constructors
- FLWR Expressions
- Expressions with Operators/Functions
- Conditional Expressions
- Quantified Expressions
- List Constructors
- Expressions to Test/Modify Datatypes

XQuery Path Expressions

- Abbreviated XPath 1.0 Syntax
  - Find figure(s) with caption “Tree Frogs” in second chapter of “zoo.xml”
  - `document(“zoo.xml)/chapter[2]/figure[caption = “Tree Frogs”]`
- Extensions
  - Dereference Operator
  - Range Predicate
  - Find captions of figures referenced by `<figref` elements in “Frogs” chapter of “zoo.xml”
  - `document(“zoo.xml”)/chapter[title = “Frogs”]/figref/@refid->fig/caption`
XQuery Element Constructor

- Start/End Tag + Enclosed List of Expressions
  - Generate an element with a computed name that contains nested elements:
    
    ```xml
    <$tagname>
    <description> $d </description>
    <price> $p </price>
    </$tagname>
    ```

XQuery For Let Where Return (FLWR)

- FOR and LET Clause
  - Generate a List of Tuples that Preserves Doc Order
- WHERE Clause
  - Applies a Predicate to Eliminate Some Tuples
- RETURN Clause
  - Executed on Resulting Tuples -> Ordered Output List
- Syntax:
  ```
  FOR var IN expr  WHERE expr  RETURN expr
  LET var := expr
  ```
FLWR Sample Expressions

- List titles of books published by MK in 98
  FOR $b$ IN document (“bib.xml”)//book
  WHERE $b/publisher$ = “Morgan Kaufmann”
  AND $b/year$ = “1998”
  RETURN $b/title$

- List each publisher and its books average price
  FOR $p$ IN
  distinct(document(“bib.xml”)//publisher)
  LET $a :=$ avg(document(“bib.xml”)
  /book[publisher = $p$]/price)

XQuery Operators and Functions

- Infix/Prefix Operators
  - e.g., Infix Operators BEFORE and AFTER
- Parenthesized Expressions
- Arithmetic/Logical Operators
- Collection Operators
  - e.g., UNION, INTERSECT, EXCEPT
- Functions Can Be Defined in XQuery
Sample Operators and Functions

Find max depth of “partlist.xml”

NAMESPACE
xsd="http://www.w3.org/2001/03/XMLSchema-datatypes"

FUNCTION depth(ELEMENT $e) RETURNS xsd:integer
{
    IF empty ($e/*) THEN 1
    ELSE max (depth($e/*))+1
}

depth(document("partlist.xml"))

XQuery Conditional Expressions

FOR $h IN //holding
    RETURN
        <holding>
            $h/title
            IF $h/@type="Journal" THEN $h/editor
            ELSE $h/author
        <holding> SORTBY (title)
XQuery Quantified Expressions

- Example 1:
  FOR $b$ IN //book
  WHERE SOME $p$ IN $b$//para SATISFIES
    contains($p$, “sailing”) AND contains($p$, “windsurfing”)
  RETURN $b$/title

- Example 2:
  FOR $b$ IN //book
  WHERE EVERY $p$ IN $b$//para SATISFIES
    contains($p$, “sailing”)
  RETURN $b$/title

XQuery List Constructors

- List encloses zero or more expressions in square brackets, separated by commas
- List of member variables: [$x$, $y$, $z$]
- Empty list: [ ]
XQuery Operators on Data Types

- `INSTANCEOF (instance, type)`
- `CAST`
  - Convert value from one datatype to another
- `TREAT`
  - Causes the query processor to treat an expression as if its datatype were a subtype of its static type

XQuery Outstanding Issues

- Integration with XPath 2.0
- Alignment of XQuery and XML Query Algebra Syntax
- Internationalization
  - e.g., Collation Sequences for Sorting, Strings ops
- XML Query Syntax
- Operators and Functions TBD
Part IV

XML Query Engines and Advanced Concepts

Various Approaches

- **XQEngine**
  - Full-text search engine for XML
  - Java APIs available
  - W3C XQuery Specification Support

- **DB2XML**
  - Standalone tool (with GUI or command line)
  - Servlet to dynamically generate XML-documents
  - DB2XML API

- Oracle XML Developer Kit (**XDK**)  
- Microsoft **SQL Server** support for XML
XML Object Persistence

- Started as **SODL and XMOP**
  - Simple Object Definition Language
  - XML Metadata Object Persistence
- XML and JavaBeans integration (e.g., BML, Coins, etc.)
- XML and EJB integration
  - See XML Development with Java 2 (chapter 8)
- XML serialization for Java (e.g., Koala, etc.)
- SOAP - XML-RPC protocol

Advanced XQuery Concepts

- Mainstream XQuery Engines
  - Software AG’s QuiP
  - H. Katz XQEngine
- Experiment with Complex Queries and QuiP
POP Frameworks

- Client-Side POP
  - IE5
- Server-Side POP
  - Cocoon & XSP
  - Rocket
  - CPAN’s Perl Framework

Part V

Conclusions
Summary

- Applying XML to Database Technology allows the viewing of database data as an XML document.
- XML Query is based on a well defined Data Model and Algebra
- Various syntaxes are possible for an XML Query Language
- XML Query Engines are infrastructure components that support XML Query

Summary (continued)

- Bindings approaches are currently implemented between XML and JavaBeans/EJBs
- Software AG’s Quip implements complex query processing as per XQuery 1.0
- Server-side POP is the approach of choice for XML processing