Motivation

- Interchange data among applications
  - Cooperate, interact, ...
  - Fundamental for e-Commerce apps
- 1/2/02, how can I interpret this?

Motivation

- Interchange data among applications
  - Cooperate, interact, ...
  - Fundamental for e-Commerce apps.
- 1/2/02
- 1234,00
- 0.5
- 23
- calculate(21000, 3)

Contextual information is required
Fundamental Semantic Heterogeneity

- Instance/entity identification problem
  - When two different information sources store information on an identical object, but do not share enough common information attributes to identify the object as the same
  - It cannot be solved algorithmically

Fundamental Semantic Heterogeneity (2)

- Two descriptions in different data sources have a similar meaning, but are quite the same
  - Neither data source (and schema) contains enough information to resolve the problem
    - E.g. prices in one DB include sales tax, while in a different DB do not
  - If data sources do not contain information to clarify the discrepancy, it cannot be resolved programmatically

Structural Semantic Heterogeneity

- When the same information is represented in two separate Apps in structurally different but formally equivalent ways. Two main reasons:
  - Normally, consequence of independent creation, design and evolution of autonomous Apps
  - Other reason, rich set of modeling constructors
- Spectrum of heterogeneity
  - Domain conflicts
  - Naming conflicts
  - Type conflicts
  - Structural conflicts

Domain conflicts

- Metadata specification differs and consequently conceptual schemas are different
  - The same entity is described differently in different domains
  - E.g. Paul is known as p123 in domain A and paul in domain B

Naming conflicts

- Objects may be represented in a different manner
  - The same attribute has different labels
  - E.g. Attribute name versus lastname

Type conflicts

- Systems represent low level atomic values differently
  - Different types are used to describe related entities
  - E.g. temperature may be of type integer in one system and of type float in another
### Structural conflicts
- Data may be managed by different DBMSs
- A different data organization or structure is used to represent the same concepts
- E.g. An address type is represented as a structure or as a single attribute of type string

### Overcoming Semantic Heterogeneity
- These heterogeneities can be overcome when
  - there is enough information in the metadata to clarify the meaning of each of the objects within the data source
- If it's not the case
  - Resolving these conflicts become difficult
  - and sometimes impossible

### Overcoming: Domain conflicts
- By using semantic dictionaries
  - Must contain mapping between domains
    - Dictionary entry with
      - paul identifies p123 in domain A
      - p123 identifies paul in domain B
  - When objects cannot be mapped 1:1 across domains, a more complicated mapping mechanism must be used
    - This can result in information loss
    - E.g. Representation of grades and marks
      - Out of ten and A, B, C, ... +, -

### Overcoming: Naming conflicts
- Require mapping functions which simply change the labels of the attributes
  - Easy to overcome
  - A naming conflict can be mistaken for a structural conflict
  - E.g. An attribute price may or may not include taxes

### Overcoming: Type conflicts
- Usually can be resolved by applying conversion functions
- E.g. Programming languages provide functions for converting strings to integers

### Overcoming: Structural confl.
- Involves decomposition or composition
  - E.g. The address type
    - represented in a structure composed into a single attribute
    - represented in a single string decomposed into a structure
Overcoming Semantic Heter.

- Ambiguity of data
  - Looking at data and metadata is sometimes not enough to fully understand its meaning
  - Completely removing ambiguity is unrealistic
- Semantic values
  - Facilitate integration of heterogeneous data
  - Help resolve many ambiguities in data

Data Integration - Issues

- Data from different sources/components is represented differently
- Different organizations/departments use different units and representation formats
- Many of the underlying assumptions about the meaning of a given data object are only implicit
- Context information is left implicit and consequently it is lost when crossing institutional boundaries

Why Semantic Metadata?

- The Internet as a global marketplace
- Business-to-Consumer:
  - interactive “point-and-click”
- Business-to-Business:
  - proprietary protocols
- Business-to-Business-to-Consumer:
  - need to extract and consolidate data for further electronic processing

Example – Travel Agency

Availability for FRANKFURT (FRA) to KENNEDY–NEW YORK (JFK)
Saturday, June 06 2009

<table>
<thead>
<tr>
<th>Flight</th>
<th>Departing City</th>
<th>Departing Time</th>
<th>Arriving City</th>
<th>Arriving Time</th>
<th>Seats</th>
<th>Mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF</td>
<td>FRA</td>
<td>10:30 AM</td>
<td>JFK</td>
<td>1:00 PM</td>
<td>0</td>
<td>S/L</td>
</tr>
</tbody>
</table>

Price Per Adult (Economy Class): DEM 182.6

Direct flight on Saturday June 6, 2009
Departing: FRA Frankfurt, Frankfurt Germany
Lufthansa, flight number 600, departing 10:30 AM, arriving 1:00 PM
Class: Y – Economy coach, Flight Dist: 3800 Miles
You can reserve this/these flight(s) at a fare of $ 180 for one adult, incl. taxes.
The MIX Model

- **MIX**: "Metadata based Integration model for data X-change"
- Combines two aspects:
  - representation of data plus additional semantic metadata
  - flexible, self-describing data model for the representation of semi-structured data

Ontology

An ontology is a formal specification of a shared conceptualization of a domain of interest.

Ontologies in MIX

- Set of concepts and their relationships that model a given domain
- Ontology concept:
  - is an abstraction of a set of real world phenomena
  - has a representation type associated that determines the physical representation of data of a given concept
- Should be based on existing standards, but must be extensible

Simple Semantic Object

- A data item with additional metadata to support its interpretation:
  
  \[
  \langle \text{Distance}, 3850, \{\langle \text{Unit}, \text{"mile"}, \langle \text{Scale}, 1 \rangle \} \rangle
  \]
  - 3850 is the recorded data value
  - Distance denotes the ontology concept
  - \{\langle \text{Unit}, \text{"mile"}, \langle \text{Scale}, 1 \rangle \} \} represents the interpretation context of the value 3850

Complex Semantic Object

- A heterogeneous collection of semantic objects grouped under a corresponding concept:
  
  \[
  \langle \text{FlightOffer}, \{
  \langle \text{ClassOfService}, \text{"Economy"}, \langle \text{ClassOfServiceCode}, \text{"FullServiceClassName"} \rangle \rangle,
  \langle \text{Price}, 1826, \langle \text{Currency}, \text{"DEM"}, \langle \text{Scale}, 1 \rangle \rangle \rangle,
  \langle \text{FlightSegment}, \{
  \langle \text{FlightNumber}, 400 \rangle,
  \langle \text{AirlineIdentifier}, \text{"LH"}, \langle \text{AirlineIdentifierCode}, \text{"TwoLetterAirlineCode"} \rangle \rangle,
  \langle \text{DepartureDate}, \text{"Jun 06 2000"}, \langle \text{DateFormat}, \text{"Mon DD YYYY"} \rangle \rangle,
  \langle \text{DepartureTime}, \text{"10:35"}, \langle \text{TimeFormat}, \text{"HH:MM"} \rangle \rangle,
  \langle \text{DepartureAirport}, \text{"FRA"}, \langle \text{AirportIdentifierCode}, \text{"ThreeLetterAirportCode"} \rangle \rangle,
  \langle \text{Service}, \text{"S"}, \langle \text{ServiceCode}, \text{"OneLetterServiceCode"} \rangle \rangle,
  \langle \text{Service}, \text{"L"}, \langle \text{ServiceCode}, \text{"OneLetterServiceCode"} \rangle \rangle
  \rangle \rangle
  \]

Conversion Function

- A function that converts semantic objects between different contexts
  
  \[
  \phi_{\text{Currency}}(\{\langle \text{Currency}, \text{"USD"} \rangle, \langle \text{Price}, 1826, \langle \text{Currency}, \text{"DEM"} \rangle \} \}) = \langle \text{Price}, 830, \langle \text{Currency}, \text{"USD"} \rangle \rangle
  \]
  - with "1 USD = 2.2 DEM"

- Provide a prerequisite for the integration of data from different sources
Semantic Equivalence

- Semantically equivalent objects represent the same information
  - \(<\text{Distance}, 3.850, \{<\text{Unit}, \text{"mile"}, <\text{Scale}, 1000>\}>\)
  - \(<\text{Distance}, 3850, \{<\text{Unit}, \text{"mile"}, <\text{Scale}, 1>\}>\)

- Determined through conversion to a common context and the comparison of their data values
- Depends in general on the context and conversion function used

Semantic Identity

- Semantically identical objects represent information about the same real world object
- Complex objects: if all identifying attributes are semantically identical (recursive)
- Simple objects: if they are semantically equivalent

Representation & Domain-specific Ontologies

- Representation Ontologies
  - domain-independent physical representation basis
  - enables exchange and reuse of concepts
  - contains concepts like Numeric-Value, or Character-String
- Domain-specific Ontologies
  - refer to a concrete subject domain
  - provide a consistent conceptualization of this domain
  - contains concepts like FlightOffer, or Distance

Data Integration based on MIX

1. Mapping to the MIX model
2. Conversion to a common context
3. Unification of semantically identical objects

Ontology-based Infrastructure

- In loosely-coupled (component-based) systems interfaces should consider the issues related with data interpretation
- Specification of arguments must allow the inclusion of metadata in order to correctly interpret them
  - Calculate(\(<\text{Price}, 21000, \{<\text{Currency}, \text{"EUR"}>)\>\)
- Arguments can be converted to the target context before using them
Adapters / Wrappers

Adapters / Wrappers (cont.)

Sem. Data Exchange - Summary

- Common interpretation basis for data
- Implicit modeling assumptions have to be made explicit
- Business-to-Business-to-Consumer:
  - need to extract, prepare and consolidate data for further electronic processing

Sem. Data Exchange – Summary (2)

- MIX:
  - allows to make explicit data semantics by using ontologies and metadata
  - supports integration, and automatic processing of Internet data
  - representation of data plus additional semantic metadata
  - flexible, self-describing data model for the representation of semi-structured data
  - conversion functions