Laboratory of Applied Computer Science (LISI)
National Engineering School for Mechanics and Aerotechnics (ENSMA)
1, avenue Clément Ader - BP 40109 - 86961 Futuroscope cedex - France

Ontologies and Functional Dependencies for Data Integration and Reconciliation

Abdelghani BAKHTOUCHI
ESI, Algiers, Algeria
a_bakhtouchi@esi.dz

Ladjel BELLATRECHE
LISI/ENSMA, Poitiers, France
bellatreche@ensma.fr

Yamine AIT AMEUR
ENSEEIHT, IRIT, Toulouse, France
yamine@enseeiht.fr

More-BI'2011, October 31, 2011, Brussels

Domain ontology: definitions

What is an ontology?

« *An explicit specification of a conceptualization* » [Gruber 93]

Conceptualization of domain classes and properties [JoDS'08]

- **Formal**
- **Consensual**
- **Capability to be referenced**

« *a formal and consensual dictionary of categories and properties of entities of a domain and the relationships that hold among them* »

A Taxonomy of Domain Ontologies

- DB: Canonical vocabulary of the concepts of a domain

  identifier + primitive concepts {classes / properties / datatypes}

  → Canonical Conceptual Ontology (CCO)

- AI: Deductive capabilities

  Concept equivalence operators

  → Non Canonical Conceptual Ontology (NCCO)

- Computational Linguistics: Terms of a domain

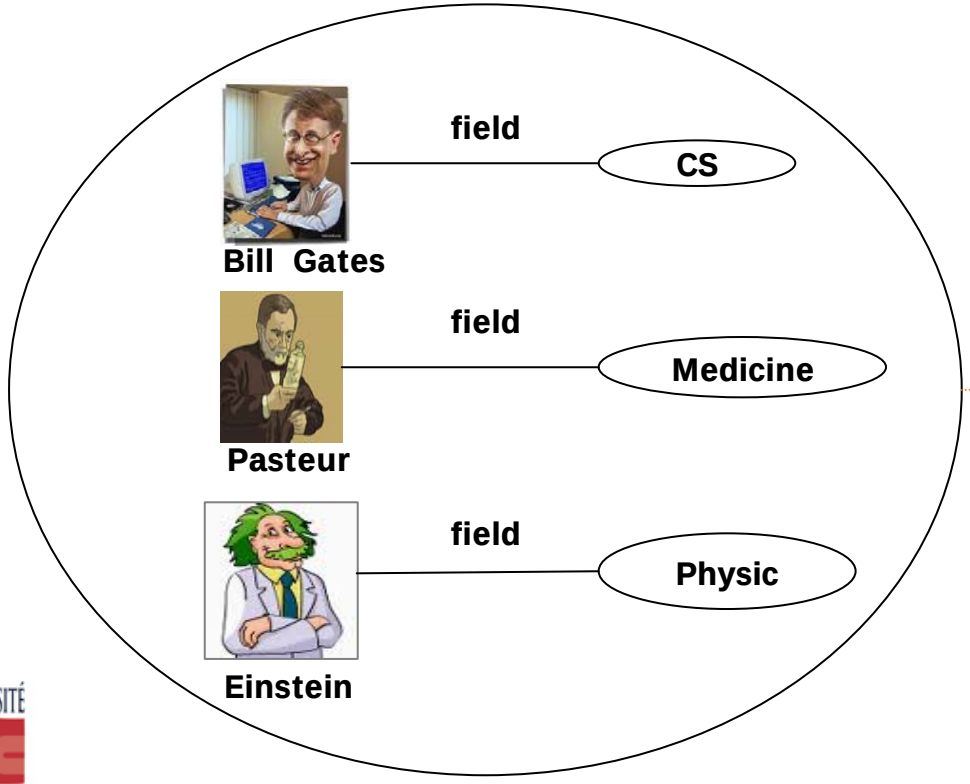
  { words } + similarities and linguistic relationships

  →Linguistic Ontology (LO)

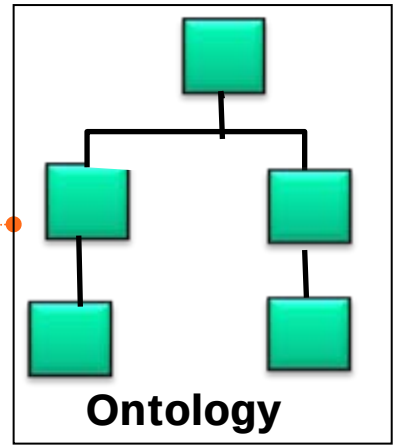
Three points of view = three complementary categories of Ontologies

Explicitation of Data Semantic

- Existence of ontological instances
  → RDF, RDF Schema, PLIB, OWL



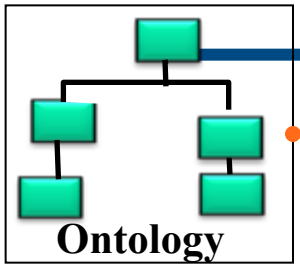
References

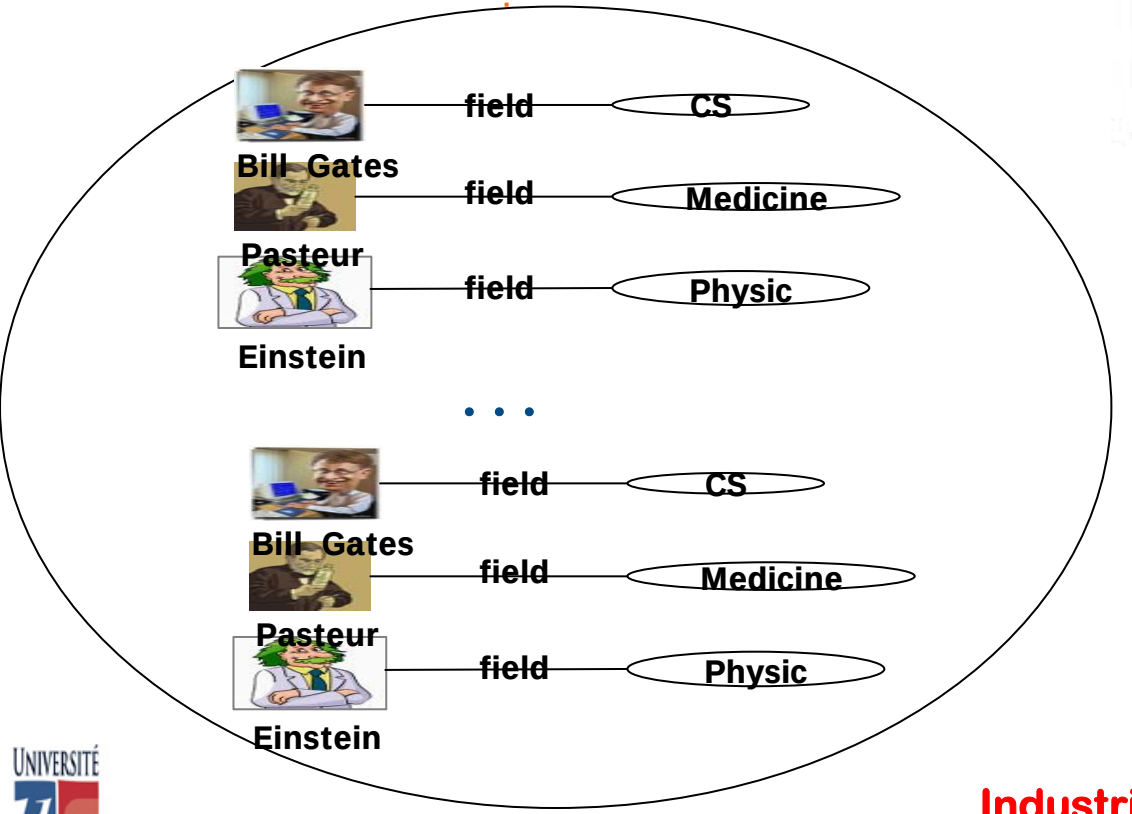


Model Level

Instance Level

Explosion of Ontological Instances



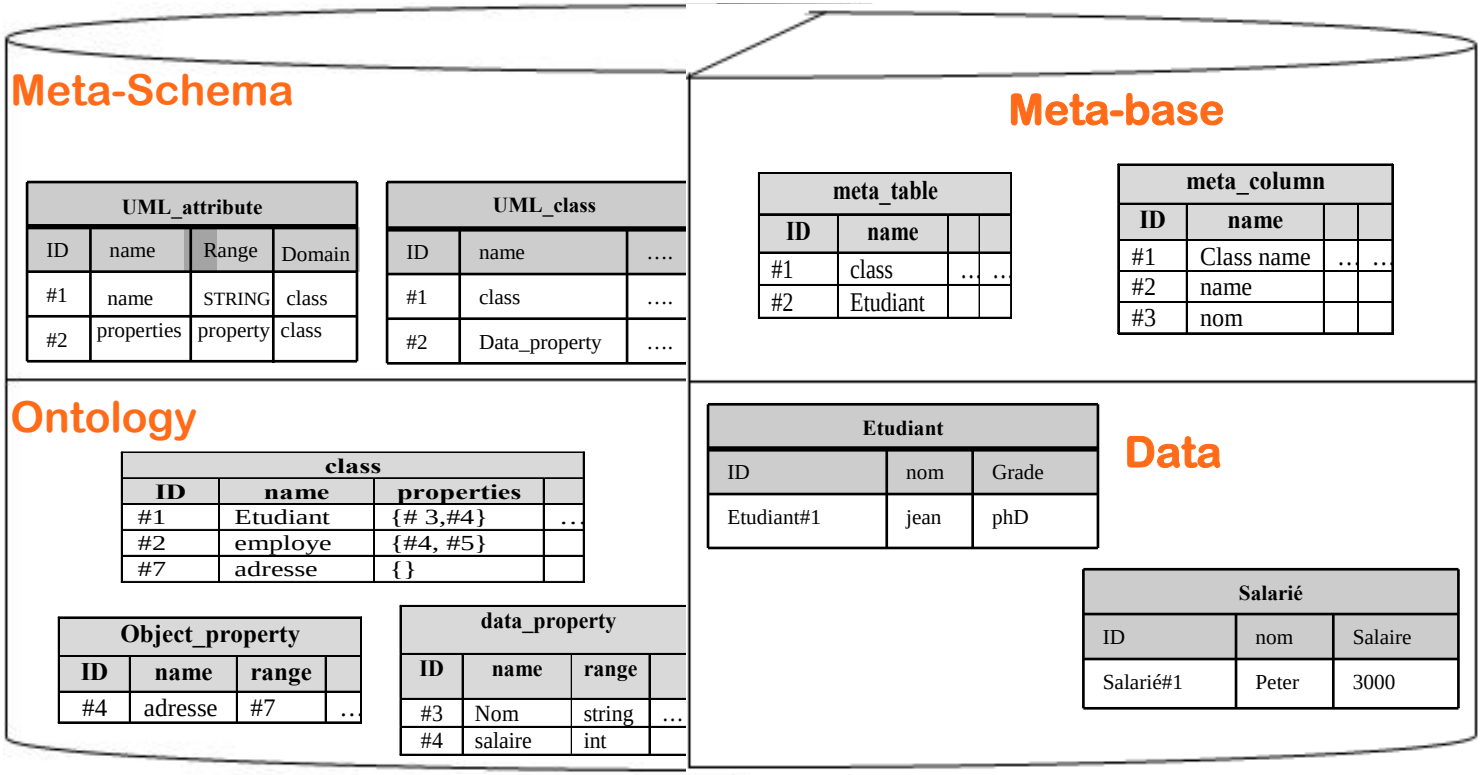


Persistance

Industrial: Oracle, IBM SOR
Academic: Sesame, Owlgres, OntoDB

Ontology-based Database Sources: OntoDB

- Schema: meta-model of ontology model
- Instances :
  - Model of ontology +
  - meta-model of the used modeling language



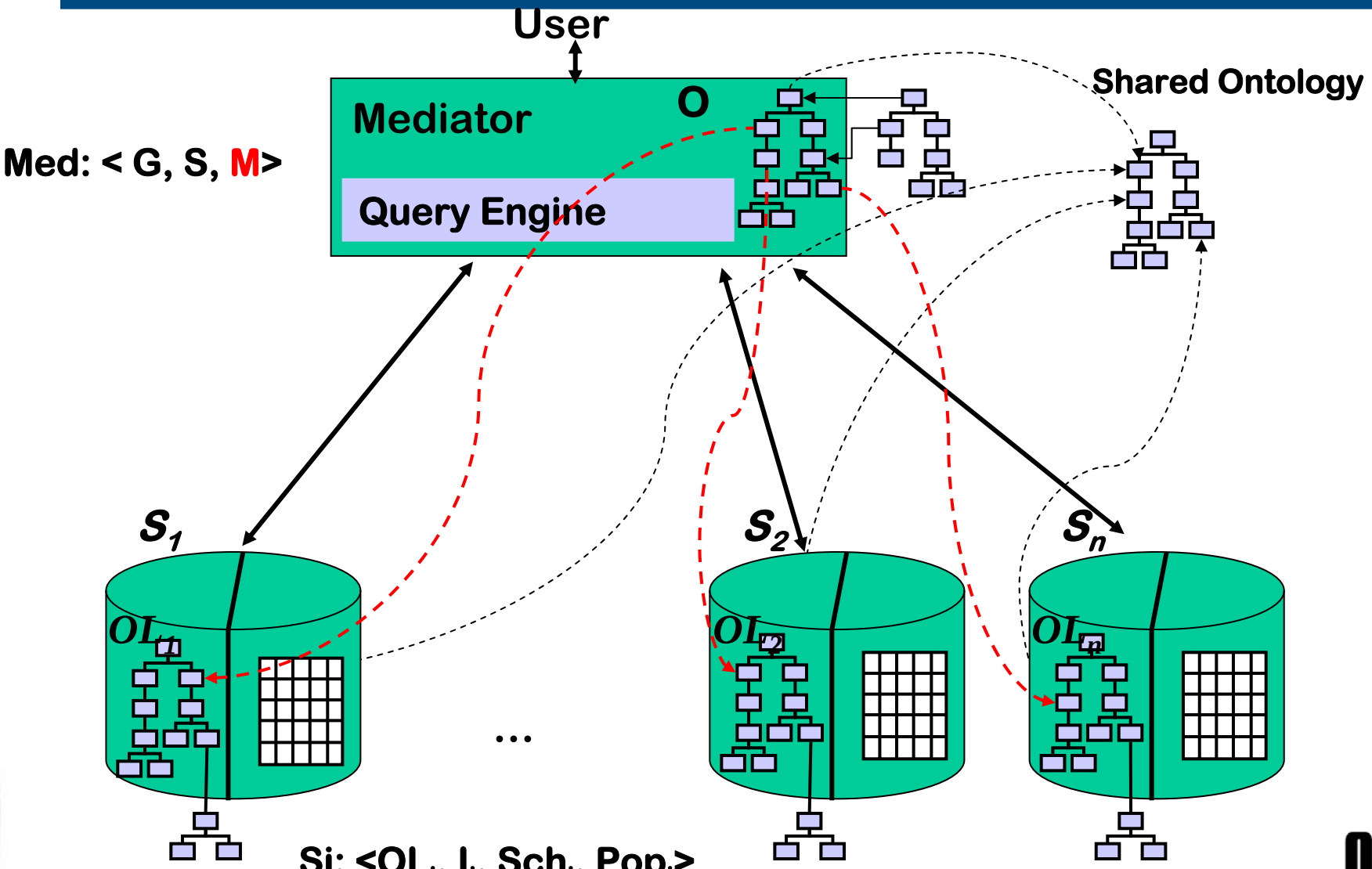
- Schema : onlology model
- Instances : ontologies

- Schema : subset of the ontology
- Instances : instances

Agenda

1. Ontology-based Integration System
2. Data Reconciliation
3. Components of our Integrated System
4. Validation
5. Conclusion & Perspectives

OBDB Candidate for Integrated Systems



Data Reconciliation

- **Existing Reconciliation Approaches:**
  1. Source Entities Representing the same Concept have the **Same Key**
     Observer, Picsel2, COIN, etc.

  → **A Strong Hypothesis that Violates Source Autonomy**

  2. Different Keys
     - Use of Statistical Methods (Affinity Measures between Concepts)

  → **Not Really Suitable for Sensitive Applications** (Bank, Engineering, …)

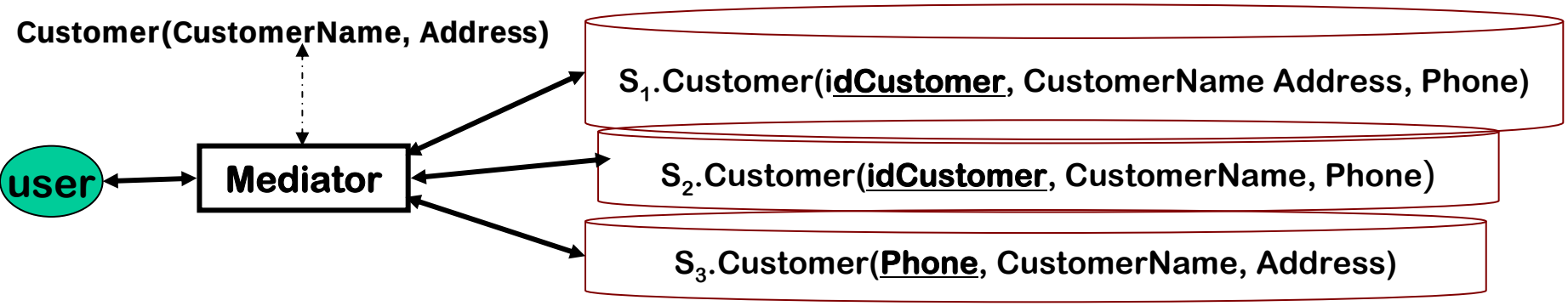
→ Data Integration and Reconciliation are Treated in **Isolated Way**

- **Our Proposal: Joint Treatment of Data Integration & Reconciliation**

→ Use of Functional Dependency (FD) Defined on Ontology Classes

→ Key Candidates become Ontological Concepts

Motivating Example



1. Only results from S_1 and S_2 may be reconciled
   → lost of a part of results.
2. Reconciliation of Results from S_1, S_2 and S_3 without Duplicate Elimination
   → Duplicates may cause errors (Fusion)
3. Duplicates elimination using Statistical Methods
   → Inexact Results

- Functional Dependencies on Customer [IGPL'11, Romero et al. 09]
  - Fd_1: IdCustomer → CustomerName
  - Fd_2: IdCustomer → Address
  - FD_3: IdCustomer → Phone
  - Fd_4: Phone → CustomerName
  - Fd_5: Phone → Address.
- Key Candidates:
  - IdCustomer (S_1, S_2)
  - Phone (S_1, S_2, S_3): Reconciliation Key

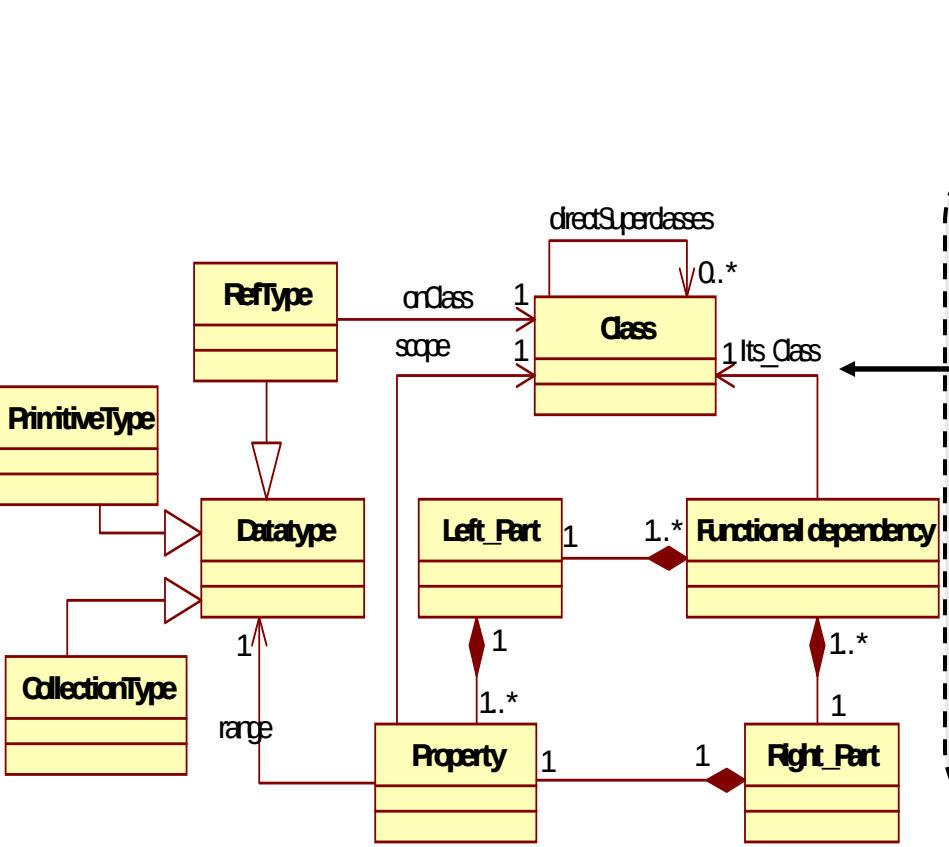
Formal Definitions of an Ontology

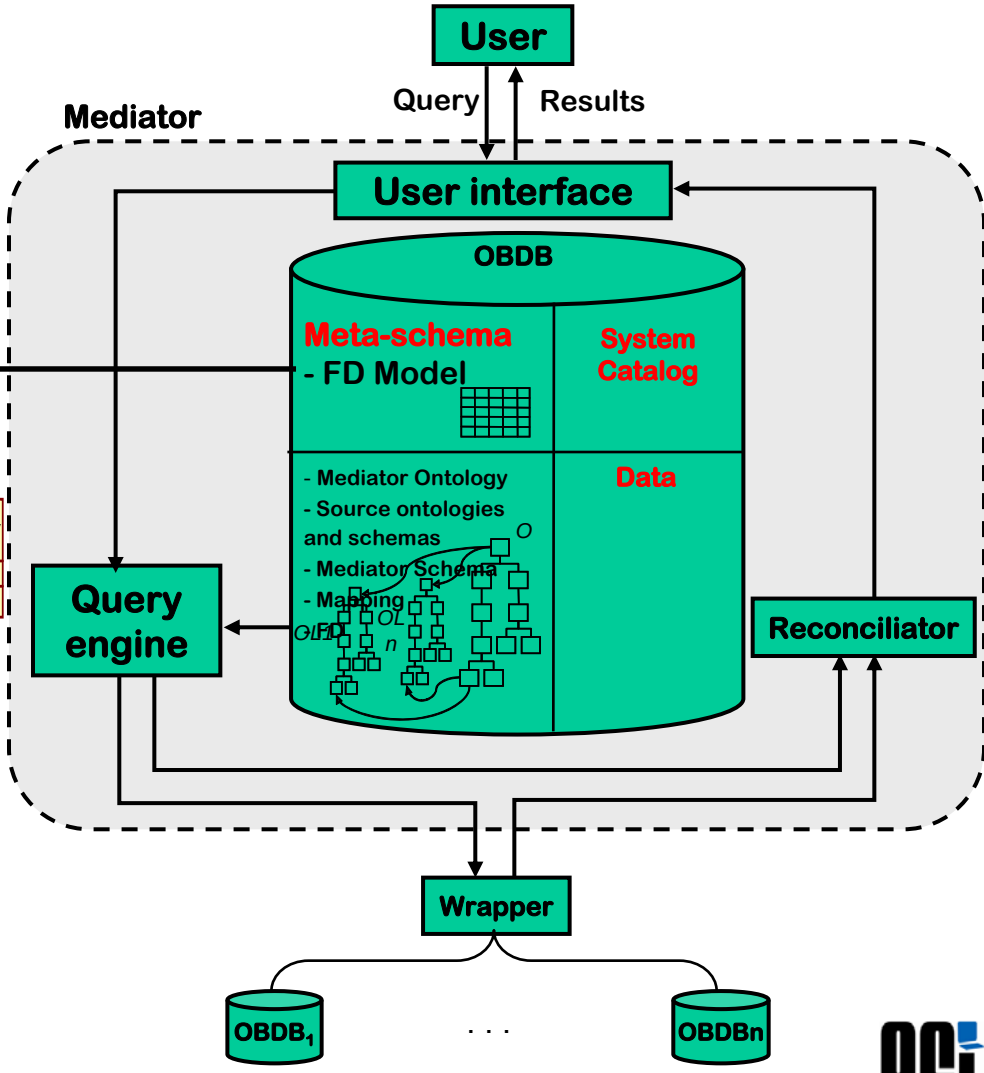
- **Traditional Definition [Jean'07, JoDS'08]**
  - O: < C, P, Sub, Applic>
    - **C**: ontology classes;
    - **P**: set of properties used to describe of ontology classes C;
    - **Sub**: $C \rightarrow 2^C$ subsumption relationship;
    - **Applic**: $C \rightarrow 2^P$ applicable properties for each class.

- **FD = (R, LP, RP): fd R : LP → RP [Calbimonte'09]**
  - **R** : root class
  - **LP**: Left Part: list of properties $\{p_1 \ldots p_n\}$
  - **RP**: Right Part : a single property

- **Formal Definition Enriched by FD**
  - O: < C, P, Sub, Applic, FD > with FD :< R, LP, RP>
  - $R \in C, \quad LP \in 2^P, \quad RP \in P$

Global Architecture



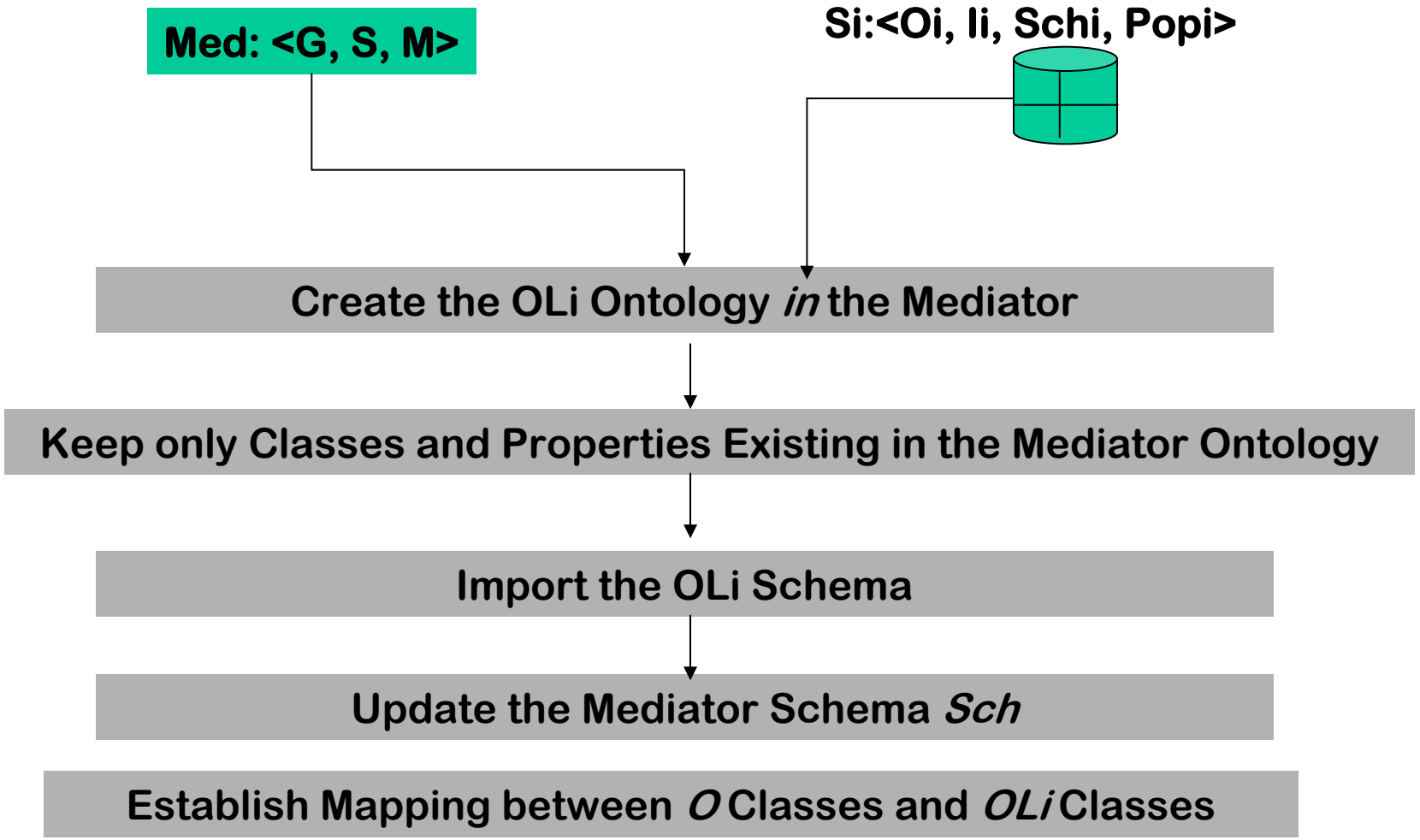


Components of Mediator

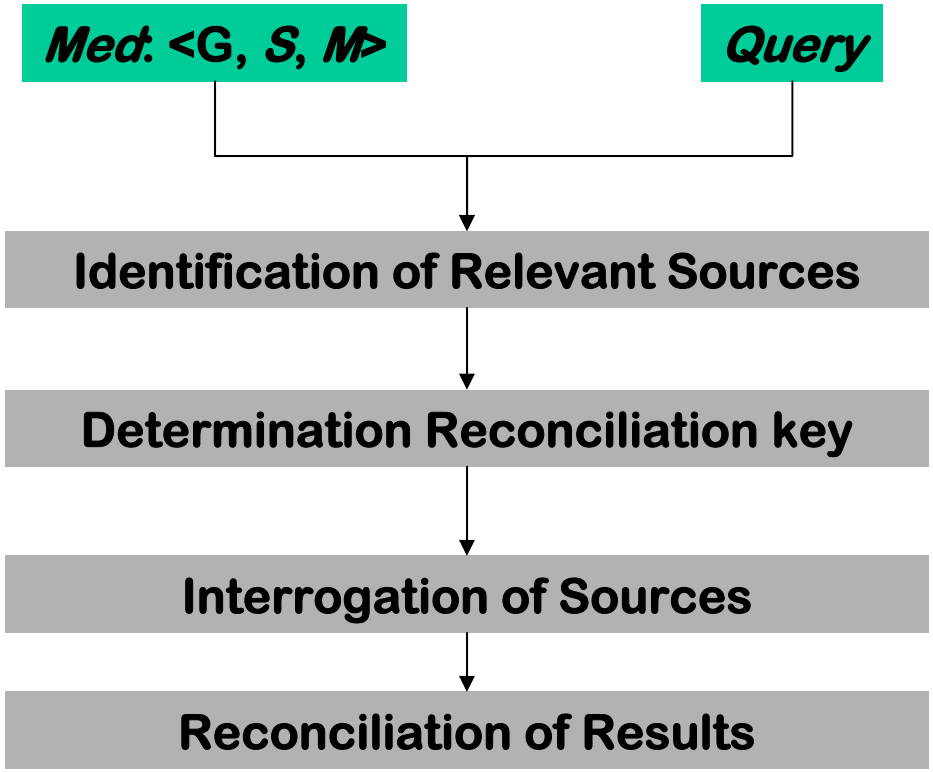
Med: <G, S, M>

1. **G: <O, Sch>**
   - O <C, P, Applic, Sub, FD> : mediator ontology
   - Sch: C $\rightarrow 2^P$ associates to each class c $\in$ C the properties describing the instances of the class c, which are valuated in at least one integrated source.
2. **S = {S_1, S_2, …, S_n}**
   - S_i:<OL_i, $SchL_i$>
3. **M: C $\rightarrow 2^{\{C_1 U \ldots U C_n\}}$**
   - Mapping between the classes of mediator ontology O and the classes of source ontologies

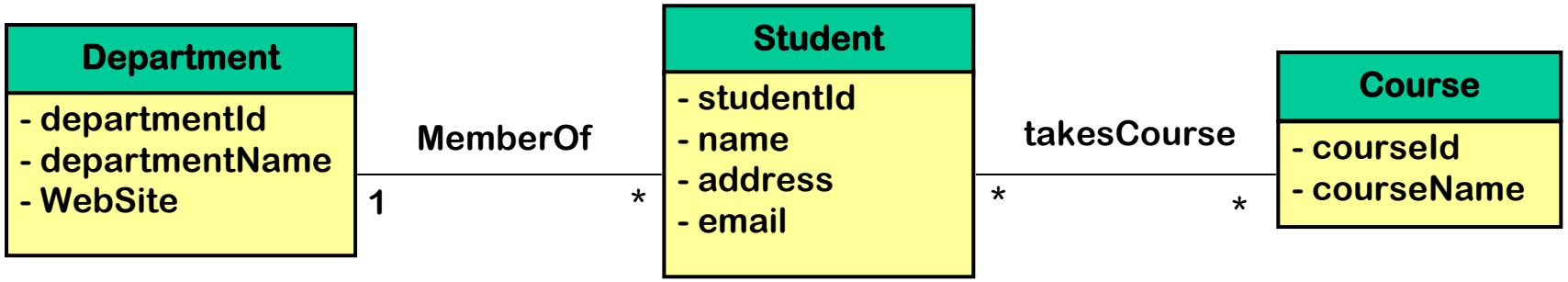
New source Integration



Query Processing



Reconciliation Key Determination



- FD
  - fd1..4 Student : studentId → name, address, email, MemberOf
  - fd5..7 Student : email → name, address, MemberOf
  - fd8,9 Department : departmentId → departmentName, WebSite
  - fd10 Course : courseId → courseName

- Query:
  - Q(x; y; z):- Student (x1), takesCourse(x1, x2), Course(x2), memberOf(x1, x3), Department(x3), name(x1, x), courseName(x2, y), departmentName(x3, z)

Reconciliation Key Determination

- **Projected Properties:**
  Proj = {Name, CourseName, DepartmentName}

- **Direct FD**
  fd_1 .. fd_{10}

- **Generated FD** (by the means of functional property: MemberOf → departmentId; MemberOf → Email)
  fd_{11} : studentId → departmentId, departmentName, WebSite
  fd_{12} : email → departmentId, departmentName, WebSite

- **Reconciliation Key (RK) Generation**
  - RK_0 contains all LP of DFs = {SudentId, Email, DepartmentId, CourseId}
  - StudentId → DepartmentId ⇒ RK_1 = {StudentId, Email, CourseId}

  - StudentId → Name, DepartmentName
  - Email → Name, DepartmentName
    1. If StudentId is valued in all sources ⇒ RK = {StudentId, CourseId}
    2. If Email is valued in all sources ⇒ RK = {Email, CourseId}

Validation (I)

- **Dataset**
  - → Lehigh University Benchmark
  - → Ontology: 45 classes, 32 properties (25 object properties)
  - → OBDB sources are generated
  - → 14 Queries
- **Experiment Environment**
  - → Intel Pentium IV, 3,2 GHz, 1 GB of Memory
- **Execution Time of a Query includes:**

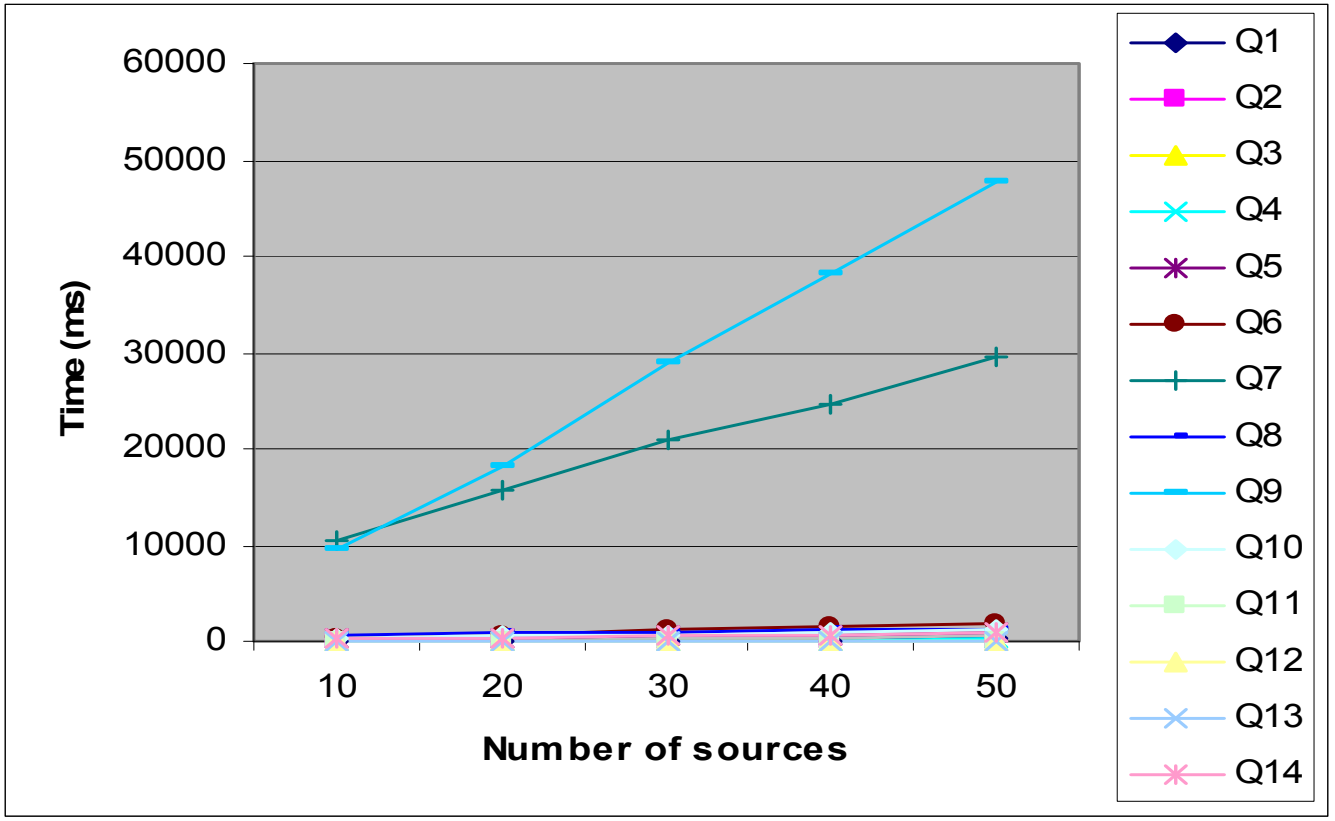
1. Identifying Relevant Source(s)
2. Finding FD that hold for a query
3. Finding Reconciliation Key

} Mediator Tasks

4. Executing Query on each Relevant Source } Source Task

Validation (II)

Query Response Time vs. Number of Sources

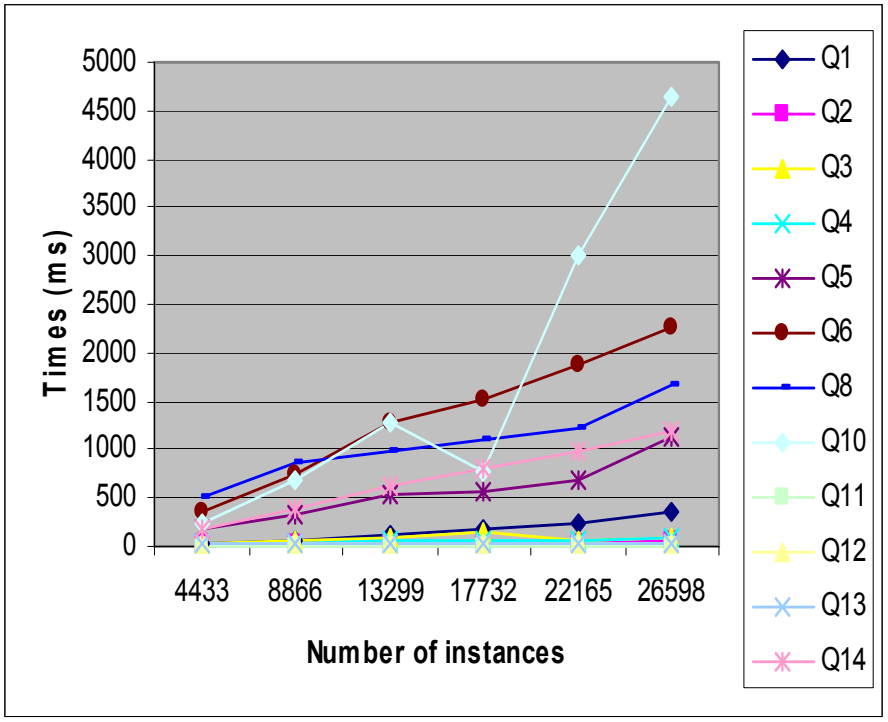


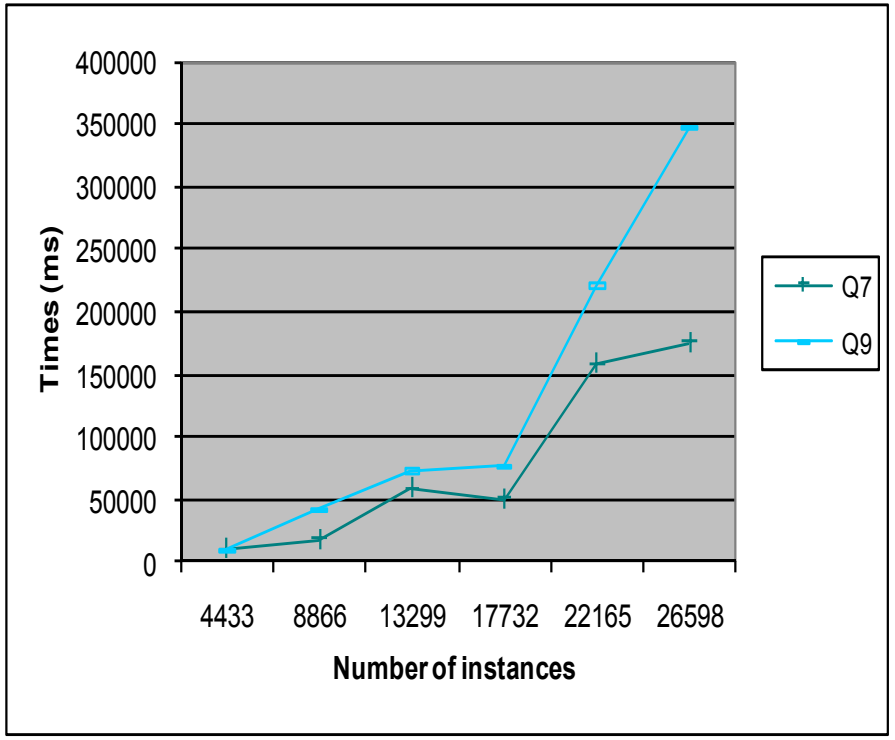
Lessons:

1. Mediator task processing time is negligible
2. Query processing time is important

Validation (III)

Query Response Time vs. Number of Instances





Conclusion & Future Works

- Enrichment of Ontology by FD
- Joint Solution for Data Integration and Reconciliation
- FD based Approach for Instance Reconciliation
- Validation using LUBM

- Intensive Experiments for Evaluating the Robustness of our System
- Development of Quality Metrics
- Ontology-based Data Warehouses