Semistructured Data and XML

How the Web is Today?

- HTML documents
- often generated by applications
- consumed by humans only
- easy access: from any server, across platforms, across organizations

Ref: (1) DASFAA'2001 tutorial notes by Dan Suciu
(2) Data on the Web – From Relations to Semistructured Data and XML, Serge Abiteboul, Peter Buneman, Dan Suciu, Morgan Kaufmann, 2000

Limits of the Web Today

- HTML not understood by applications, applications cannot consume HTML
- HTML wrapper technology is brittle
  - screen scraping brittle
  - Wrapper will not work if web page format changed
- Database technology: client-server
  - Still vendor specific
- OO technology (Corba : **C**ommon **O**bject **R**equest **B**roker **A**rchitecture) requires controlled environment.

  Note: The Common Object Request Broker Architecture (CORBA) is a standard developed by the Object Management Group (OMG) to provide interoperability among distributed objects. CORBA is the world's leading middleware solution enabling the exchange of information, independent of hardware platforms, programming languages, and operating systems. CORBA is essentially a design specification for an Object Request Broker (ORB), where an ORB provides the mechanism required for distributed objects to communicate with one another, whether locally or on remote devices, written in different languages, or at different locations on a network.
- companies merge, form partnerships; need interoperability fast

Paradigm Shift on the Web

- From documents (HTML) to data (XML)
- From information retrieval to data management
- From relational model to semistructured data model
- From storage to transport
- new Web standard XML from W3C
  - XML = data
  - XML generated by applications
  - XML consumed by applications
- data exchange
  - across platforms: enterprise interoperability
  - across enterprises

Web: from collection of documents to data and documents

But Needs a Paradigm Shift Too

- Web data differs from (relational) database data:
  - self-describing, schema-less
  - structure changes without notice
  - heterogeneous, deeply nested, irregular
  - documents and data mixed together
- designed by document experts, not database experts
- need Web data management

Semistructured Data

Origins:

- integration of heterogeneous sources
- data sources with non-rigid structure
- biological data
- *Web data*

Examples of Semi-Structured Data

name: Peter Chen
email: pchen@lsu.edu, chen@bit.csc.lsu.edu

name:
    first name:Elisa
    last name:Bertino
email: bertino@cs.purdue.edu

name: Phil Bernstein
affiliation: Microsoft Research

The Semistructured Data Model

- **TSIMMIS** - The Stanford-IBM Manager of Multiple Information Sources - is a system for integrating information. (1994-1997)
- The **goal** of the TSIMMIS Project is to develop tools that facilitate the rapid integration of heterogeneous information sources that may include both structured and semistructured data. TSIMMIS has components that:
  - translate queries and information (source wrappers);
  - extract data from World Wide Web sites;
  - combine information from several sources (mediator);
  - allow browsing of data sources over the Web.

The Semistructured Data Model (cont.)

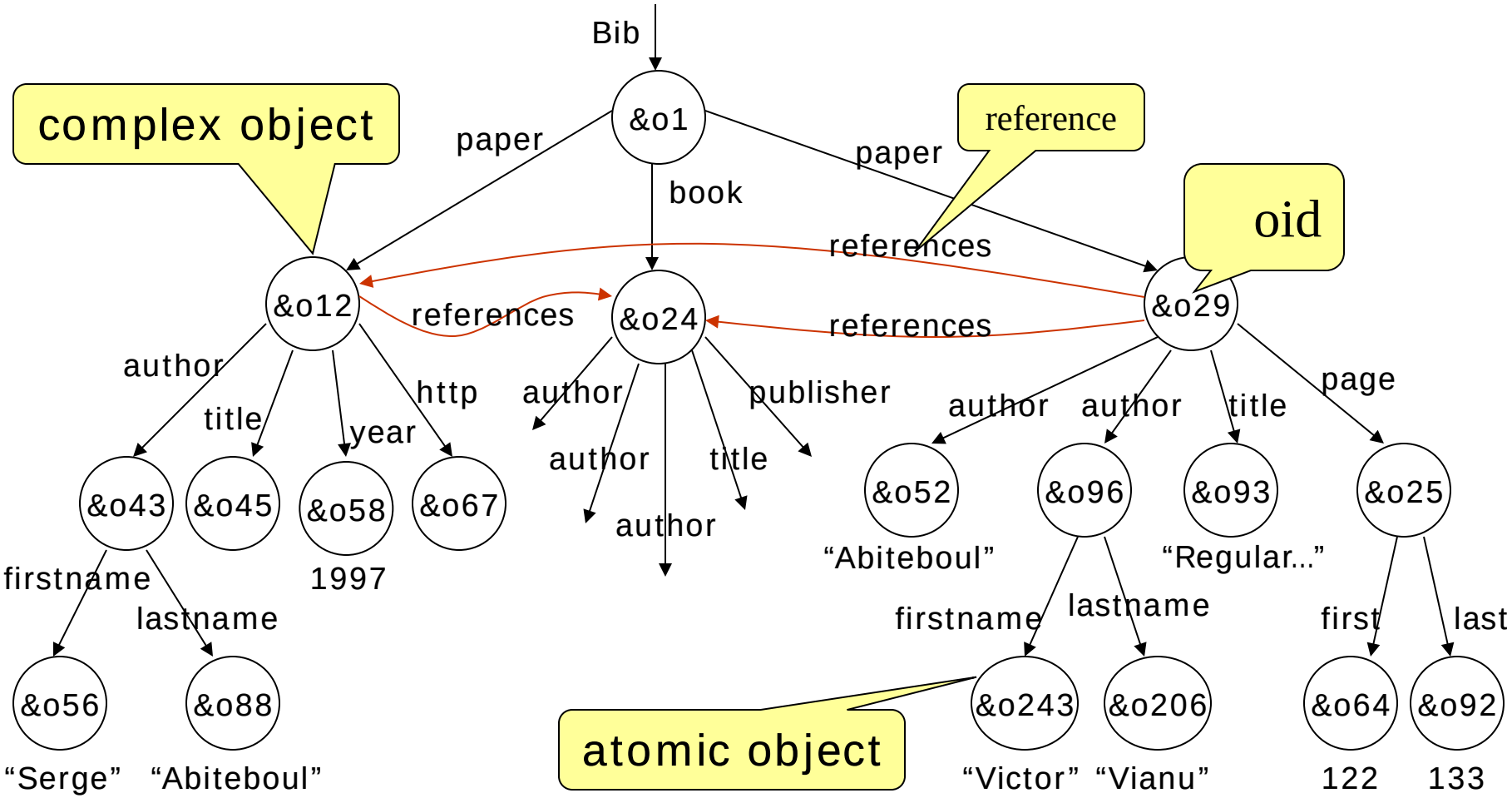
- They also developed a system called **LORE** (for Lightweight Object REpository), and a query language called **LOREL**, aimed specially at handling semistructured data.
- The data model used in Lore is a lightweight object model called **OEM** (for Object Exchange Model). It is one standard to express semi-structured data, another way is XML.
- OEM does not require strong typing of its objects and is flexible in other ways that address desideratum.

  It is a simple, self-describing model with object nesting and identity.

The Semistructured Data Model (cont.)

- Lore is primarily for storing and querying data obtained from other information sources.
- Lore itself also is lightweight, it is a repository and a query engine but not a full-feature database management system.
- Lore does not provide transaction management, concurrency control, or recovery.
- Lorel, the query language supported by Lore, is a compatible extension to the OQL object-oriented query language, with new features designed specially for querying semistructured data.
- They have migrated Lore to fully support XML; see "From Semistructured Data to XML: Migrating the Lore Data Model and Language", (1999), http://infolab.stanford.edu/lore/pubs/data.html#XML

The Semistructured Data Model (cont.)



Object Exchange Model (OEM)

Syntax for Semistructured Data

```
Bib: &o1 { paper: &o12 { … },
           book:  &o24 { … },
           paper: &o29
             { author: &o52 “Abiteboul”,
               author: &o96 { firstname: &o243 “Victor”,
                              lastname:  &o206 “Vianu” },
               title: &o93 “Regular path queries with constraints”,
               references: &o12,
               references: &o24,
               page: &o25 { first: &o64 122, last: &o92 133 }
             }
         }
```

Problem: Who should assign the node oid values? How to find the oid value of an object in order to use it as an IDREF(S) attribute value? (see DTD lecture notes, similar to foreign key in relational databases)

Syntax for Semistructured Data

May omit oid's:

```
{ paper: { author: "Abiteboul",
           author: { firstname: "Victor",
                     lastname: "Vianu" },
           title: "Regular path queries ...",
           page: { first: 122, last: 133 }
         }
}
```

Problem: Don't have references. How to implement IDREF(S) this way?

Characteristics of Semistructured Data

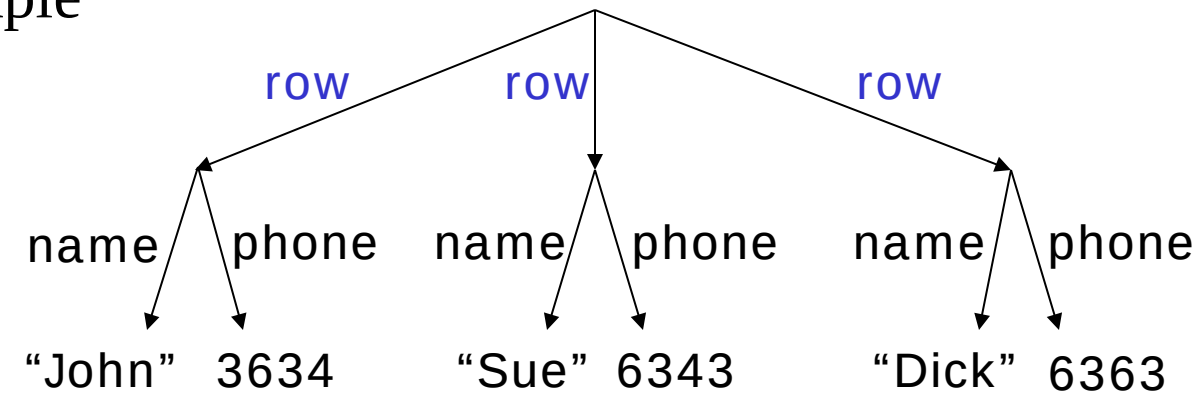
- missing or additional attributes
- multiple attributes
- different types in different objects
  - E.g. name or {first_name, last_name}
- heterogeneous collections
- It provides a flexible format for data exchange between different types of databases.
- It can represent the information of some data sources that cannot be constrained by schema
- The schema if any can easily be changed.

Self-describing, irregular data, no a priori structure

Comparison with Relational Data

A simple table example

name	phone
John	3634
Sue	6343
Dick	6363




```
{ row: { name: “John”, phone: 3634},
  row: { name: “Sue”,  phone: 6343},
  row: { name: “Dick”, phone: 6363},
}
```

Semistructured data has hierarchical structure but relational data is flat.