Outline

- **Information Integration Problem**
- Heterogeneous Information Resources Integration
- Analyzed Integration Systems
- Important Integration Principles and Comparison Criteria
Data Integration Problem

- The current period of IT development is characterized by an explosive process of information models creation.
- **Distributed infrastructures**: OMG, Semantic Web, SOA, digital libraries, information grid, …
- **Information models**: data models, workflow models, process service composition models, semantic models
- **Accumulation** of based on such models information resources, the **number** of which **grows exponentially**

- Data Integration Projects World-Wide (Patrick Ziegler, 2007) – 183 projects
Data Integration: Example

- Annotators extract key information from email messages. This information is used to probe the relational source data to retrieve additional facts needed for the target schema.

**Source Data and Schema**

<table>
<thead>
<tr>
<th>DEALERS</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dealership</td>
<td>DealerID</td>
<td>Size of Dealership</td>
<td>Owner</td>
<td>Annual Revenue</td>
</tr>
<tr>
<td>Oshkosh Billboy Ford</td>
<td>bb32</td>
<td>300 cars per month</td>
<td>Bill Boy</td>
<td>$5M</td>
</tr>
</tbody>
</table>

| TRANSACTIONS | | | | | | |
|--------------|---------|----------------|----------|---------|--------|
| TransID | Customer | DealerID | Auto Model | Date Sold | Price |
| 1234 | Jutt, John J. | bb32 | Galaxy | 4/21/07 | 5000 |

**Target Schema**

<table>
<thead>
<tr>
<th>Auto Model</th>
<th>Customer</th>
<th>Dealership</th>
<th>Date Sold</th>
<th>Price</th>
<th>Size of Dealership</th>
<th>Problem Date</th>
<th>Problem Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galaxy</td>
<td>John J. Jutt</td>
<td>Oshkosh Billboy Ford</td>
<td>4/21/07</td>
<td>5000</td>
<td>300 cars per month</td>
<td>Oct. 28, 2007</td>
<td>brakes are squealing</td>
</tr>
</tbody>
</table>
Types of Information Integration Systems

- Data warehousing
- Virtual Data Integration
- Message Mapping
- Object Relational Mapping
- Document Management
- Portal Management
Data warehousing

- Data warehouse – database that consolidates data from multiple sources
- Each resource may have a DB schema that differs from the warehouse schema. So data has to be reshaped into common warehouse schema
- Extract-Transform-Load (ETL) tools
  - cleansing operations
  - reshaping operations
Virtual Data Integration

- Gives the illusion that data sources have been integrated without materializing data
- Offers a mediated schema against which users can pose queries
- The implementation, often called a query mediator system, translates the user’s query into queries over the data sources and integrates the result of those queries so that it appears to have come from a single integrated database
- Resources are heterogeneous in that they may use different database systems and structure the data using different schemas
Message Mapping

- **Message-oriented middleware** helps integrate independently developed applications by **moving messages** between them.

- If messages pass through a broker, the product is usually called an **enterprise application integration** (EAI) system.

- If a broker is avoided through all applications’ use of the same protocol, then the product is called an **enterprise service bus**.

- If the focus is on defining and controlling the order in which each application is invoked, then the product is called a **workflow system**.
Enterprise Service Bus
Object Relational Mapping

- Application **programs** today are typically written in an **object-oriented** language, but the **data** they access is usually stored in a **relational** database.
- Mapping applications to databases requires **integration** of the **relational and application schemas**.
- **Differences** in **schema constructs** can make the mapping rather **complicated**.
- **Object-to-relational mapper** offers a high-level **language** in which to **define mappings**.
- Resulting **mappings** are then **compiled** into **programs** that translate **queries and updates** over the object-oriented interface into queries and updates on the relational database.
Document Management

- Much of the information is contained in documents.
- To promote collaboration and avoid duplicated work in a large organization, this information needs to be integrated and published.
- Integration may simply involve making the documents available or integration may mean combining information from these documents into a new document.
- In some applications, it is useful to extract structured information from documents. The ability to extract structured information of this kind may also allow businesses to integrate unstructured documents.
Portal Management

- One way to **integrate** related information is simply to **present** it all, side-by-side, **on the same screen**
- A **portal** is an **type of integration in mind**
- Portal design requires a mixture of
  - content management (to deal with documents and databases) and
  - user interaction technology (to present the information in useful and attractive ways)
Outline

- Information Integration Problem
- **Heterogeneous Information Resources Integration**
- Analyzed Integration Systems
- Important Integration Principles and Comparison Criteria
Heterogeneous Information Resources Integration

- Information **Resource driven** approach
  - moving from sources to problems (an integrated schema of multiple sources is created independently of a definition of specific application)
  - is not scalable with respect to the number of sources
  - does not make semantic integration of sources in a context of specific application possible
  - does not lead to justifiable identification of sources relevant to specific problem,
  - does not provide the required information system stability w.r.t. evolution of the observation sources (e.g., appearance of a new information source relevant to the problem lead to reconsideration of the integrated schema)
Heterogeneous Information Resources Integration (2)

- **Problem driven** approach
  - moving from a problem to the sources (a description of an application subject domain (in terms of concepts, data structures, functions, processes) is created, into which sources relevant to the application are mapped)
  
- assumes creation of subject mediator that supports an interaction between an application and sources on the basis of the application subject domain definition

- removes the disadvantages mentioned for the approach driven by information sources
Integration Using Views

- **Global As View (GAV)**
  - According to GAV a global schema is defined in terms of the pre-selected sources

- **Local As View (LAV)**
  - Sources are defined as views over the mediator schema

- **Both As View (BAV)**
  - Based on the use of reversible schema transformation sequences. LAV and GAV view definitions can be fully driven from BAV

- **GLAV**
  - Later a variation of LAV allowing the head of the LAV view definition rules to contain any source schemas query and hence is able to express the case where a source schemas are used to define the global schema constructs (GAV)
GAV Example

Global Schema:
- MasterStudent(studentName)
- University(uniName)

Local Schemas:
- S1.Catalogue(nomUniv, programme) - catalog of teaching programs
- S2.Erasmus(student, course, univ) - Erasmus exchange program
- S3.CampusFr(student, program, university) - programs of French universities
- S4.Mundus(program, course) - international master programs

\[\text{MasterStudent}(N) \supseteq S2.Erasmus(N,C,U), S4.Mundus(P,C)\]
\[\text{MasterStudent}(N) \supseteq S3.CampusFr(N,P,U), S4.Mundus(P,C)\]
\[\text{University}(U) \supseteq S1.Catalogue(U,P)\]
\[\text{University}(U) \supseteq S2.Erasmus(N,C,U)\]
\[\text{University}(U) \supseteq S3.CampusFr(N,P,U)\]
S2.Erasmus(S,C,U) ⊆ Student(S), EnrolledInCourse(S, C),
PartOf(C, P), OfferedBy(P, U), EuropeanUniversity(U),
EuropeanUniversity(U’), RegisteredTo(S,U’),
U ≠ U’
Outline

- Information Integration Problem
- Heterogeneous Information Resources Integration
- **Analyzed Information Integration Systems**
- Important Integration Principles and Comparison Criteria
Information Integration Systems

- Agora
- AutoMed
- Infomaster
- PICSEL
- SIRUP
- Information Manifold
- MedMaker
- SYNTHESIS
- Karma
- MetaMed
Agora (2000)

- **Approach:** LAV
- **Canonical model:** XML
- **Query language:** Xquery
- **Resources:** XML, Relational

On the basis of *Le Select* DI system
Agora (II)

- External DM is XML, internal DM is relational
- XML data sources are modeled by a generic relational schema (tables `Element`, `ElemContent`, `Attribute`, `ElemAttribute`, …)
- *Quilt* queries are translated into SQL
- Keyword search is used when the particular structure of the documents is not known (or partially known)
  - Helps novice users to browse the information content available
  - Used as a filter for structured queries

- **Approach**: BAV
- **Canonical model**: HDM (hypergraph data model)
- **Query language**: AIQL
- **Resources**: Relational, XML, flat files
Model definitions repository (MDR) defines how a data modeling language is represented as combinations of nodes, edges and constraints in the HDM

Schema transformation repository (STR) defines schemas in terms of the data modeling concepts in the MDR, and transformations to be specified between those schemas

Schema integration in relies on the reuse of specific sequences of primitive transformations

- schema and data independent parameterized template transformations

Schema matching - discovering semantic relationships between schema objects

- equivalence, subsumption, intersection, disjointness, incompatibility
- automatic discovery of the relationships by performing a bidirectional comparison of schema objects

YATTA (YAT for transformation-based approach) is a variation of the YAT model to support semistructured data
Infomaster (1997)

- **Approach**: LAV
- **Canonical model**: KIF
- **Query language**: KQML
- **Resources**: Relational, Z39.50, custom pages
SIRUP (2004)

- **Approach:** LAV
- **Canonical model:** ICONCEPT
- **Query language:** SQL-like
- **Resources:** Relational, XML, ontology

- **Approach**: GAV
- **Canonical model**: Object Echange Model (OEM)
- **Query language**: MSL (Mediator Specification Language)
- **Resources**: Relational, Semi-Structured
Information Manifold (1996)

- **Approach:** LAV
- **Canonical model:** CARIN-Classic
- **Query language:** Datalog-like
- **Resources:** XML, Relational, semi-structured, …
**Approach:**
LAV

**Canonical model:**
CARIN

**Query language:**
CARIN (Datalog-like)

**Resources:**
Services
SYNTHESIS (2006)

- **Approach:** LAV
- **Canonical model:** SYNTHESIS
- **Query language:** Syfs

- **Resources:**
  - XML, services, Relational, Object-Relational, e.t.c.
Karma (2015)

- Exploiting semantics to solve the problem of big data variety
  - Clean and normalize data
  - Build a model or semantic description of each source

**Approach**: keyword search + data exchange

**Canonical model**
- Data formats are converted into *nested relational model*
- Fields of a data set are mapped into classes and properties of common *ontology*
- Data is converted into *RDF* using the ontology

**Query language**
- keyword search for virtual integration
- SPARQL for materialized RDF data

**Resources**: spreadsheets, relational databases, web services
- Formats: CSV, JSON, XML
Using *integration operators* for generating multiple *user defined mediator views* above the grid participating ISs

- **Approach**: GAV
- **Canonical model**: MGKM - RDF/OWL-based domain model ontology
- **Query language**
  - User language – SPARQL
  - Internal query language - OQLI3
- **Resources**: relational databases through JDBC/ODBC
Outline

- Information Integration Problem
- Heterogeneous Information Resources Integration
- Analyzed Information Integration Systems
- **Important Integration Principles and Comparison Criteria**
Important Integration Principles

- ASME Criteria
  - Abstraction
  - Selection
  - Modeling
  - Explicit Semantic

- Principles
  - Integration **Approach**
  - **Extensible** Canonical Informational Model
  - Semantic Schema Matching
  - Problem solving specification
ASME Criteria

- **Abstraction** refers to shielding users from low-level heterogeneities and underlying data sources
- **Selection** means the possibility of user-specific selection of data and data sources for individual integration
- **Modeling** corresponds to the availability of means to incorporate user-specific ways to perceive a domain of interest for which integrated data is desired in the process of data integration
- **Explicit semantics** refers to means for explicitly representing the real-world semantics of data
Integration Principles

**Integration Approach**
- LAV removes the disadvantages of GAV
- Abstraction + Modeling = Approach (LAV, GAV, …)
- Criterion – **Approach** (“A”)

**Extensible Canonical Informational Model**
- Resources are **heterogeneous**, so the **unification of resources** models in the frame of some **unifying information model** called **canonical** is required
- **Unification requires** a technique of **matching the specifications** of various resources
- **Refinement relation**: It is said that specification A refines specification D, if it is possible to use A instead of D so that the user of D does not notice this substitution
- Criterion – **Unification** (“U”)
- Criterion – **Selection** (“S”)

37
Integration Principles (2)

- **Semantic Schema Matching**
  - Resource Registration require metadata (ontology)
  - Criterion – **Explicit Semantic** (“E”)

- **Problem solving specification**
  - Application domain specification includes: concepts, data structures, **functions**, processes
  - Criterion – **Functionality** (“F”)
  - Architecture Extensibility (integration into existing problem solving infrastructures – web-services, semantic web, grid-infrastructures, virtual observatories etc.; hybrid infrastructures creation)
  - Criterion – **Hybrid** (“H”)
  - User Friendly Integration Tools Availability (for application domain experts – astronomers, biologists, …)
  - Criterion – **Tools** (“T”)
Comparison Criteria

- **AUSEFHT**
  - Approach
  - Unification
  - Selection
  - Explicit Semantic
  - Functionality
  - Hybrid
  - Tools
## Comparison Results

<table>
<thead>
<tr>
<th>System</th>
<th>A</th>
<th>U</th>
<th>S</th>
<th>E</th>
<th>F</th>
<th>H</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agora</td>
<td>LAV</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>AutoMed</td>
<td>BAV</td>
<td>Yes</td>
<td>No</td>
<td>Partially</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Infomaster</td>
<td>LAV</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>SYNTHESIS</td>
<td>GLAV</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>PICSEL</td>
<td>LAV</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SIRUP</td>
<td>LAV</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Information Manifold</td>
<td>LAV</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>MedMaker</td>
<td>GAV</td>
<td>No</td>
<td>Yes</td>
<td>Partially</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Explicit Semantic Criteria: Comments

- Agora (No)
  - resource schemas are defined

- AutoMed (Partially)
  - BAV transformations contain more information than LAV or GAV views (semantically reachier)
  - Schema matching is provided, semantic relationships between schema objects based on the comparison of their intentional domains (sets of real-world entities represented by the schema objects) are defined
    - Similarities are established on the basis of element names, data types, numerical and non-numerical statistics, Naïve Bayes classification of instances

- Infomaster (No)
  - resource schemas are defined

- SYNTHESIS (Yes)
  - semantics is defined using ontologies
  - integration is checked using DL containment and ADT refinement

- PICSEL (Yes ?)
  - Service specifications are used to enrich initial ontology

- SIRUP (Yes)
  - semantics is defined using ontologies

- Information Manifold (No)
  - resources schemas and capabilities are defined

- MedMaker (Partially ?)
  - MSL and OEM can be seen as a form of first order logic
  - source schema can be retrieved