Managing Semi/Unstructured Data

Mukesh Mohania
IBM India Research Lab
mkmukesh@in.ibm.com

Outline

- Unstructured, XML and Semi-structured Data
- Techniques for storing XML/Semi-structured data
- XML Query Over Relational Data
- Streaming Data (semi-structured) Management
- Active Integration of Information
- Semantic Web
- Applications
- Content Manager Architecture

Unstructured Information

- On-line business information is unstructured -- mainly text.
- 80% of content is unstructured.
  - Static content: word processor documents, html files, emails, text files, many more
  - Dynamic content: extracted from underlying databases
  - Anything on the web (static or dynamic)
- Properties of Data on Web
  - Web data cannot be constrained by a type or schema.
  - It has irregular structure and deeply nested.
  - Its structure keeps evolving.
  - Web data is very much distributed and linked.
  - Data having such properties called semi-structured data.

XML: eXtensible Markup Language

- World Wide Web Consortium (W3C) standard to complement HTML
- HTML: Text + Presentation (no data)
- XML: Data + Structure (describes contents)
- Two modes
  - Well formed XML: schema-less, semi-structured data, user-defined tags, self-describing data
  - Valid XML: contains DTD for tags specification and grammar of the document, not completely schema-less
- Used for data exchange, transformation, and integration; bridge for data exchange on the web
- XML Standards: Schema (XML Schema), XSL, RDF, XPATH, Xquery and others

XML Example

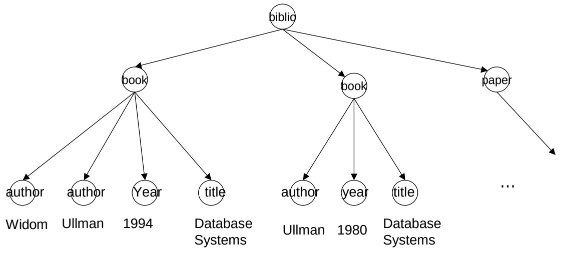
Well formed XML

```
<? XML VERSION=“1.0” STANDALONE=“YES” ?>
<Here-is-my-tag>
     <another my-tag>
        …
     </>
</>
```

Valid XML

```
<? XML VERSION=“1.0” ?>
<!DOCTYPE BIBLIO [
          <!ELEMENT BIBLIO (BOOK*, PAPER*)>
          <!ELEMENT BOOK (Author+, Year, Title)>
          <!ELEMENT PAPER (Author+, Year, Title, Source)>
          <!ELEMENT Author (#PCDATA)>
           …
]>
```

Tree for XML Data



Ordered Elements (except attributes)

Semi-structured Data

- Schema-less and self-describing, but the schema is attached to the data itself
- Schema is defined before/after the data, may not be enforced, schema may be extracted from data or from queries (like type inference in PL)
- Origins
  - Integration of heterogeneous sources (Web + DB + … = ?)
  - Data sources with non-rigid structure (biological data)
  - Web data

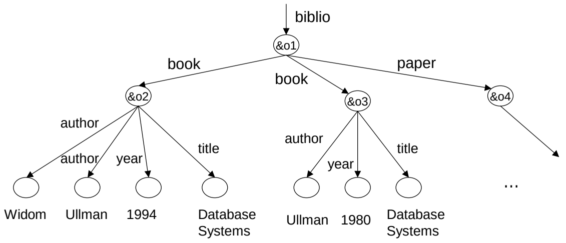
Schema…

- The need for schema
  - Optimize query processing
  - Facilitate integration of multiple data sources
  - Improve storage
  - Construct indexes
  - Describe contents of database to improve browsing and query formulation
  - Forbid certain types of updates

A Bad Example: As of April 1, 3 of 12 major banks of Japan (Dai-ichi Kangyo, Fuji and Industrial banks) were merged into World's biggest bank, called Mizuho Bank Ltd, …… database integration conflicts caused six days of chaos involving more than 30,000 transaction errors and more than 2.5 million delayed debits ….(ATM) transaction errors.

SoI: Computerworld Inc. by Kuriko Miyake, IDG News Service, April 08, 2002.

Semi-structured Data Model



Unordered elements

Example: Object Exchange Model

Techniques for Storing XML

- Why new storage techniques?
  - To support the characteristics of XML data and queries
    - Optional elements, repetition of tags, ordering, mixed contents (structured data embedded in large text fragments), etc.
    - Document order and structure, full text search, transformation

Techniques for storing XML

1. Store the entire document as a file in a file system or as a BLOB in a RDBMS (Flat streams)
   - Fast store/retrieve whole documents or big continuous parts of documents
   - Access the documents' structure through parsing
2. Using existing models
   - Mapping from XML graph/tree into Relational, OO, LDAP directories
   - Take advantages of Indexing, recovery, transactions, updates, query optimization, security, etc
   - No support for mixed content
   - XML document recovery is expensive!
   - Introduces additional layers in DBMS, therefore slower
3. Mixed (both files and relational tables)… but Redundant
4. Native XML data model
   - Logical data model is XML
   - Physical storage features designed for XML

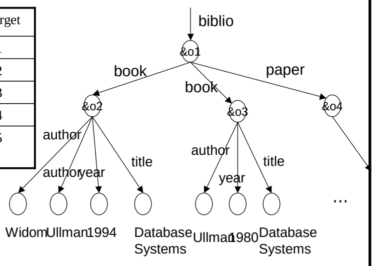
Mapping into Relational Model

- **Edge Relation:** Store all edges in one table and scalar values in another table
- **Schema-driven**
  - Mapping from schema constructs to relational
  - Fixed mapping from DTD to relational schema
  - Flexible mapping from XML Schema to relational
- **Universal Relation:** Full outer join, but redundancy
- Captures node identity & document order
- Element reconstruction requires multiple joins
- Does not use DTD or XML schema

Edge Relation Example

Edge table

Source	Ordinal	Tag	Flag	Target
&0	1	biblio	ref	&1
&1	1	book	ref	&2
&1	2	book	ref	&3
&1	3	paper	ref	&4
&2	1	author	string	&5
…				



Node	Value
&5	Widom
&6	Ullman

Value table

Schema Driven Mapping

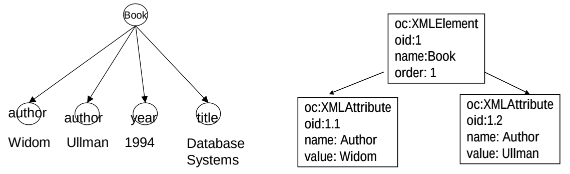
- *Repetition* : separate tables
- *Non-repeated* sub-elements may be “inlined”
- *Optionality* : nullable fields
- *Choice* : multiple tables or universal table
- *Order* : explicit ordinal value
- *Mixed content* ignored
- Element reconstruction may require multi-table joins because of normalization

LDAP Example

```
XMLElement OC {
  SUBCLASS OF {XMLNode}
  MUST CONTAIN {order}
  MAY CONTAIN {value}
  TYPE order INTEGER
  TYPE value STRING }
```

```
XMLAttribute OC {
  SUBCLASS OF
    {XMLNode}
  MUST CONTAIN {value}
  TYPE value STRING }
```

- Tailored to evolving Schema
- Captures node identity & document order



oc:XMLElement
oid:1
name:Book
order: 1

oc:XMLAttribute
oid:1.1
name: Author
value: Widom

oc:XMLAttribute
oid:1.2
name: Author
value: Ullman

Native XML Storage

- Verbatim files
  - Appropriate for small documents, grep-style querying
- Natix (University of Mannheim, Germany)
  - Hybrid: verbatim files + page-level storage
  - Semantically partition large document into subtrees based on tree structure
  - Store each subtree in one record (unit of storage) that is atomic
  - Proxy nodes are used to connect subtrees in different records
  - Primitives for read/write/insert/delete of element
  - Record size need not be statically configured, can be a dynamic value; adapting to the size and structure of document at runtime
  - Reconstruction of original tree by replacing proxies by subtrees
  - Core of XML storage system
  - No explicit use of DTDs or XML schema
  - Xyleme uses Natix as underlying storage manager
  - No query language support

Commercial Databases

- IBM DB2 XML Extender
  - Pure relational mapping
    - Decomposition of XML and mapping into relational tables
  - Mixed content
    - CLOBs (Character Large Objects) + side tables for indexing structured data embedded in text
- Oracle 9i
  - Canonical mapping into user-defined object-relational tables
  - Stores XML documents in CLOBs
- MS SQL Server
  - Generic Edge technique with inlined scalar values
  - Text content modeled in CLOBs

XML Query Language: Requirements

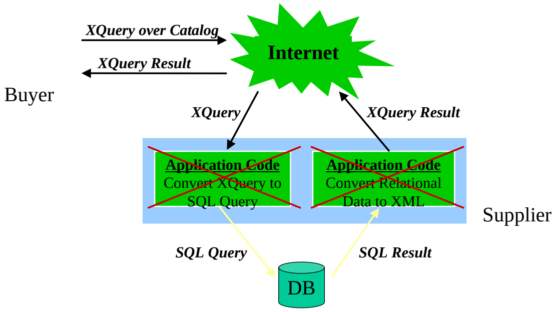
- Expressive power
  - Should support all relational algebraic operators
  - Restructuring operations – reduction, merge, …
- Formal Semantics
  - Important for dealing with query transformation and optimization
- Output delivery Mode
  - The output of a query should be (at least) in the same language as the input
- Query Languages: Xquery, XML-QL, YATL, Lorel, WebSQL

XML Query Over Relational Data

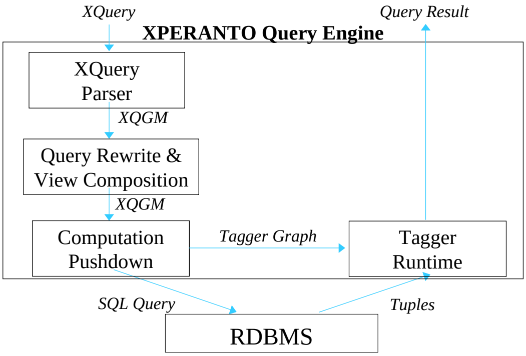
- Most web data will continue to be stored in relational databases (more than 90%)
  - Need some way to execute XML query over relational data and then convert the results into XML data
- XPERANTO (IBM) allows existing relational data to be *viewed* and *queried* as XML.

Web Services Example

Supplier provides an XML View of its Data



XPERANTO; High Level Architecture



XQGM

- Intermediate representation：
  - General enough to capture semantics of a powerful language such as XQuery
  - Easy translation to SQL
- XQGM based on DB2’s QGM and XML Algebra
- XQGM consists of:
  - Operators
  - Functions (invoked inside operators)
- Functions capture manipulation of XML entities (elements, attributes, etc.)
  - XML construction functions
  - XML navigation functions

Data Stream

- A data stream is a sequence of data items X_1, X_2, …, X_n, coming continuously from single or multiple sources where random access to data is not allowed.

Data Stream Characteristics

Strongly regular: strongly periodic (inclusive zero time interval between two data items), only one type of data, schema can be derived or conforms schema.

Weakly regular: weakly periodic (follows some time interval), mixed types of data but follows the order, schema can be derived.

Irregular: aperiodic, types of data unknown, no order, schema cannot be derived.

DBMS vs. DSMS

- Traditional DBMS
  - data stored in finite, persistent data sets
  - assumes “one-time” query against data
  - focus on precise answer computed by stable query plans
- Data Stream Management System (DSMS)
  - Allow some or all of the data being managed to come in the form of continuous, possibly very rapid, time varying, ordered data streams
  - Queries may be continuous (not just one-time)
    - Evaluated continuously as stream data arrives
    - Answer updated over time
  - Key ingredient in executing queries is Approximation
  - Main memory computations
  - DSMS = merely DBMS with enhanced support for triggers, temporal constructs, data rate management?

Weakly Regular or Irregular Data Streams: Issues

- Schema discovery and evolution
- Filtering data interest to applications
- Unbounded memory requirements
  - Materialization of Views
- Approximate Query Answering
  - Techniques for data reduction and synopsis construction
    - random sampling, histograms, sliding windows, etc
- **Online processing**
  - Many data streams applications need online processing
    - E.g., detecting denial-of-service attacks, detecting Service-Level Agreement violations, admission control and traffic policing, etc
  - Offline processing is indeed appropriate for some applications
    - E.g., capacity planning, determining pricing plans

Active functionalities over streaming data

- Provides real-time functionalities that is needed in several advanced applications.
  - Alert a doctor when the blood pressure of a patient goes below X, heart beats less than Y and ECG touches Z.
  - Sell all my INTC stocks at the higher trading price exchange if the price difference at any time between two exchanges is more than 2%.
  - Cancel my tomorrow's flight if there is a terrorists attack in the region of flying.
- Events can be defined on composition of data streams that can trigger some pre-defined actions (notification and alert, database change, etc.)
- Context can be associated with the events
  - INTC was trading higher at NASDAQ at 9:32 AM since CEO of INTC rang the opening bell.

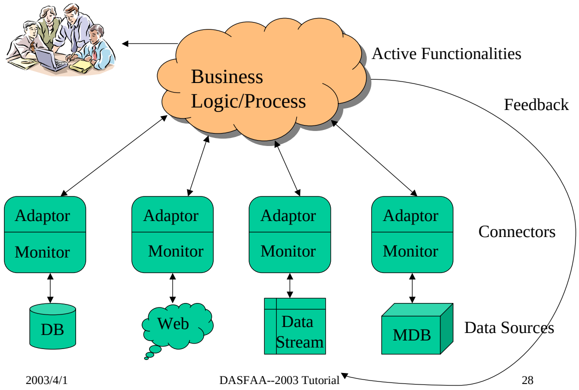
Event Based (Active) Information Integration

- On-demand integration
- Dissemination of selective information
- Tuned to change in business processes
- Autonomic computing
- Major shift in Industry

Products: Crossworlds, WMQI, MQWF, BEA WebLogic Integrator Integrator, MS BizTalk, Web Methods Enterprise

These products solve some aspects of event based integration of applications/data.

Architecture



Active Rules

An active rule is composed of three components:

Event (E): Monitor - Detect - Evaluate
Condition (C): Derive - Analyze - Evaluate
Action (A): Collaborate - Integrate - Effect

Monitoring Events

- Many underlying operational systems do not have the capability of defining triggers or publish events.
- Sometimes the owner does not want the operations systems to be touched since they are executing thousands of transactions and no change, of whatsoever, is allowed in application or anywhere in these systems.

The question is: how to monitor or sense the changes (change detection) in the operational systems which may trigger to flow the information across underlying systems for integrating them?

Polling

- Design a set of queries that are executed periodically.
- Compare the results of the same query with the previous materialized results of the same query. Find any change occurred in underlying operational system.
- If there is any change, determine whether the change is related to the registered event or not.

- Issues
  - Materialization of previous results (up to what degree?)
  - Not all changes can be monitored by querying
  - Design of optimized queries for change detection
  - Frequency of querying

Semantic Web

'Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling Computers and people to work in cooperation.'

Source: Time Berners-Lee, James Hendler and Ora Lassila, 'Semantic Web', Scientific American, May 2001

Semantics

- `meaning or relationship of meanings, or relating to meaning' (Webster)
- is concerned with the relationship between the linguistic symbols and their meaning or real-world objects
- meaning and use of data (Information System).

Importance:

- Effective use of web information
- To make information context sensitive
- Derive new information or topic based history
- Support new services for e-business, e-gov etc.

Semantic Web

- Semantic Web: Data + Metadata +URI …….
  - Metadata: Labeling and structuring information in a document
  - URI (Universal Resource Identifier): an universal and unique name for any resource
  - provides intelligent content

- Issues
  - How to annotate documents?
  - Building annotators for each vertical application?
  - Design and evolution of rich ontology
  - Categorize unstructured text
  - Automatically create tags based on tags itself
  - Personalization/Notifications/Alerts

Ontology

- An ontology is a specification of conceptualization.
- Standardizes meaning, description, representation of involved concepts/terms/attributes
- Captures the semantics involved via domain characteristics, resulting in semantic metadata
- 'Ontological commitment' forms basis for knowledge sharing and reuse
- Examples: WorldNet, Cyc, MeSH (Medical Subject Headings), Uncefact (product classification)

Ontology Languages

- Ontology languages are semantic markup languages,
- DAML: DARPA Agent Markup Language
- OWL: Web Ontology Language is the successor of DAML + OIL (Ontology Inference Layer), currently developed by W3C web ontology group, and based on RDF ideas.

Open Directory Project (ODP): Classification/Taxonomy & Directory (www.dmoz.org)

Ontology Definition

- The body of the ontology consists of
  - Classes
  - Properties
  - Instances (for use in class definition)
- The main component of an ontology is a taxonomy (a class hierarchy)

Applications

- Designing a scrap book on web
  - Topic based "copy and paste of information" in a logical order
  - Finding relationships between documents
  - Making your own web world

- Creation of a Web space abstraction
  - Classification of documents
  - Annotating these documents
  - Report/History Generation
  - Monitoring the changes
  - Maintenance of web space abstraction

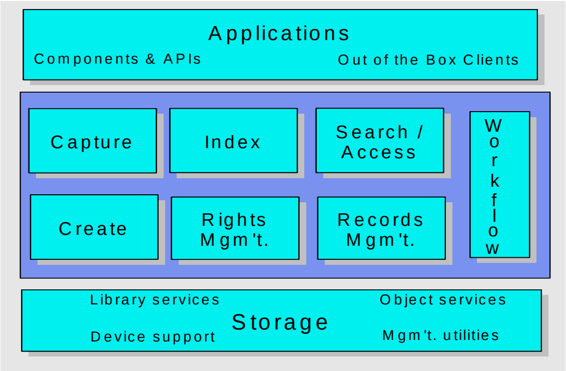
Managing Unstructured Data: IBM Content Manager (CM)

- provides a formal mechanism for creation, maintenance and distribution of information (including unstructured content) within an enterprise
- supports version control, lifecycle management, searching and taxonomy (hierarchical classification of content) of documents
- efficient management of content and document routing capabilities (Workflow)
- supports variety of new data types for text documents, static images, video clips, audio files, and many more.

Content: Issues

- **Paper overwhelms the workspace**
- **No concurrent access; one user at a time**
- **Easy to lose or miss-file**
- **Security is poor**
- **Hard to find folder / document when needed**
- **Hard to find digital assets to reuse them**
- **Video and audio don't fit in a folder**
- **Workstation footprint not enough to hold large Video or voice files**
- **No Table Of Contents for folders**
- **Can't use automated search**
- **Costs to manage and distribute files**
- **PC files are stored in disparate servers, copies made and filed**
- **Documents not immediately available, leads to poor customer service**
- **Workflow means "pick up and move the folder"**
- **No cross enterprise folder of your entire customer relationship**
- **If it's not electronic, can't access over web - Can't do e-business**
- **Need ability to repurpose content (Web Publishing)**
- **Need Common infrastructure for ECM (Develop specific clients)**

High Level Architecture of CM



References

- Phil Bohannon, Juliana Freire, Prasan Roy, Jérôme Siméon, **From XML Schema to Relations: A cost-based Approach to XML Storage**, ICDE 2002
- Michael J. Carey,Jerry Kiernan, Jayavel Shanmugasundaram, Eugene J. Shekita, Subbu N. Subramanian, **XPERANTO: Middleware for Publishing Object-Relational Data as XML Documents**, VLDB 2000
- Daniela Florescu, Donald Kossman, **A Performance Evaluation of Alternative Mapping Schemes for Storing XML Data in a Relational Database**, IEEE Data Eng. Bulletin 1999
- P.J. Marron, G. Lausen, **On Processing XML in LDAP**, VLDB 2001
- Carl-Christian Kanne, Guido Moerkotte, **Efficient Storage of XML Data**, Technical Report 8/99, University of Mannheim, 1999
- Feng Tian, David J. DeWitt, Jianjun Chen, and Chun Zhang, **The Design and Performance Evaluation of Various XML Storage Strategies**, Technical report, University of Wisconsin
- W3C XML representation of a relational database In http://www.w3.org/XML/RDB.html
- W3C Recommendation. **Extensible Markup Language (XML) 1.0** (Second Edition) In http://www.w3.org/TR/REC-xml
- Sihem Amer-Yahia, and Mary Fernandez, **Techniques for Storing XML**, ICDE tutorial, 2002.

References (contd…)

- Carl-Christian Kanne, **Natix: A Native XML Base Management System,** Ph.D. Thesis, University of Mannheim, Germany, 2002
- A. Bonifati and S. Ceri, **Comparative analysis of five XML query languages**, SIGMOD Record, March 2000.
- Gregory Cohena, Serge Abiteboul and Amelie, **Detecting Changes in XML Documents**, ICDE 2002
- Sourav Bhowmick, Sanjay Kumar Madria, Wee Keong Ng, Ee-Peng Lim, **Detecting and Representing Relevant Web Deltas using Web Join**, ICDCS 2000
- B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, **Models and Issues in Data Stream Systems**, PODS 2002