

Semantic Data Exchange

Mariano Cilia
cilia@informatik.tu-darmstadt.de

Motivation

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

2

Motivation

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

3

Motivation

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

4

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

5

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

6

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

7

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

8

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

10

Naming conflicts

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

11

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

12

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

13

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

14

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, ... +, -

15

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

16

Overcoming: Type conflicts

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

17

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

18

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

19

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

20

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

21

Example

Availability for FRANKFURT (FRA) to KENNEDY-NEW YORK (JFK)
Saturday, June 06 2000

| Select | Airline | Flight | Departing City | Time | Arriving City | Time | Stops | Meal |
|--------|---------|--------|----------------|-------|---------------|-------|-------|------|
| | LH | 400 | FRA | 10:35 | JFK | 13:00 | 0 | S.L. |

Price Per Adult (Economy Class): DEM 1826

Direct flight on Saturday June 6, 2000
Departing: FRA Frankfurt, Frankfurt Germany
Arriving at: JFK John F. Kennedy Int'l Airport, New York New York
Lufthansa, flight number 400, departing 10:35 AM, arriving 1:00 PM
Class: Y - Economy Coach; Flight Dist.: 3850 Miles

You can reserve this/these flight(s) at a fare of \$ 830 for one adult, incl. taxes.

Example

Availability for FRANKFURT (FRA) to KENNEDY-NEW YORK (JFK)
Saturday, June 06 2000

| Select | Airline | Flight | Departing City | Time | Arriving City | Time | Stops | Meal |
|--------|---------|--------|----------------|-------|---------------|-------|-------|------|
| | LH | 400 | FRA | 10:35 | JFK | 13:00 | 0 | S.L. |

Price Per Adult (Economy Class): DEM 1826

Direct flight on Saturday June 6, 2000
Departing: FRA Frankfurt, Frankfurt Germany
Arriving at: JFK John F. Kennedy Int'l Airport, New York New York
Lufthansa, flight number 400, departing 10:35 AM, arriving 1:00 PM
Class: Y - Economy Coach; Flight Dist.: 3850 Miles

You can reserve this/these flight(s) at a fare of \$ 830 for one adult, incl. taxes.

Example

Availability for FRANKFURT (FRA) to KENNEDY-NEW YORK (JFK)
Saturday, June 06 2000

| Select | Airline | Flight | Departing City | Time | Arriving City | Time | Stops | Meal |
|--------|---------|--------|----------------|-------|---------------|-------|-------|------|
| | LH | 400 | FRA | 10:35 | JFK | 13:00 | 0 | S.L. |

Price Per Adult (Economy Class): DEM 1826

Direct flight on Saturday June 6, 2000
Departing: FRA Frankfurt, Frankfurt Germany
Arriving at: JFK John F. Kennedy Int'l Airport, New York New York
Lufthansa, flight number 400, departing 10:35 AM, arriving 1:00 PM
Class: Y - Economy Coach; Flight Dist.: 3850 Miles

You can reserve this/these flight(s) at a fare of \$ 830 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

25

Ontology

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

26

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

27

Simple Semantic Object

- A data item with additional metadata to support its interpretation:

< Distance, 3850, {<Unit, "mile">, <Scale, 1>} >

- **3850** is the recorded data value
- **Distance** denotes the ontology concept
- **{<Unit, "mile">, <Scale, 1>}** represents the interpretation context of the value 3850

28

Complex Semantic Object

- A heterogeneous collection of semantic objects grouped under a corresponding concept

```
< FlightOffer, {
  < ClassOfService, {
    "Economy", {<ClassOfServiceCode,
    "FullServiceClassName">} >,
  < Price, 1826, {<Currency, "DEM">, <Scale, 1>} >,
  < FlightSegment, {
    < FlightNumber, 400 >,
    < AirlineIdentifier, "LH", {<AirlineIdentifierCode,
    "TwoLetterAirlineCode">} >,
    < DepartureDate, "Jun 06 2000", {<DateFormat, "Mon DD
    YYYY">} >,
    < DepartureTime, "10:35", {<TimeFormat, "HH:MM">} >,
    < DepartureAirport, "FRA", {<AirportIdentifierCode,
    "ThreeLetterAirportCode">} >,
    ...
  < Service, "S", {<ServiceCode, "OneLetterServiceCode">} >,
  < Service, "L", {<ServiceCode, "OneLetterServiceCode">} >
  > } >
```

29

Conversion Function

- A function that converts semantic objects between different contexts

$\varphi_{\text{Currency}}(\{<\text{Currency}, \text{"USD"}>\}, \{<\text{Price}, 1826, \{<\text{Currency}, \text{"DEM"}>\}>\})$

$= \{<\text{Price}, 830, \{<\text{Currency}, \text{"USD"}>\}>\}$

with "1 USD = 2.2 DEM"

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

30

Semantic Equivalence

- Semantically equivalent objects represent the same information

< Distance, 3.850, {<Unit, "mile">, <Scale, 1000>} >
< Distance, 3850, {<Unit, "mile">, <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

31

Semantic Equivalence

- Semantically equivalent objects represent the same information
 - < Distance, **3850**, {<Unit, "mile">, <Scale, 1>} >
 - < Distance, **3850**, {<Unit, "mile">, <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

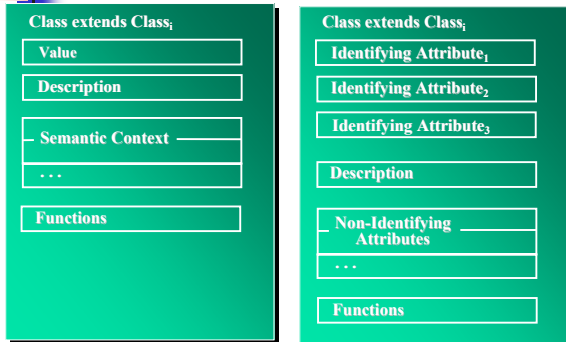
32

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

33

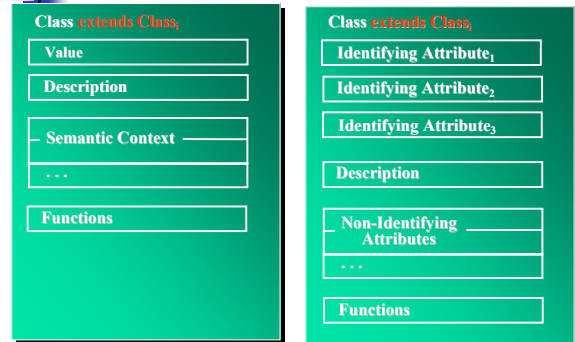
Representing Ontology Concepts



Simple Semantic Objects

Complex Semantic Objects 34

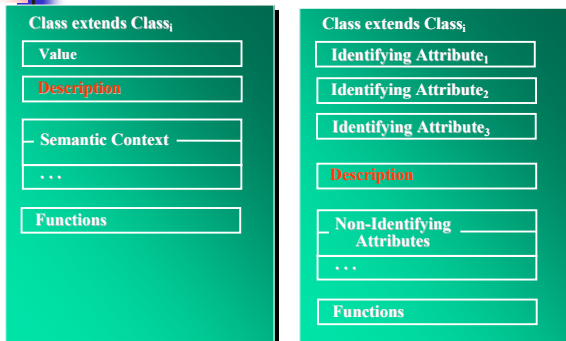
Representing Ontology Concepts



Simple Semantic Objects

Complex Semantic Objects 35

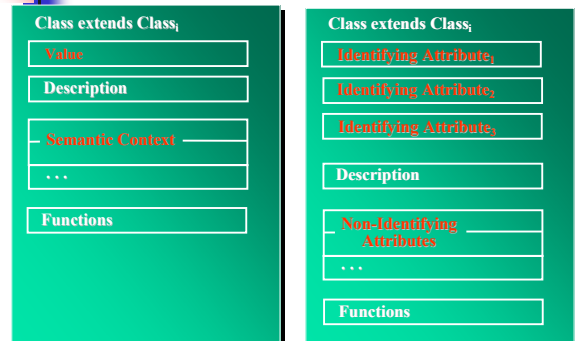
Representing Ontology Concepts



Simple Semantic Objects

Complex Semantic Objects 36

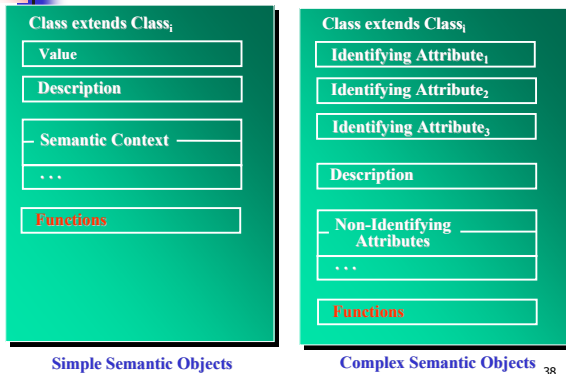
Representing Ontology Concepts



Simple Semantic Objects

Complex Semantic Objects 37

Representing Ontology Concepts



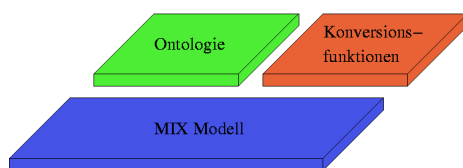
38

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

39

Abstract Model



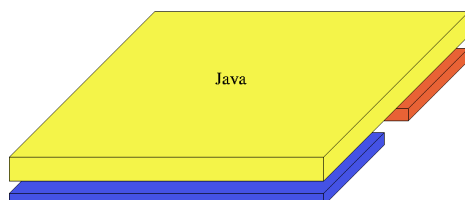
40

Representing Ontology Concepts

- Advantages of using Java for the representation of ontology concepts:
 - it avoids any impedance mismatch between programming and ontology specification language
 - making concepts available as pre-compiled classes allows their shipping, and that of the corresponding data objects, between different platforms

41

Java Implementation (MIBIA)



42

Representing Ontology Concepts

- Disadvantages of using Java for the representation of ontology concepts:
 - Only Java programs can manipulate concepts
 - Representation dependency
 - It is not easy to
 - Identify instances
 - Share context among instances
 - Make relationships among concepts explicit

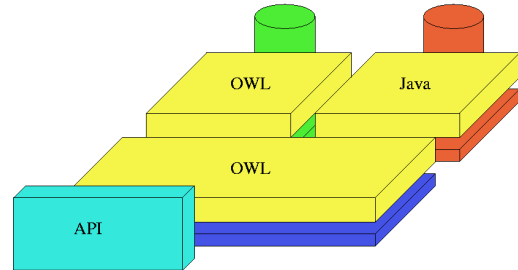
43

Representation

- Description of semantic objects using OWL (Web Ontology Language)
 - Candidate Recommendation of the W3C
 - Based on XML and therefore programming language and platform independent
 - Use of URI as global identifiers (needed for context-sharing)

44

Current Implementation



45

OWL Representation - Sample

```

<owl:Thing rdf:ID="Price1">
  <mix:type>Simple</mix:type>
  <mix:concept>EngMath#Price</mix:concept>
  <rep:value>
    <xsd:string rdf:value="99.95"/>
  </rep:value>
  <rep:semanticContext
    rdf:resource="#CurrencyContext"/>
</owl:Thing>
    
```

46

OWL Representation – ... Sample

```

<rep:SemanticContext rdf:ID="CurrencyContext">
  <rep:aspect rdf:resource="#EuroCurrency"/>
</rep:SemanticContext>

<owl:Thing rdf:ID="EuroCurrency">
  <mix:type>Simple</mix:type>
  <mix:concept>EngMath#Currency</mix:concept>
  <rep:value>
    <xsd:string rdf:value="Euro"/>
  </rep:value>
</owl:Thing>
    
```

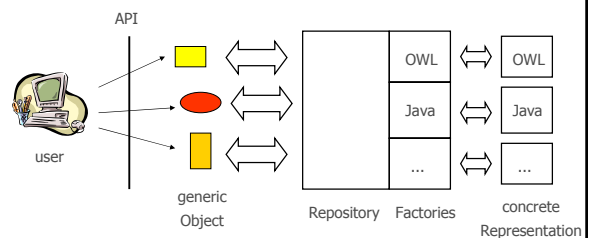
47

Current API

- Generic objects abstract from its representation
- Access and manipulation through a clean interface
- Store and load by using the repository
- Serialization and deserialization using factories

48

Current Implementation



49

