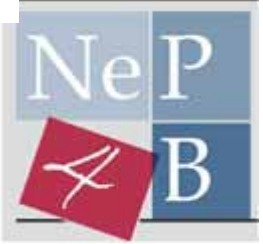
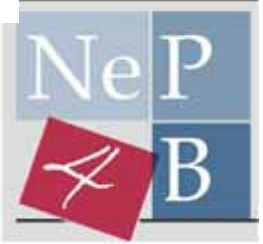


Data and Service Integration

Sonia Bergamaschi – Università di Modena e Reggio Emilia
Francesco Guerra – Università di Modena e Reggio Emilia
Andrea Maurino – Università di Milano – Bicocca
Carlo Batini – Università di Milano - Bicocca

Outline

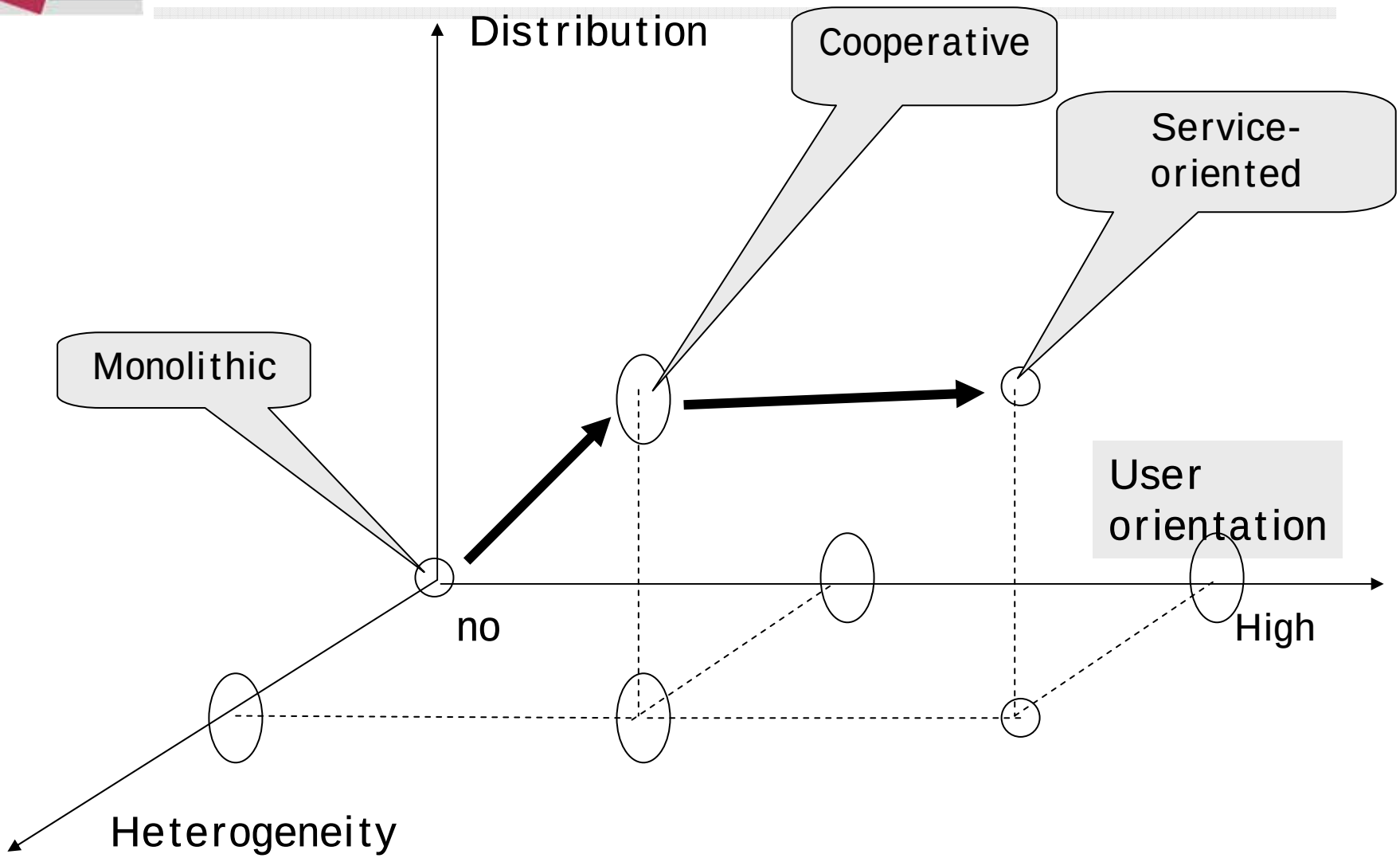
- Why to integrate data and services ? (5 minutes)
- Most relevant issues in data integration (70 minutes- Sonia)
  - State of the art on data integration
- Most relevant issues in services integration (20 minutes-Andrea)
  - State of the art on service integration
- Approaches for data and services integration (90 minutes)
  - The nep4b project (5 minutes - Sonia)
  - The MOMIS system (15 minutes - Sonia)
  - Data & services integration (60 minutes - Andrea)
- Future Directions (5 minutes – Andrea)



WHY DATA AND SERVICE INTEGRATION?

b1

Evolution of information systems



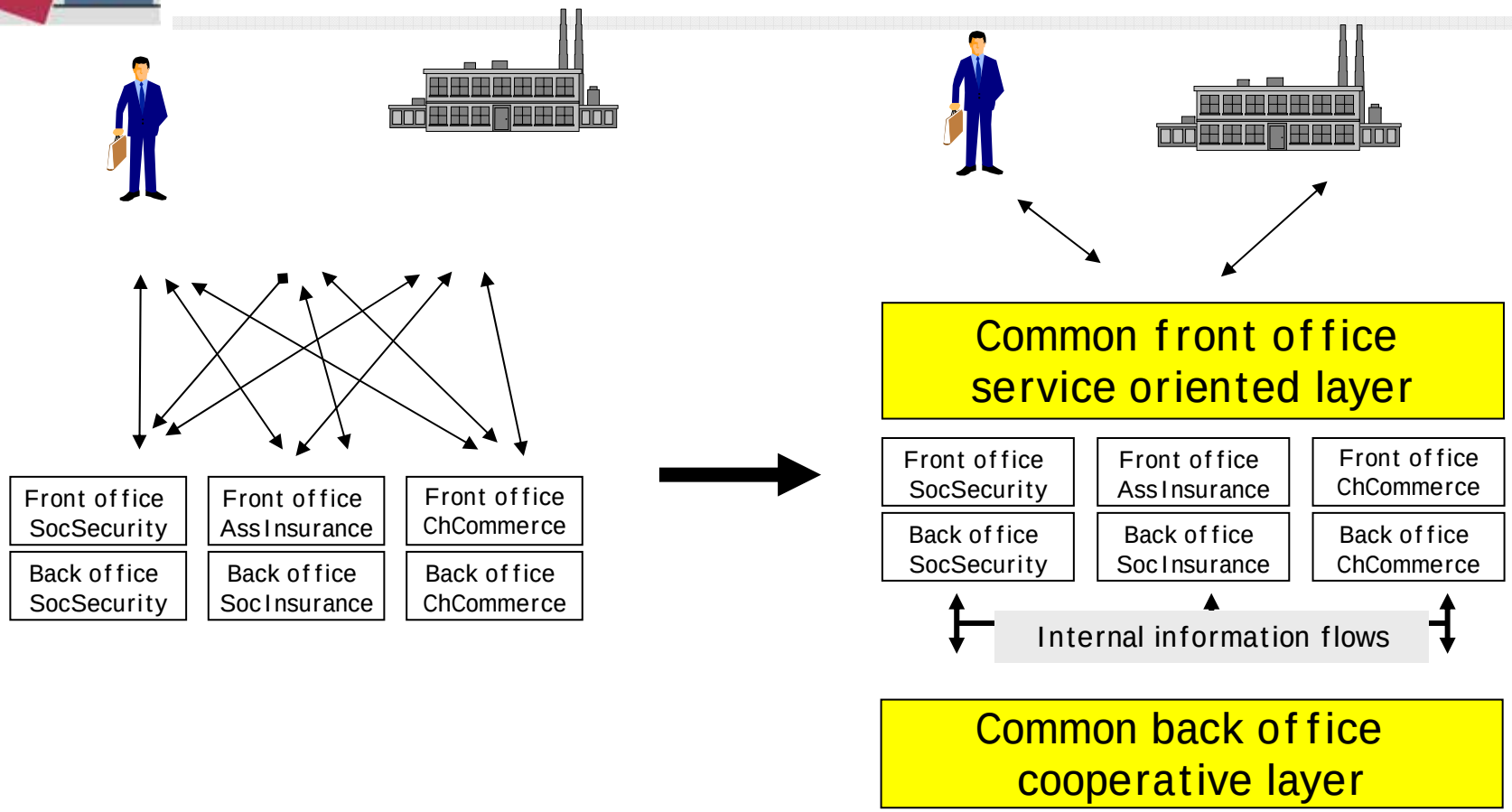
b1

The second area we have to consider in order to express (?) priorities concern the types of information systems. Here we hinerit and adapt a classification proviede by Valduriez for data base management systems, based on the degree of distrubution, heterogeneity and autonomy.

Among all possible combinations of levels, apart from CISs, we are interested to some extent in traditional Monolitic systems, considered in several methodologies. While only in open problems we will discuss DQ in Peer to Peer systems, whose importance in research is growing.

batini; 03/11/2004

Related evolution of enabling ICT architectures



a. Technological architecture in the traditional interaction

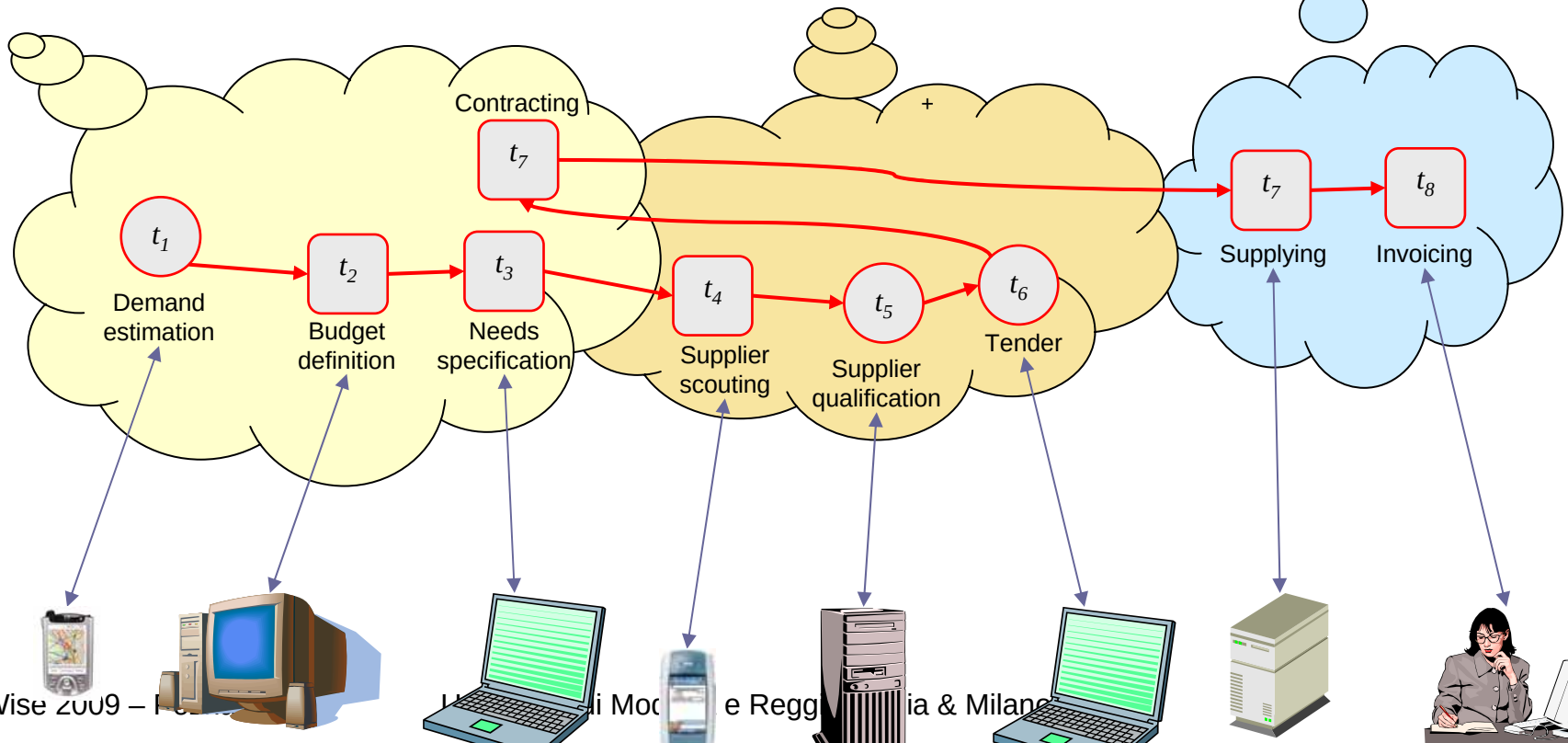
b. Technological architecture in the new interaction

An eProcurement Example: the service side

Administration

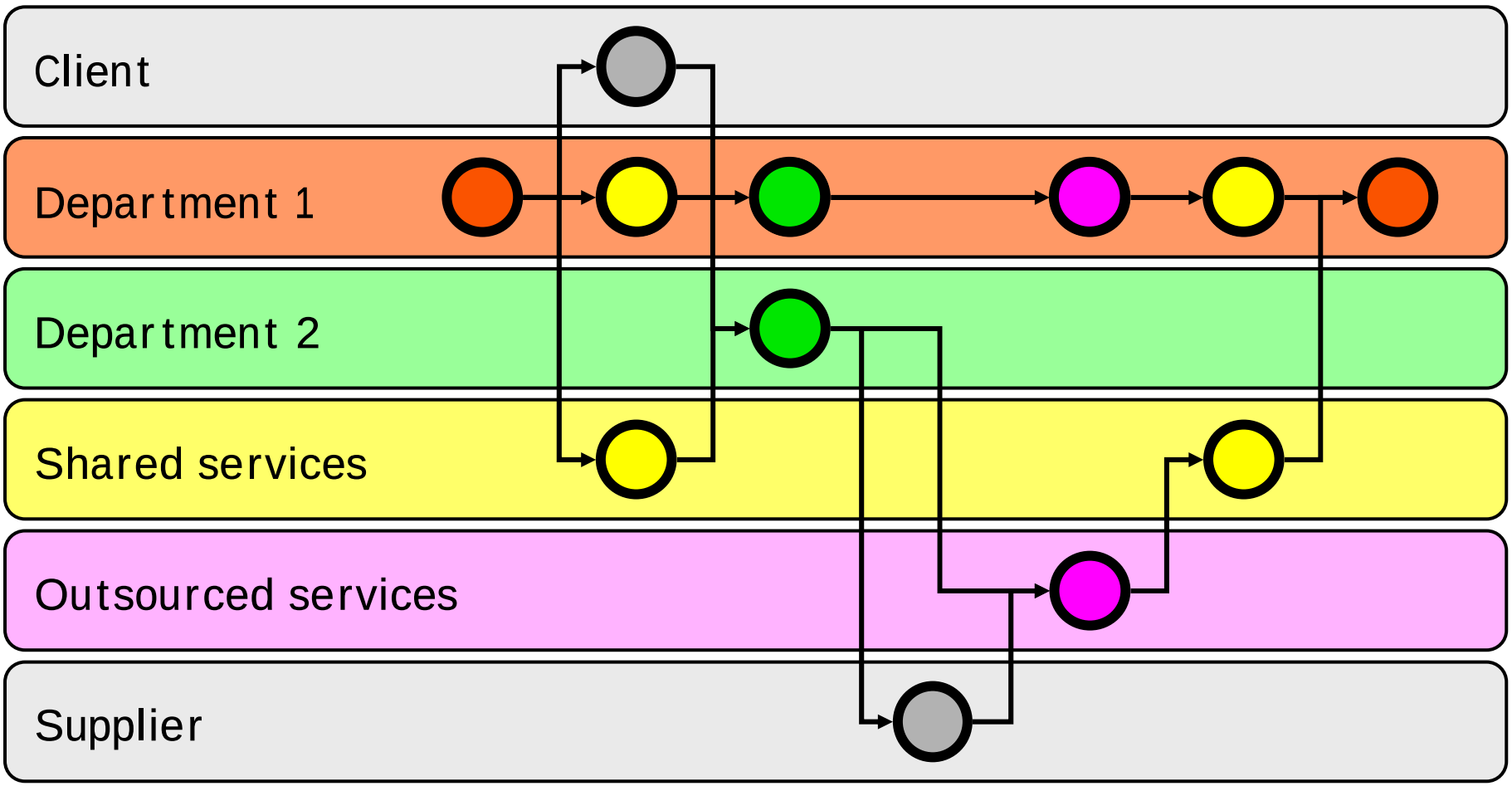
eProcurement Outsourcer

Supplier



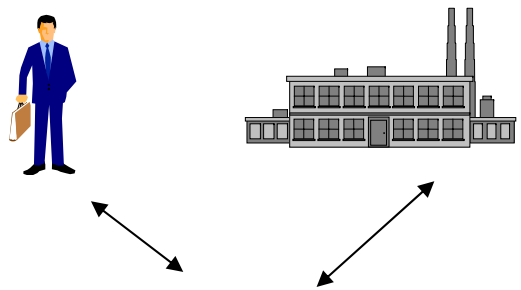
Services are shared among different stakeholders

Services have to be composed to support, besides internal business processes, the network processes involving suppliers and customers



To achieve effective access and composition of services we need some sort of……

Enables the discovery of services useful for business processes and the comparison and choice of cost/ quality optimal services



Common front office service oriented layer

Front office SocSecurity | Front office AssInsurance | Front office ChCommerce

Back office SocSecurity | Back office SocInsurance | Back office ChCommerce

Internal information flows

Integrated Representation of services

Common back office cooperative layer

Enables interoperability, composition and reuse of services

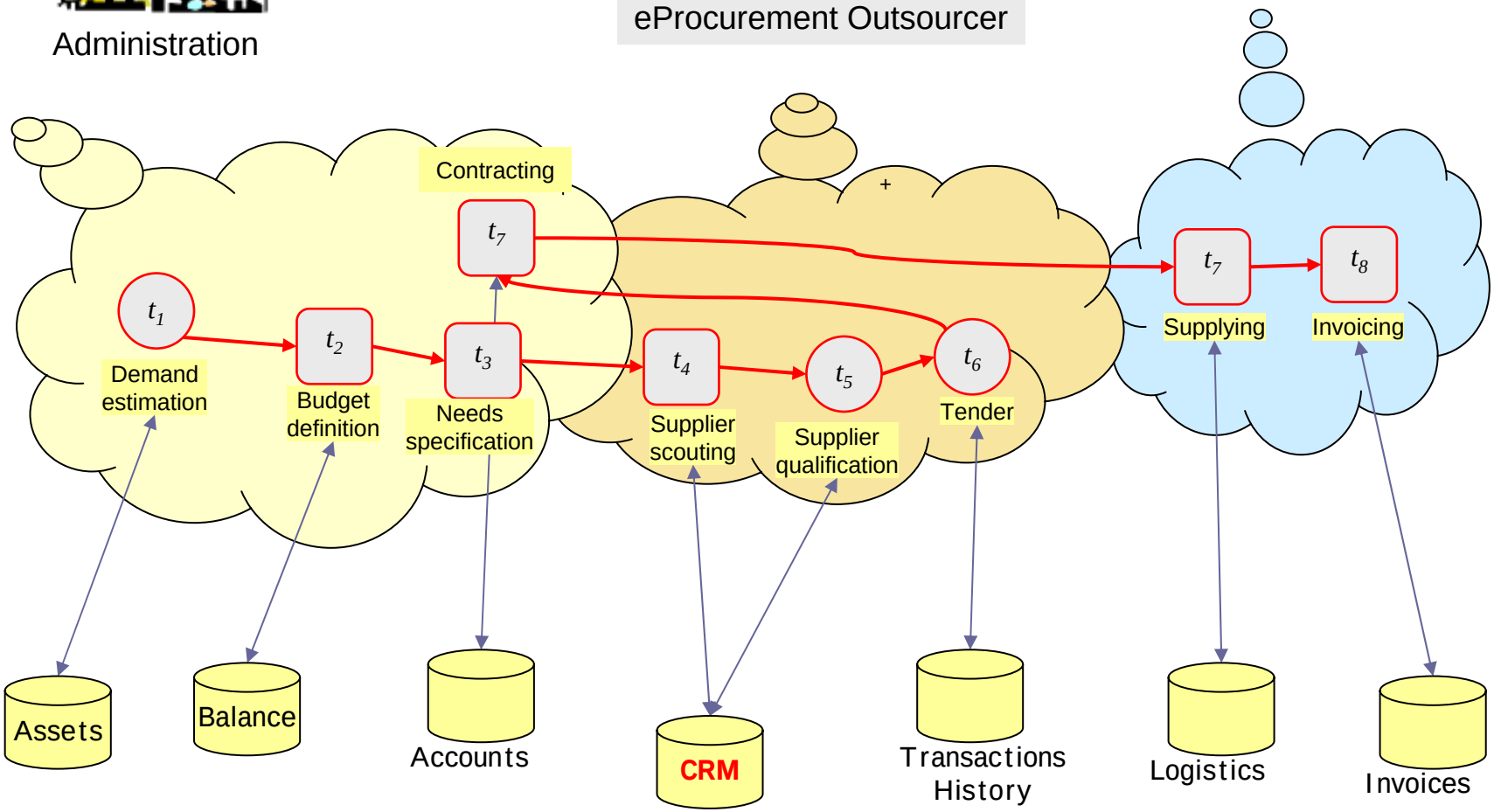
b. Technological architecture in the new interaction

An eProcurement Example: the data side

Administration

eProcurement Outsourcer

Supplier

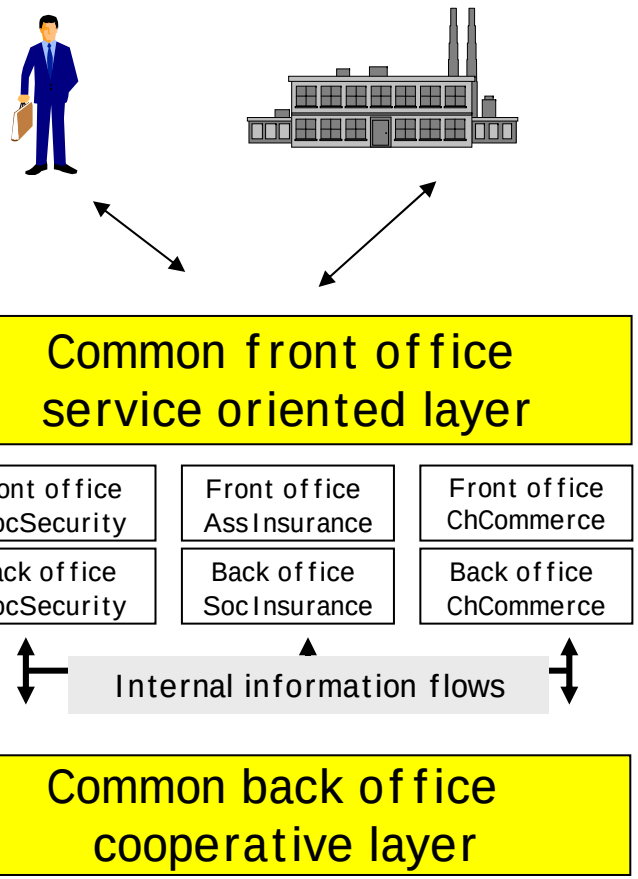


To achieve effective access to and integration of heterogeneous data we need some sort of……

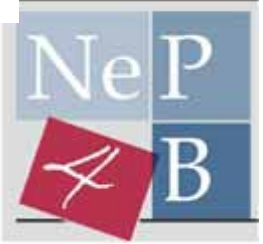
Provides and integrated representation of fragmented data and enables access, integration, reconciliation and fusion of data

Integrated Representation of data

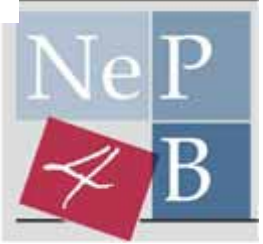
Enables interoperability, integration, redundancy and quality check, and sharing of data



b. Technological architecture in the new interaction



MOST RELEVANT ISSUES IN DATA INTEGRATION



Outline

- What is data integration?
  - Definition and architectures
  - Analysis of the main research issues in data integration
- Data Integration Systems
  - Academic (in Appendix)
  - Commercial
- Future trends

Data integration

- The research community has been investigating data integration for about 20 years: different research communities (database, artificial intelligence, semantic web) have been developing and addressing issues related to data integration:
  - Definitions, architectures, classification of the problems to be addressed
  - Data Integration problems have been analyzed in different perspectives and different approaches have been proposed
  - Developed benchmarks allow the evaluation and the comparison of the approaches (THALIA benchmark)
  - Several commercial software suites have been released and are on testing in real environments

Data Integration

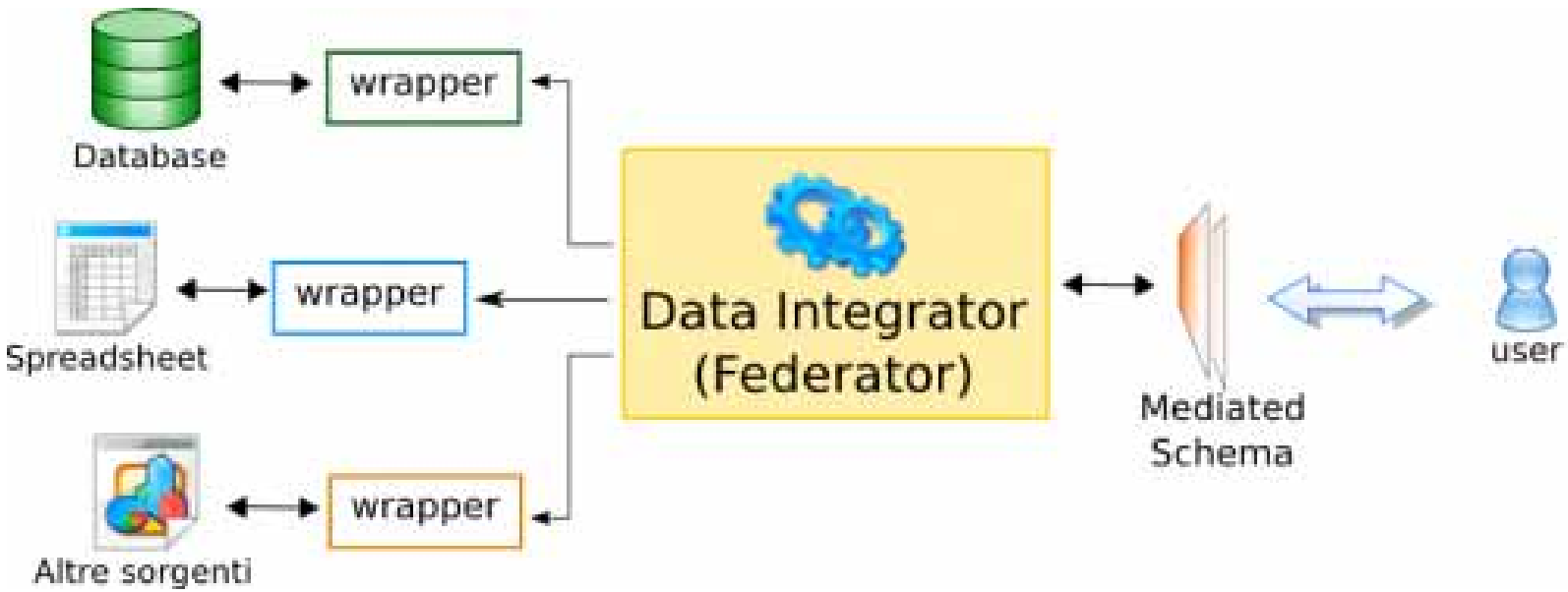
- Modern enterprises are often organized as “virtual networks”, where enterprises operate through inter-enterprise cooperative processes
- To manage inter-enterprise processes and data exchange a key issue is to mediate among the heterogeneity of different information systems:

  Data Integration is a technological solution to build a shared and integrated knowledge base

- Data Integration has to deal with the problems arising from the heterogeneity of data sources:
  - **Structural Heterogeneity:**
    - Different data models
    - Same model but different conceptualization chosen
  - **Semantic Heterogeneity**:
    - Different meaning and interpretation
    - Two schemata might use the same term to denote distinct concepts (homonyms), or different terms to denote the same concept (synonyms)

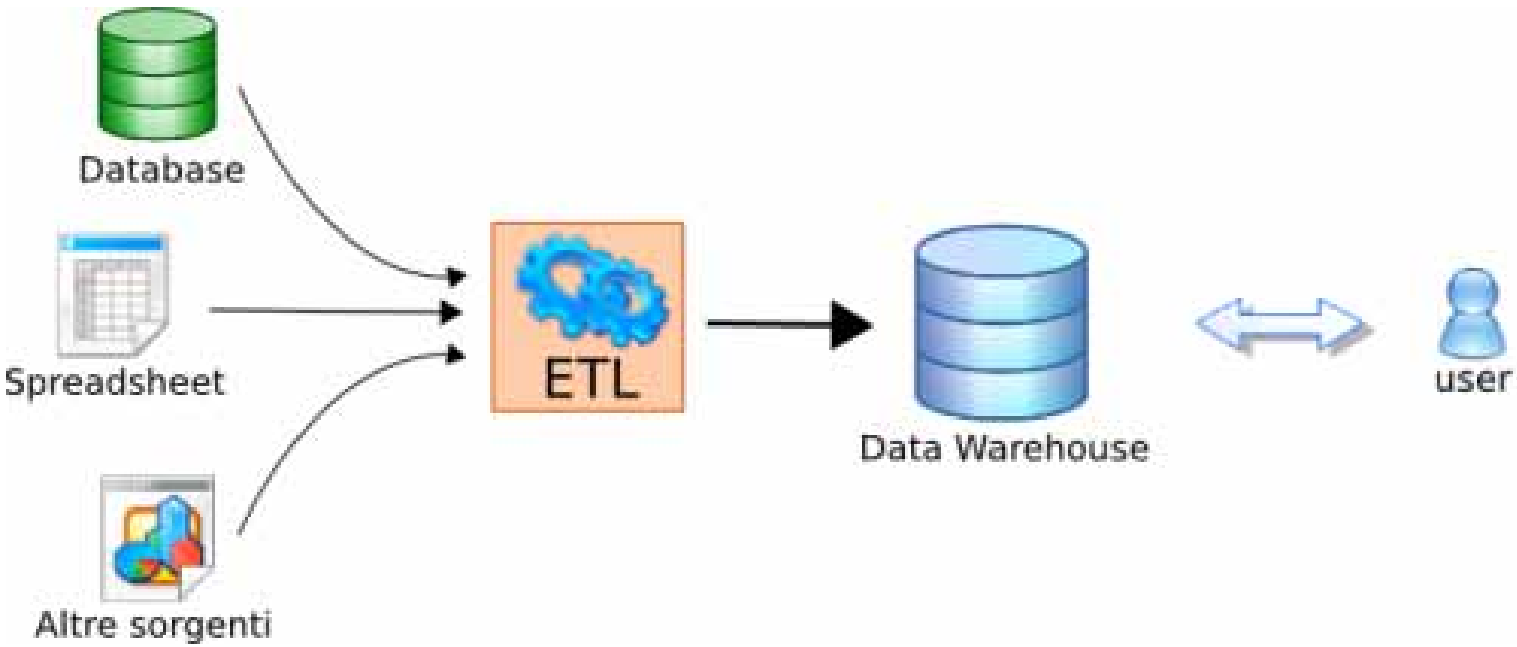
- **Ontologies** are considered fundamental instruments for semantic **interoperability**, allowing a unified data access based on data semantics

Virtual Data Integration



- A system designer builds a *Mediated Schema* over which a user poses queries. The source databases are interfaced with *wrappers* if needed. *Data reside at the data sources*
- Wrapper are format translators

Data Warehouse: Materialized Data Integration

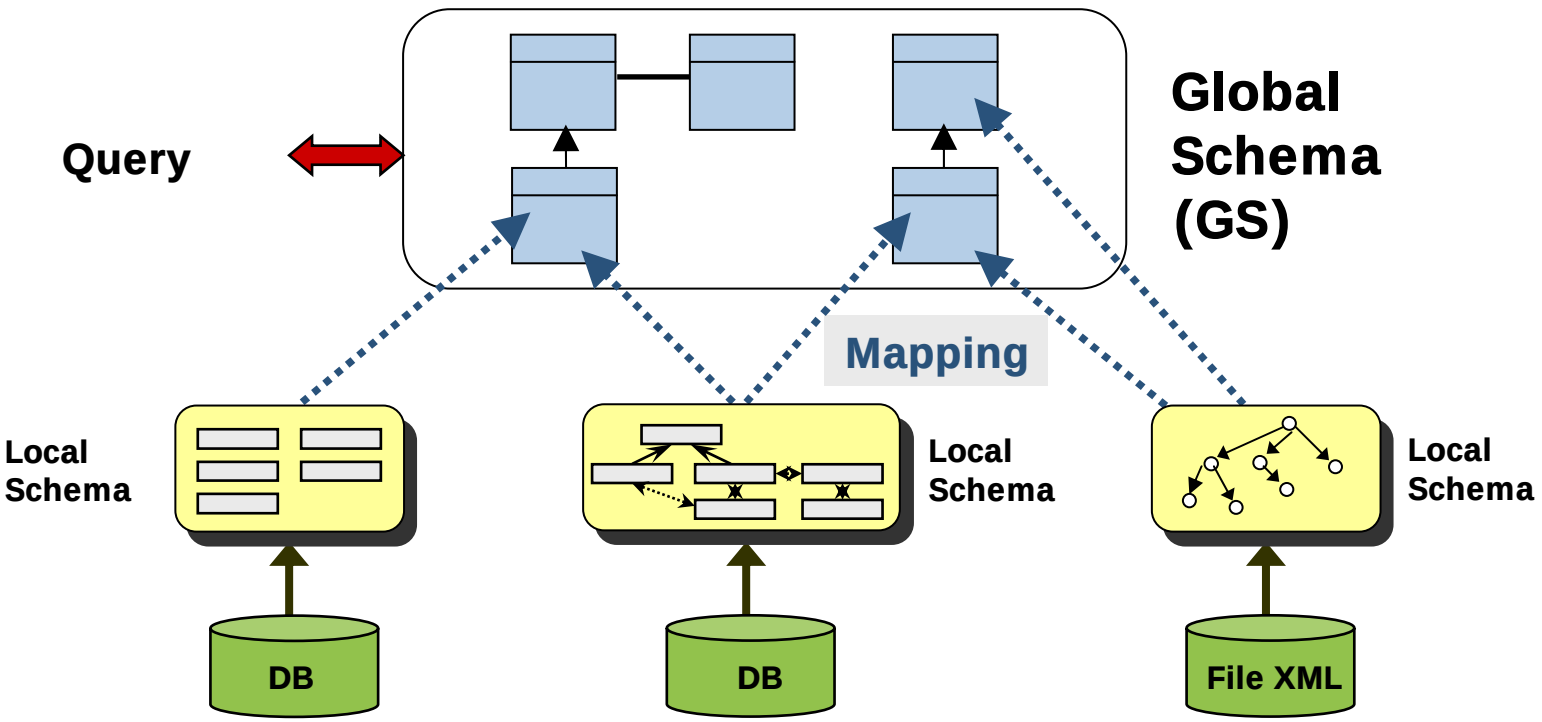


- **Data warehouse** [Inmon (1997)]: "Subject-oriented, integrated, time-variant (temporal), non volatile collection of summary and detailed data, used to support strategic decision-making process for the enterprise"
- The information from the source databases is extracted, transformed then loaded into the data warehouse. ETL = Extraction Transformation Loading

Integration of Heterogeneous & Distributed Data Sources

- Data integration provides a Global Virtual Schema (GS) that

"Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data" (Global Virtual Schema (GS)) [Lenzerini, 2002]



Data integration – several approaches

- Data integration stands for several approaches for combining data from different data sources [Hull, 1997]:
  - **Integrated read-only views: Mediation.** To support an integrated, read-only, view of data that resides in multiple databases (the majority of academic and commercial systems)
  - **Multiple databases sharing information: Federation.** This architecture provides a framework whereby each database extends its schema to incorporate subsets of the data held in the other member databases
  - **Integrated read-write views: Mediation with update.** An extension of the mediation architecture to support updates against an integrated view

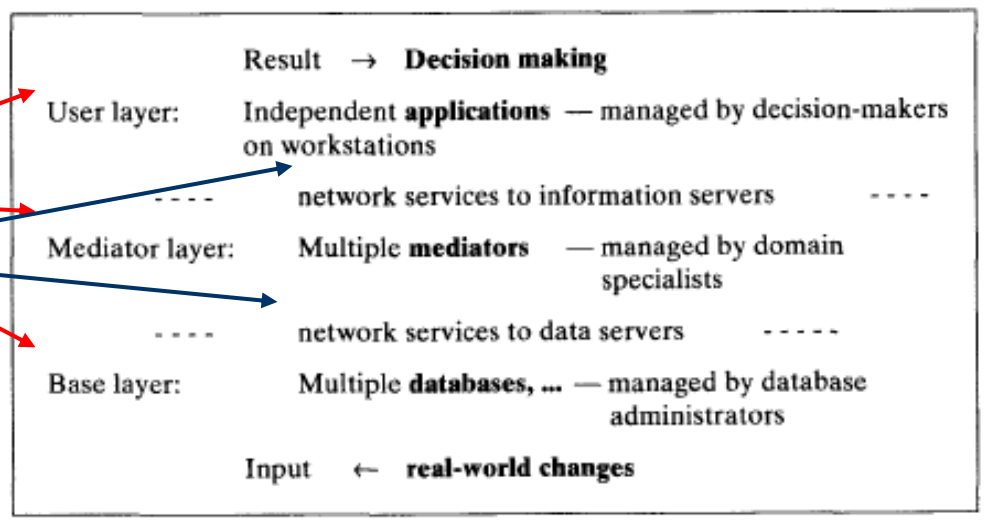
Data integration – Future directions

- **Peer Data Management Systems** [Halevy et al. 2003]. A centralized, easily extensible data management architecture in which any user can contribute new data, schema information, or even mappings between other peers' schemas. PDMSs represent a natural step beyond data integration systems, replacing their single logical schema with an interlinked collection of semantic mappings between peers' individual schemas"
- **On-the-fly data integration systems.** They rely on techniques for creating a unified representation of several data sources as automatic as possible (see Data Integration - New Trends)

Mediators

- The mediators definitely represent one of the most studied architecture for data integration
- "A mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications" [Wiederhold, 1992]

- A mediator is functionally represented by:
  - three layers
  - two interfaces



Mediators (2)

- The mediator builds a unified schema of several (heterogeneous) information sources and allows a user to formulate a query on it
- The user query is transformed in a set of sub-queries, one for each data source involved in the query
- The results are collected by the Mediator, merged and shown to the user
- We may divide the interactions with a mediator in two phases:
  1. The creation of the unified representation (Publishing phase)
  2. The formulation and the execution of a query in the unified representation (Querying phase)

Mediators – relevant challenges

Publishing Phase

Visualizing the unified schema

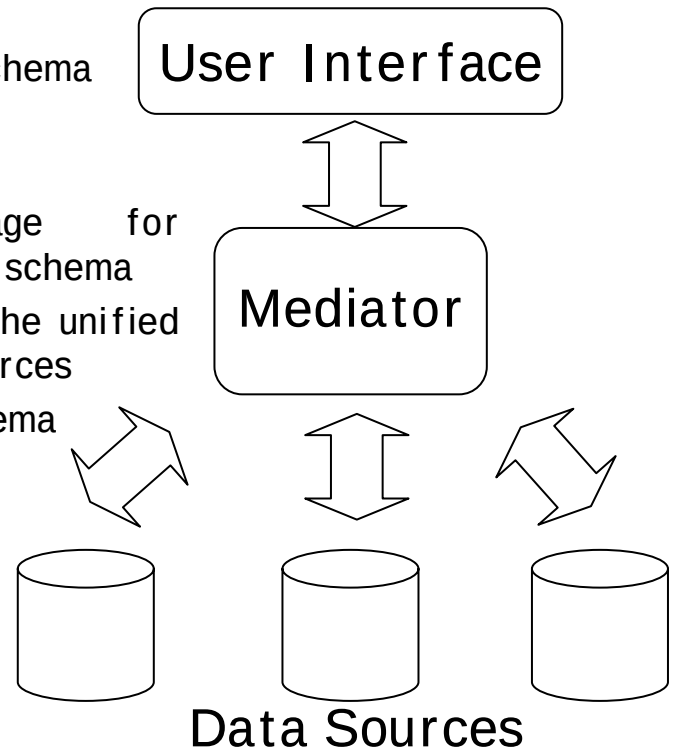
Model and language for representing the unified schema

Matching and Mapping the unified schema and the local sources

Building the unified schema

Managing updates

Schema extraction



Querying Phase

Model and Language for formulating queries

Model and language for querying the schema

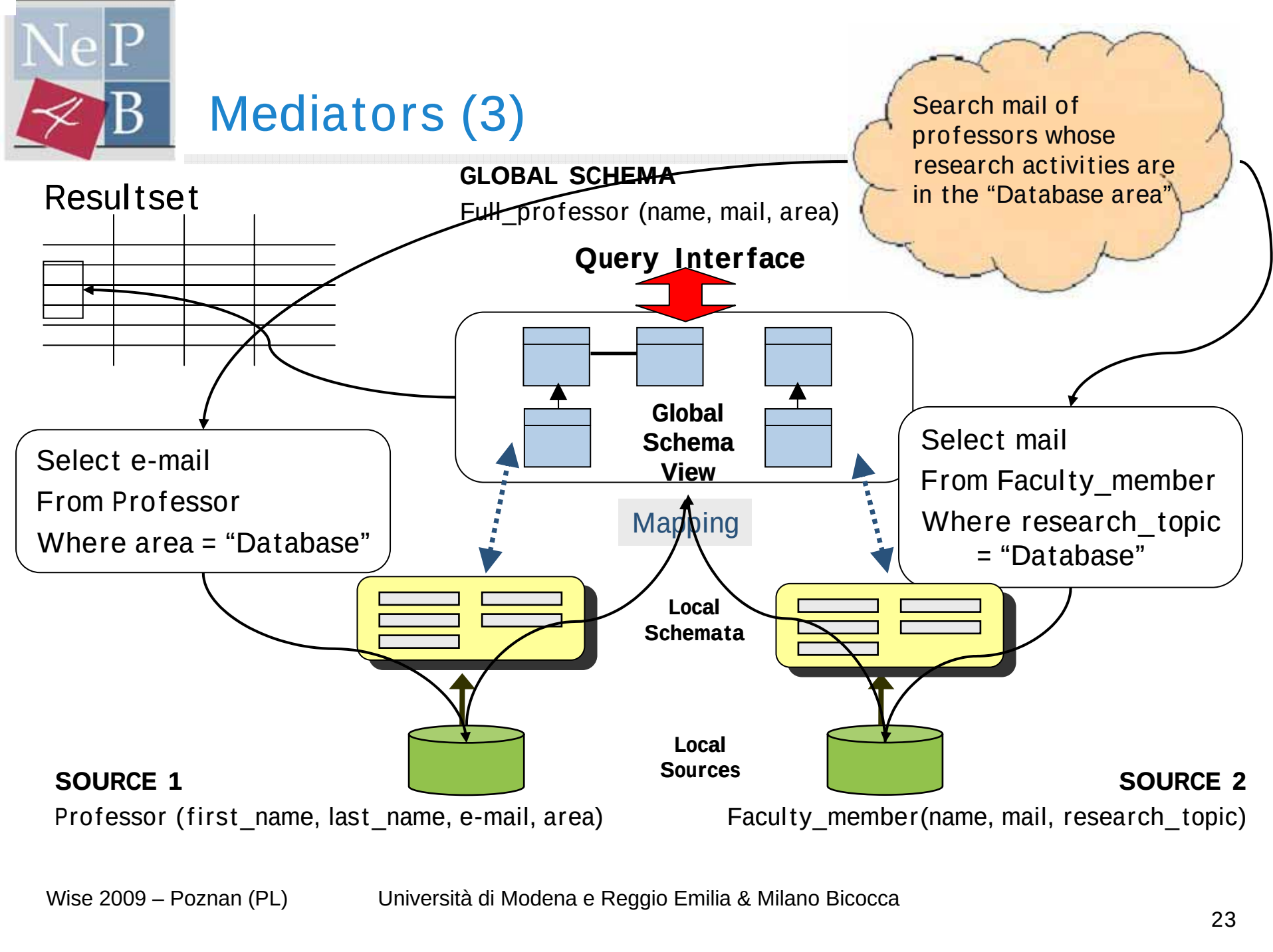
Query unfolding / rewriting

Data fusion and cleaning

Query transformation and execution

Out of the scope of this presentation

Mediators (3)


Search mail of professors whose research activities are in the “Database area”
GLOBAL SCHEMA
Full_professor (name, mail, area)
Resultset
Query Interface
Global Schema View
Mapping
Select e-mail
From Professor
Where area = “Database”
Select mail
From Faculty_member
Where research_topic = “Database”
Local Schemata
Local Sources
SOURCE 1
Professor (first_name, last_name, e-mail, area)
SOURCE 2
Faculty_member(name, mail, research_topic)

Mediators – relevant challenges

Publishing Phase

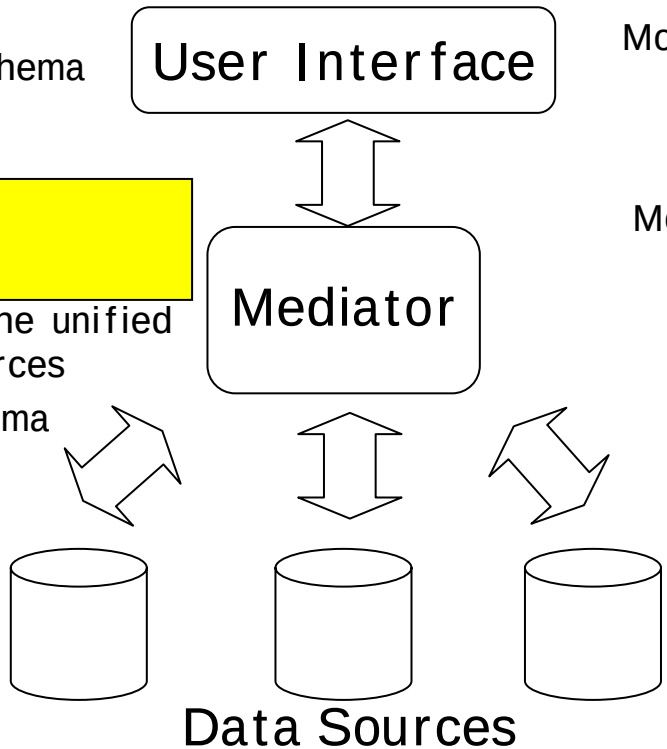
Visualizing the unified schema

Matching and Mapping the unified schema and the local sources

Building the unified schema

Managing updates

Schema extraction



Querying Phase

Model and Language for formulating queries

Model and language for querying the schema

Query unfolding / rewriting

Data fusion and cleaning

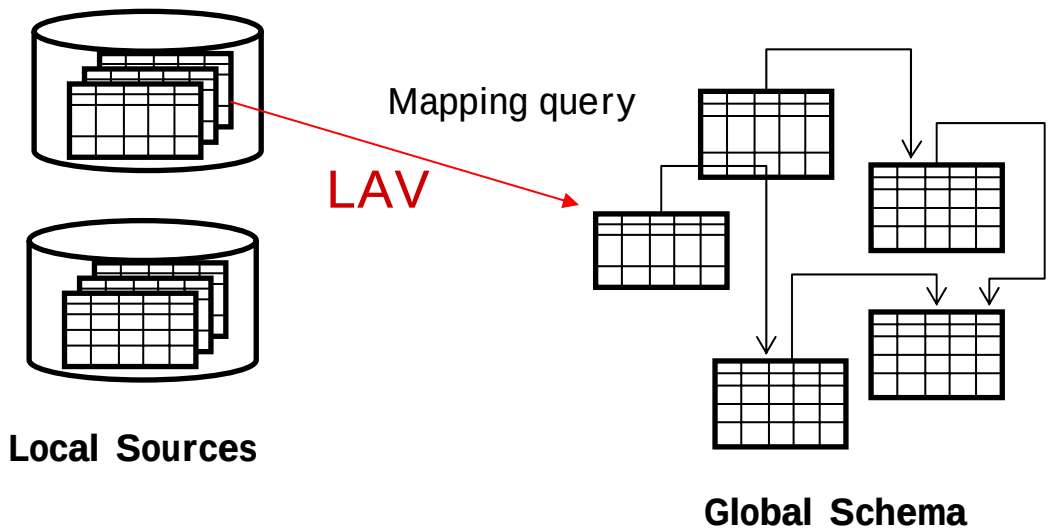
Query transformation and execution

Model and language for representing the unified schema [Lenzerini, 2002] /LAV

- Two basic approaches for specifying the mapping in a data integration system have been proposed in the literature, called local-as-view (LAV), and global-as-view (GAV)

- The LAV approach is based on the idea that the content of each local source should be characterized in terms of a view over the global schema

- It implies a-priori design of the Global Schema
- Query management difficult (more difficult than in GAV)
- It supports the extensibility of the system: adding a new source simply means enriching the mapping with a new query mapping, without other changes

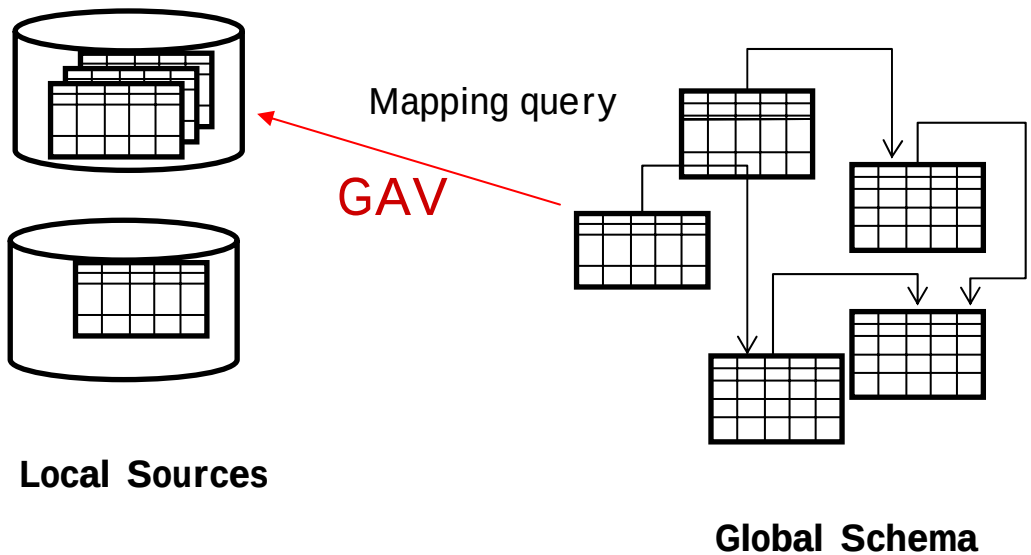


Model and language for representing the unified schema [Lenzerini, 2002] /GAV

- Global-as-view (GAV)

  From the modeling point of view, the GAV approach is based on the idea that the content of each element of the global schema should be characterized in terms of a view over the sources

- It favors the system in carrying out query processing, because it tells the system how to use the sources to retrieve data
- Extensibility of the system more difficult than in LAV



LAV approach: example

- Global Schema with all the professors
  Prof_DB (name, email, area, country)

- Local Source S1 contains professors from the database area
  S1 (name, email, country)

- Local Source S2 contains Italian professors
  S2 (name, email, area)

Create view S1 as Select name, email, country From Prof_DB Where area = “database”	Create view S2 as Select name, email, area From Prof_DB Where country = “Italy”

GAV approach: example

Source S1 contains professors from the database area

S1 (name, email, country)

Source S2 contains Italian professors

S2 (name, email, area)

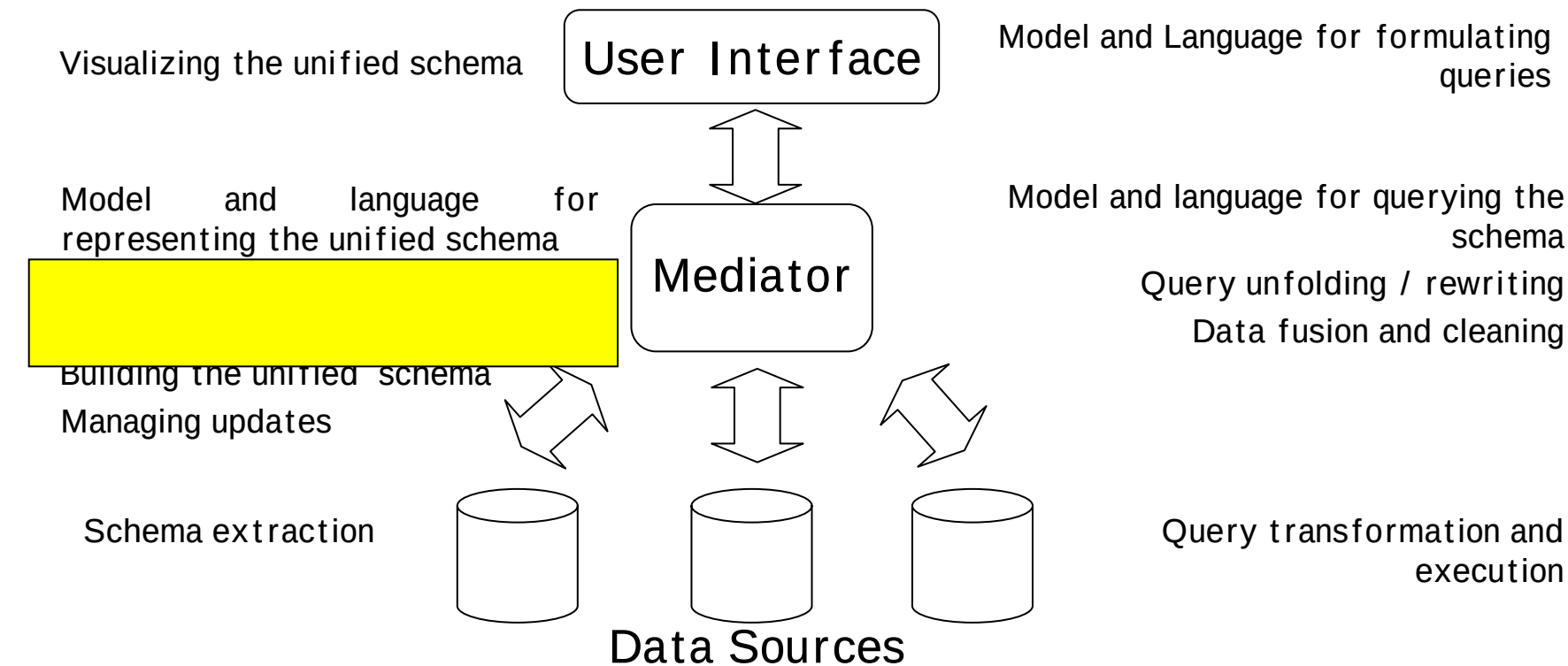
Global Schema Prof_DB (name, email, area, country)

(S1 JOIN S2) UNION {tuples of S1 and not in S2 with null values for the area attribute and tuples of S2 and not in S1 with null values in the country attribute}

Languages for representing the unified schema

- Several languages have been exploited for representing the unified schema:
  - Relational Languages
    - SQL
  - Object Oriented Models
    - ODL
    - UML
  - Semistructured Languages
    - RDF/RDF-S
    - XML

Mediators – relevant challenges

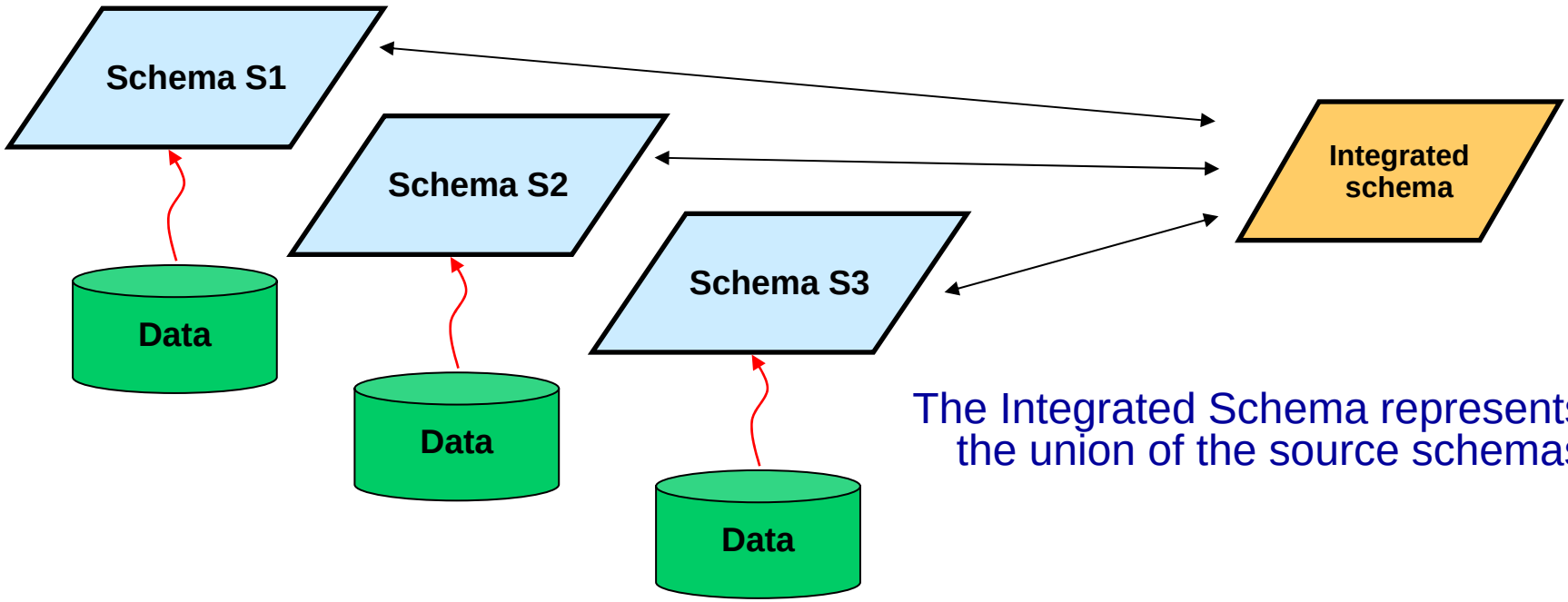


Techniques for matching and mapping unified schema and local sources

- Techniques for matching and mapping unified schema and local sources have been developed in both GAV and LAV approaches
  - Schema matching takes two schemas as input and produces a mapping between elements of the two schemas that correspond semantically to each other [Rahm 2001]
  - Instance matching, a.k.a. record linkage, is the task of quickly and accurately identifying records corresponding to the same entity from one and more data sources [Castano 2008]

Schema mapping vs data integration

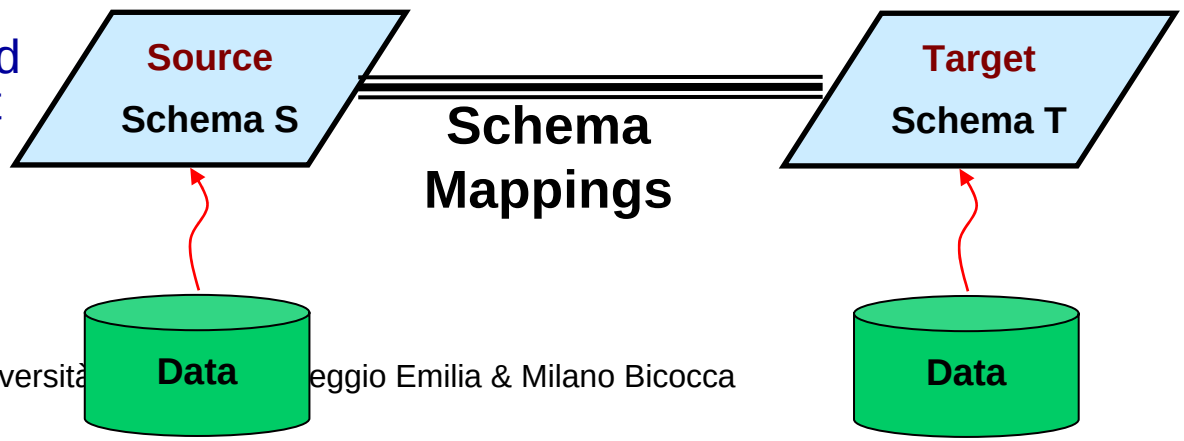
Schema Integration



The Integrated Schema represents the union of the source schemas

Schema Mapping

Schemas pre-exist and may describe different data semantics

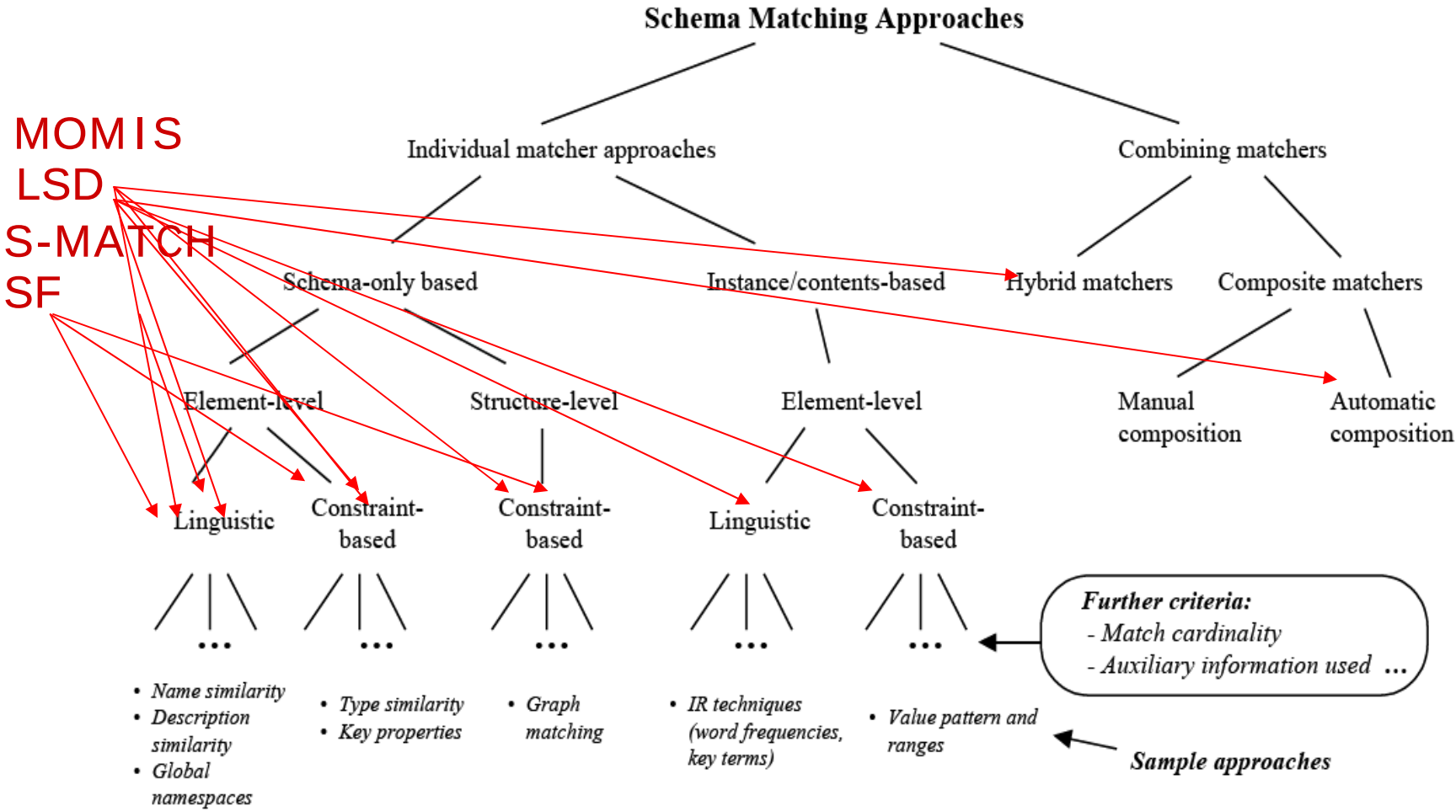


Schema matching and Instance matching

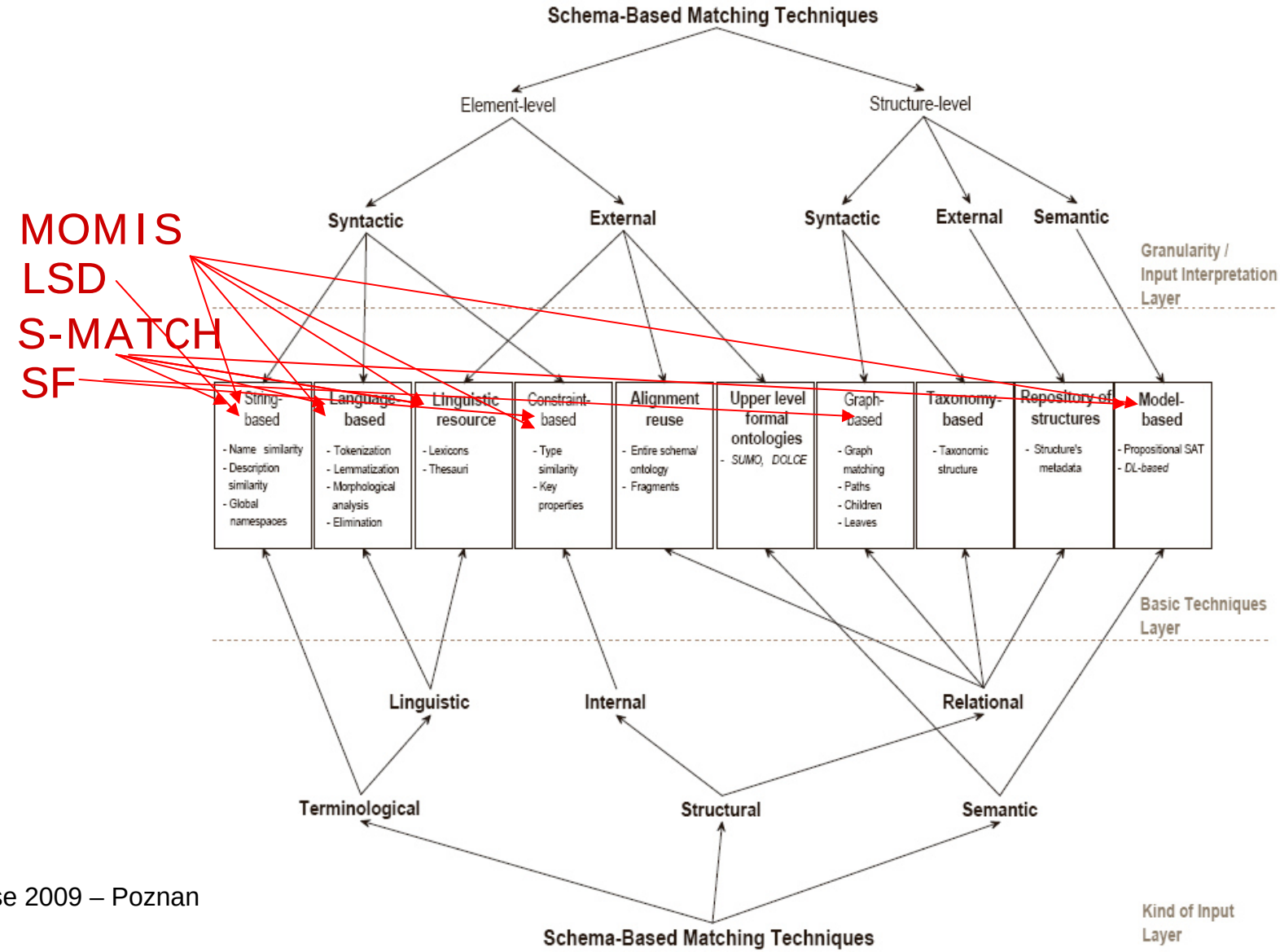
- Schema and Instance matching have been extensively investigated by the research community, since they represent the core of data integration systems
  - Schema matching typically relies on:
    - Analysis of the schema labels
    - Analysis of the schema structures
  - Instance matching is based on techniques for entity resolution providing each object with an identifier
- Schema matching is slightly affected by source dynamics (the data source schema is typically fixed)
- Instance analysis is computationally a heavy task, since involves a great number of elements

Schema matching - classification

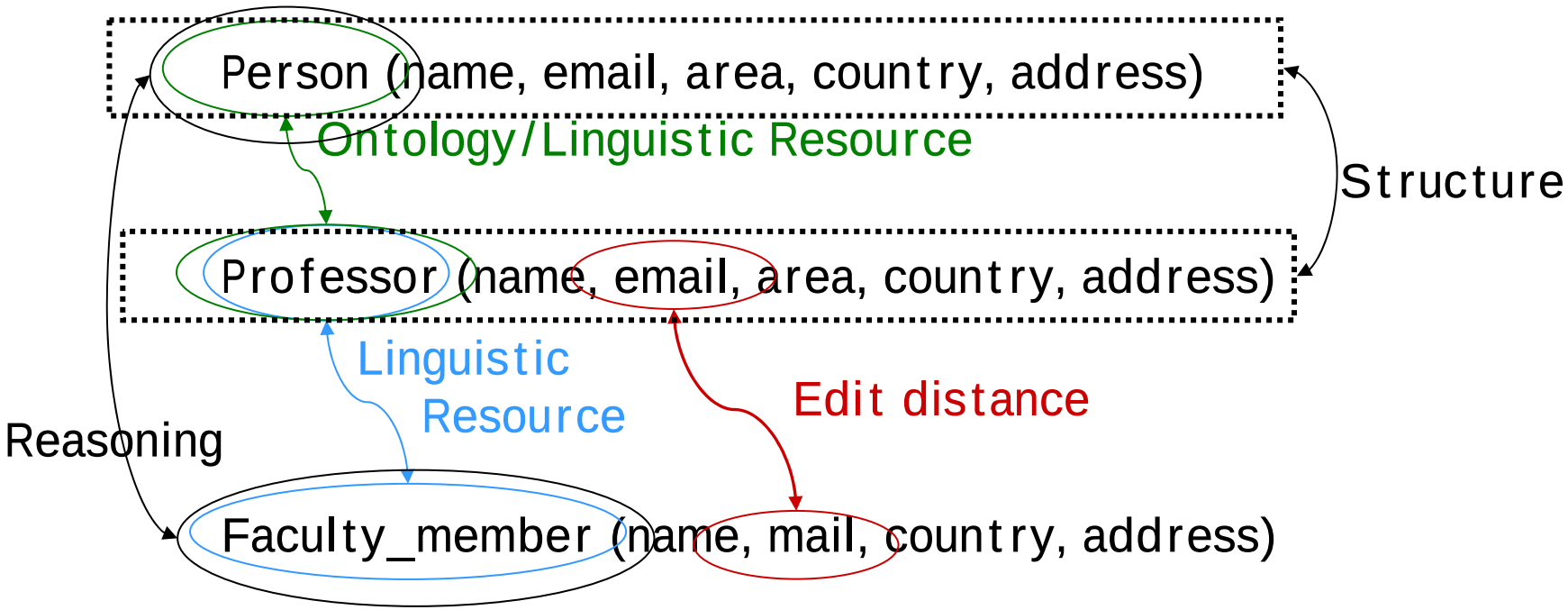
- Schema matching may be classified according to several criteria In [Ramh 2001] (data base area)



Schema Matching – classification [Shvaiko 2005] (ontology alignment area)



Schema Matching: a simple example



Schema Matching - classification

- In [Noy 2004] a survey of ontology-based integration techniques is shown
  - Mapping discovery
    - Using shared ontologies (MOMIS, S-Match)
    - Using heuristics and machine-learning (LSD, SF)
  - Representation of mappings
    - GaV and LaV approaches
  - Reasoning with mappings
    - Applying description Logics techniques (MOMIS, S-Match)

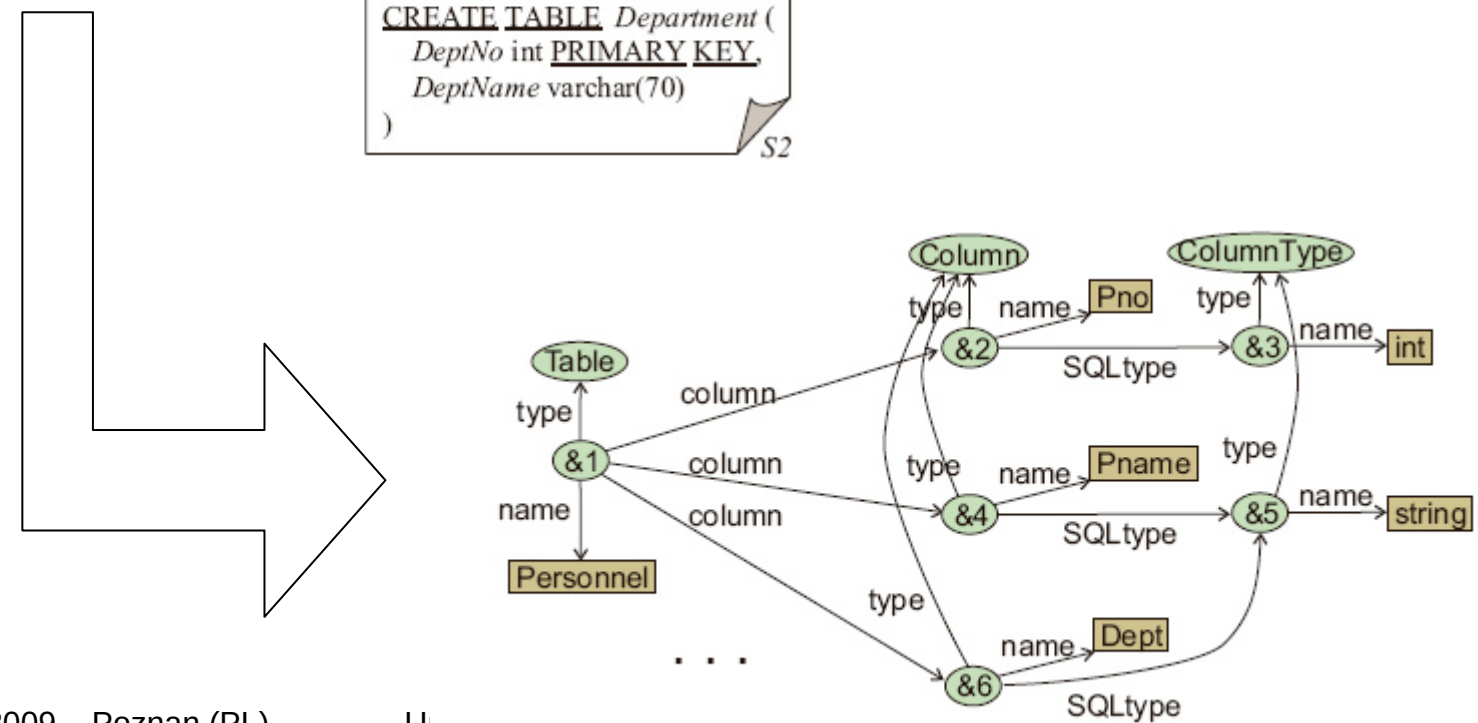
Example - Similarity Flooding (SF)[Melnik 2002]

- It is a simple structural algorithm that can be used for matching diverse data structures
  - Such data structures, which we call *models*, may be data schemas, data instances, or a combination of both
- First, the models to be matched are converted into directed labeled graphs.
  - These graphs are used in an iterative fixpoint computation whose results tell us what nodes in one graph are similar to nodes in the second graph
- For computing the similarities, the intuition is that elements of two distinct models are similar when their adjacent elements are similar
  - Whenever any two elements in models G1 and G2 are found to be similar, the similarity of their adjacent elements increases
- Depending on the particular matching goal we then choose a subset of the resulting mapping using adequate filters are chosen
  - After the algorithm run, a human to check and if necessary adjust the results is expected
- The Protoplasm matcher [Bernstein et al 2004] is based on this algorithm

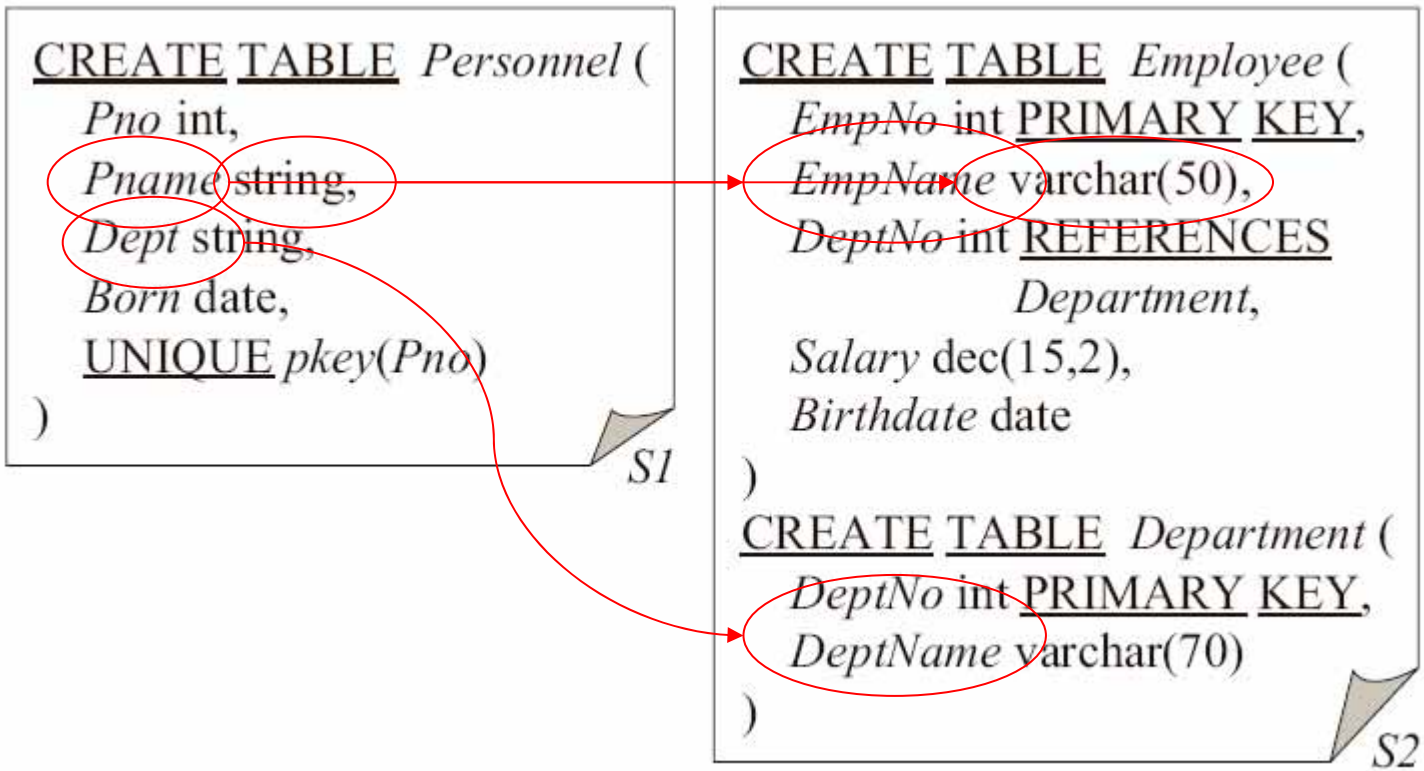
Example - Similarity Flooding (2)

```
CREATE TABLE Personnel (
    Pno int,
    Pname string,
    Dept string,
    UNIQUE pkey(Pno)
)
```
S1

```
CREATE TABLE Employee (
    EmpNo int PRIMARY KEY,
    EmpName varchar(50),
    DeptNo int REFERENCES
                Department,
    Salary dec(15,2),
    Birthdate date
)
CREATE TABLE Department (
    DeptNo int PRIMARY KEY,
    DeptName varchar(70)
)
```
S2



Example - Similarity Flooding (3)



Pname is Similar to EmpName → Personnel.Pname is similar to Employee.EmpName

Personnel.Pname is similar to Employee.EmpName → string is similar to varchar

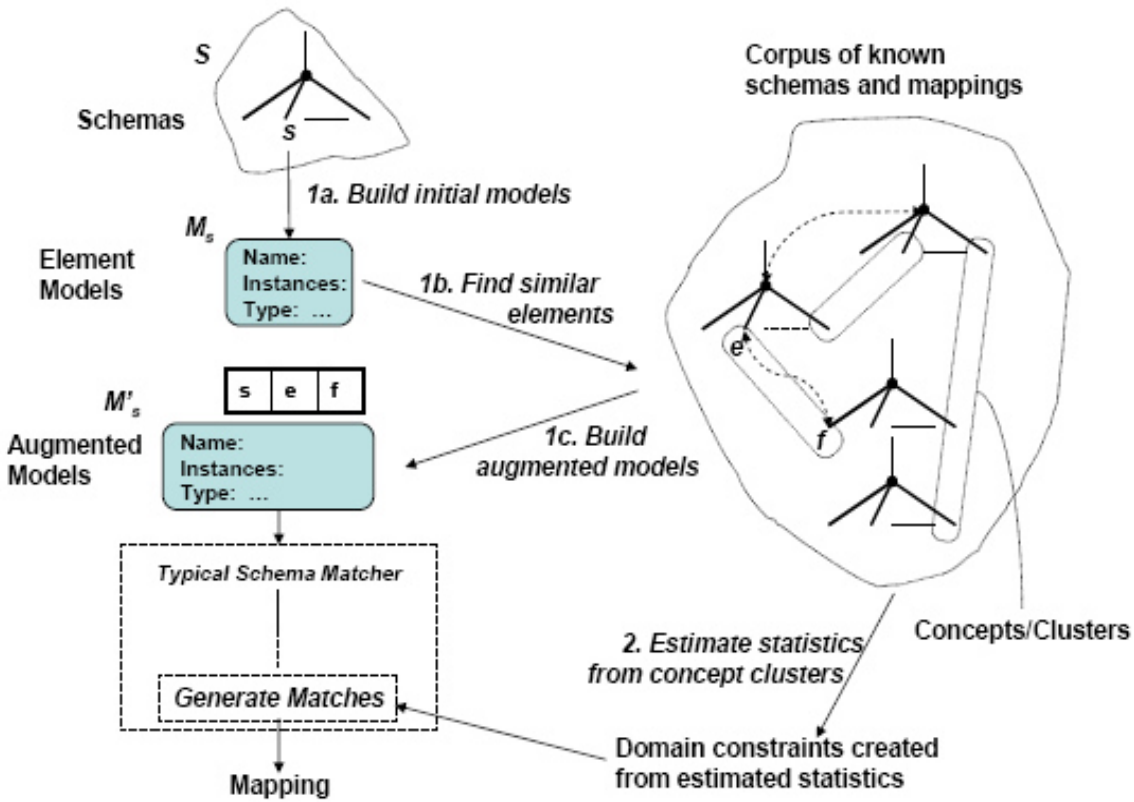
string is similar to varchar → Personnel.Dept is similar to Department.DeptName

Example – Corpus-based schema matching [Madhavan, 2005]

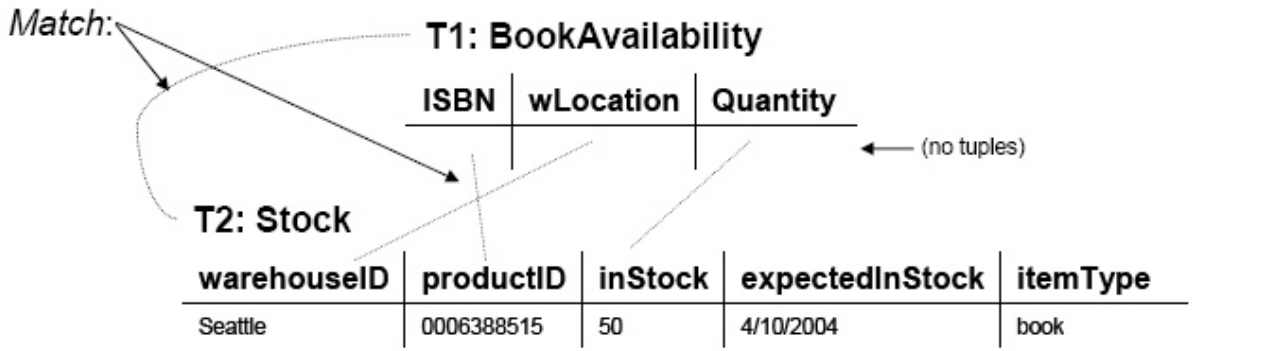
- A corpus of schemas and mappings between some schema pairs is provided
- The Corpus-Based Augment Method leverages this variety of representations by using the corpus to add information to each element's model

In particular:

- Domain concepts and their representational variations
- Relationships between concepts
- Domain constrains



Example – Corpus-based schema matching (2)



Mapping: **BookAvailability**(i, w, q) → **Stock**(i, w, q, NULL, "book")

(a) Matches and mappings between BookAvailability and Stock

T3: BookStock

ISBN	Warehouse	Qty
1565115147	Atlanta	140

T4: ProductAvailability

productID	warehouseID	Quantity	bookORcd
078878983X	Seattle	5354	book

(b) Other schemas about product availability

Combining the evidence in T3 with T1 and T4 with T2 better enables us to match T1 with T2:

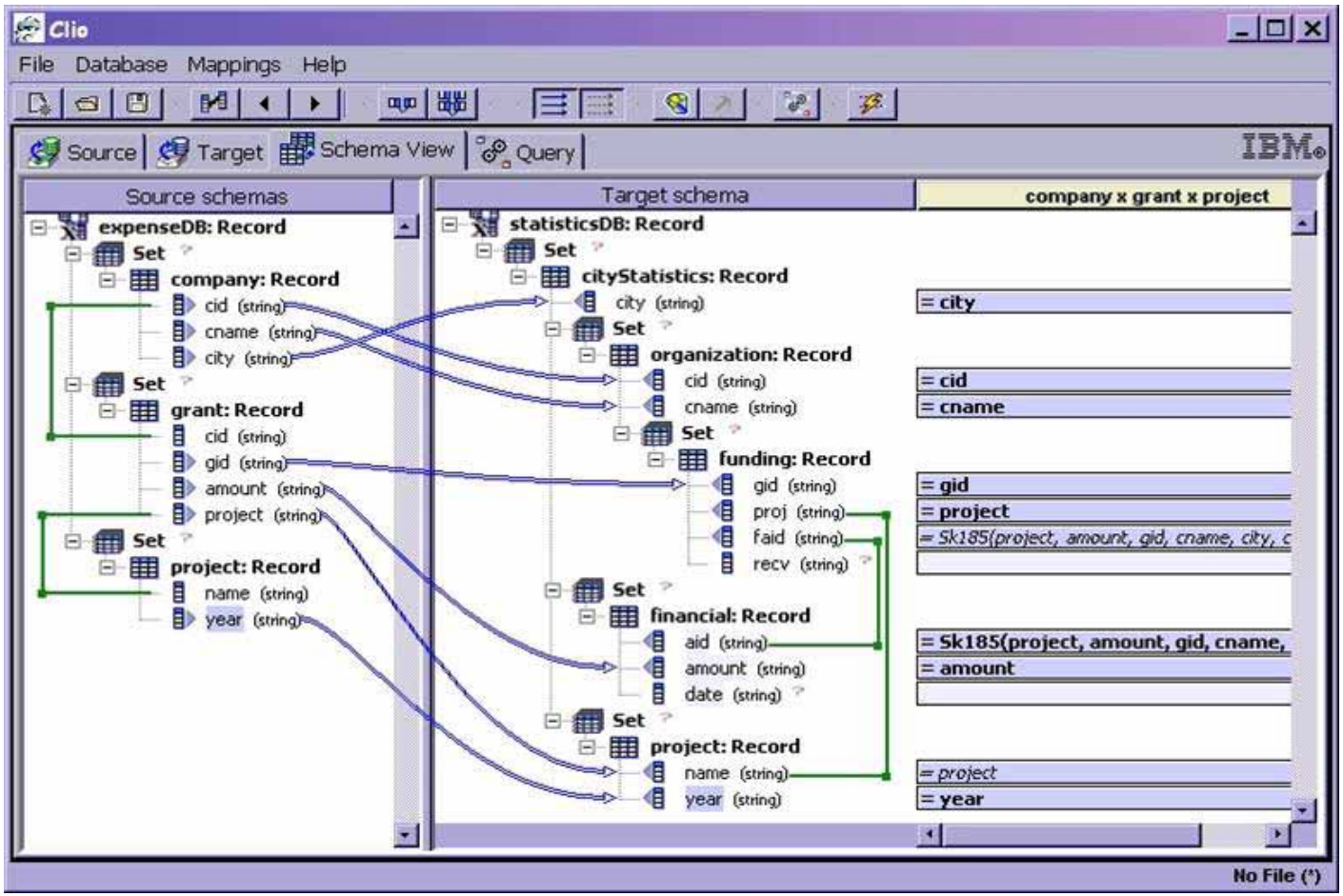
1. there is increased evidence for particular matching techniques, e.g., alternative names for an element,
2. there is now evidence for matching techniques where there earlier was none, e.g., considering T3 with T1 includes data instances (tuples) where there were none initially

Schema mapping: The Clio Project

http://queens.db.toronto.edu/project/clio/index.php

- The Clio project [Fagin et al 2009] is a joint project between the IBM Almaden Research Center and the University of Toronto begun in 1999
- Clio's goal is to radically simplify information integration, by providing tools for the conversion of data between representations
- Clio pioneered the use of schema mappings that describe the relationship between data in two heterogeneous schemas
- From this high-level, non-procedural representation, Clio can automatically generate either a view, to reformulate queries against one schema (source) into queries on another for data integration (target schema), or code, to transform data from one representation to the other for data exchange

Schema mapping: The Clio Project



Clio exploits the schema and its constraints to generate a set of alternative mappings

Schema Matching

Projects

name	status
PIX	Active
E-services	Active

Grants

gid	project	recipient	manager	supervisor
g1	PIX	AT&T	Fernandez	Belanger
g2	PIX	AT&T	Srivastava	Belanger
g3	E-services	Bell-labs	Benedikt	Hull

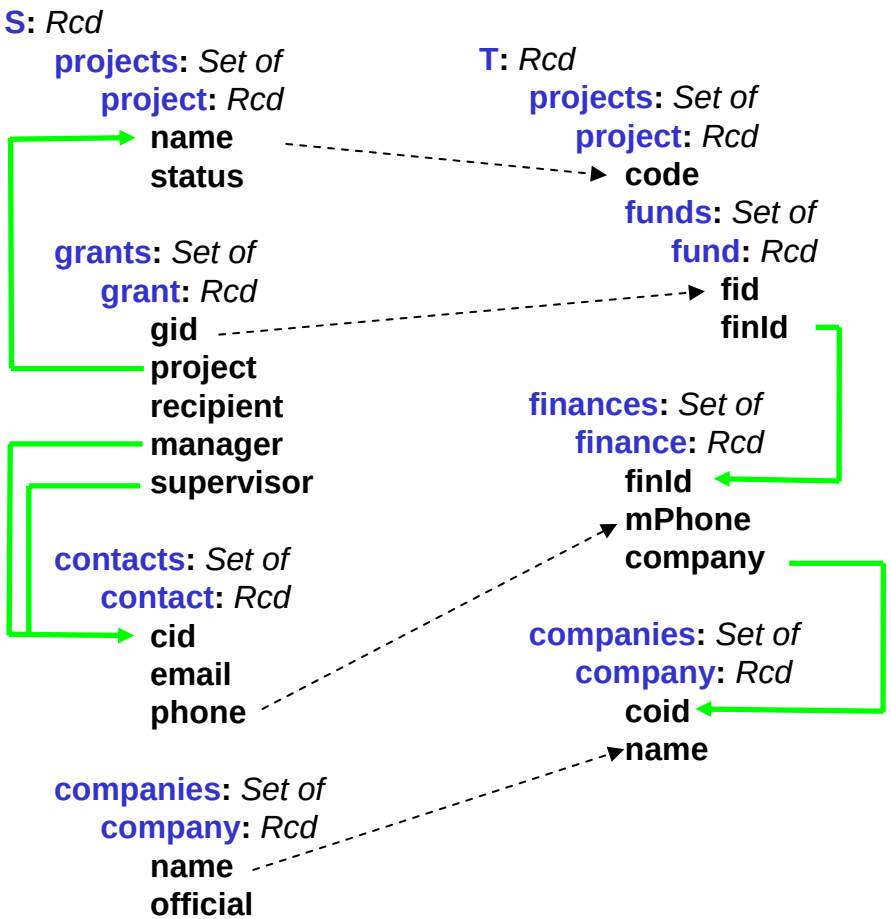
Contacts

cid	email	phone
Benedikt	benedikt@research.bell-labs.com	5827766
Hull	hull@research.bell-labs.com	5824509
Srivastava	divesh@research.att.com	3608776
Belanger	dgb@research.att.com	3608600
Fernandez	mff@research.att.com	3608679

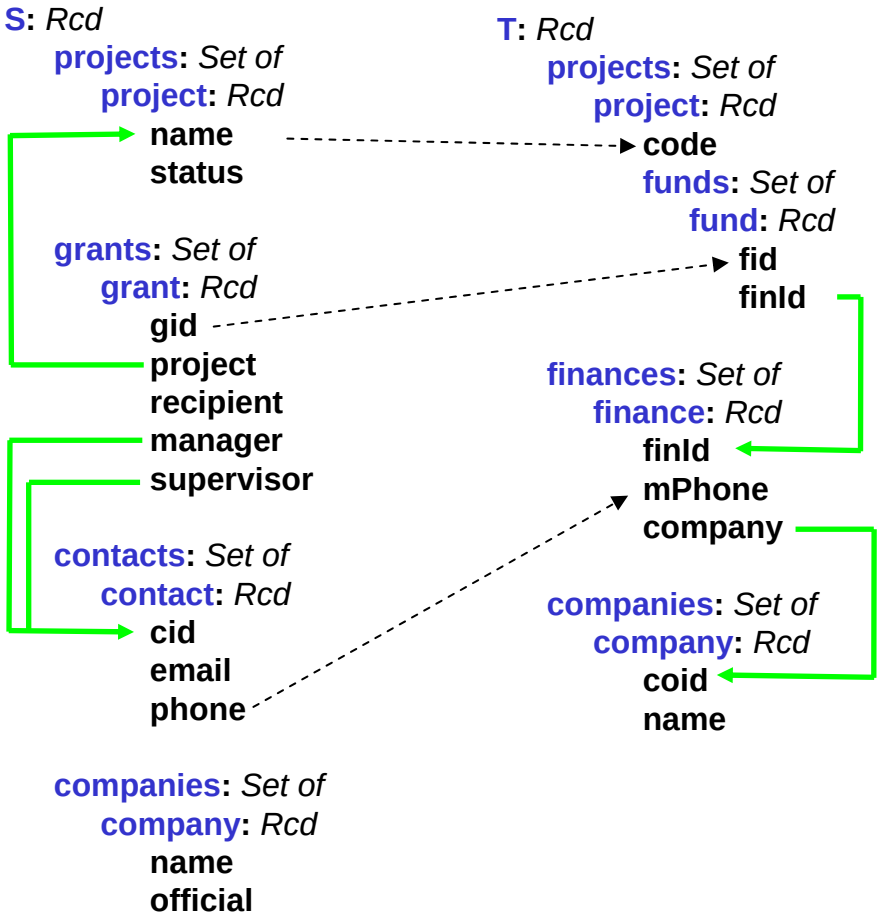
Companies

name	official
AT&T	AT&T Research Labs
Lucent	Lucent Technologies, Bell Labs Innovations

- Perform schema matching
  - By human or by a matching tool
  - Result: **correspondences** (the black lines)
    - ✓ But no clear semantics



From Correspondences to Mappings



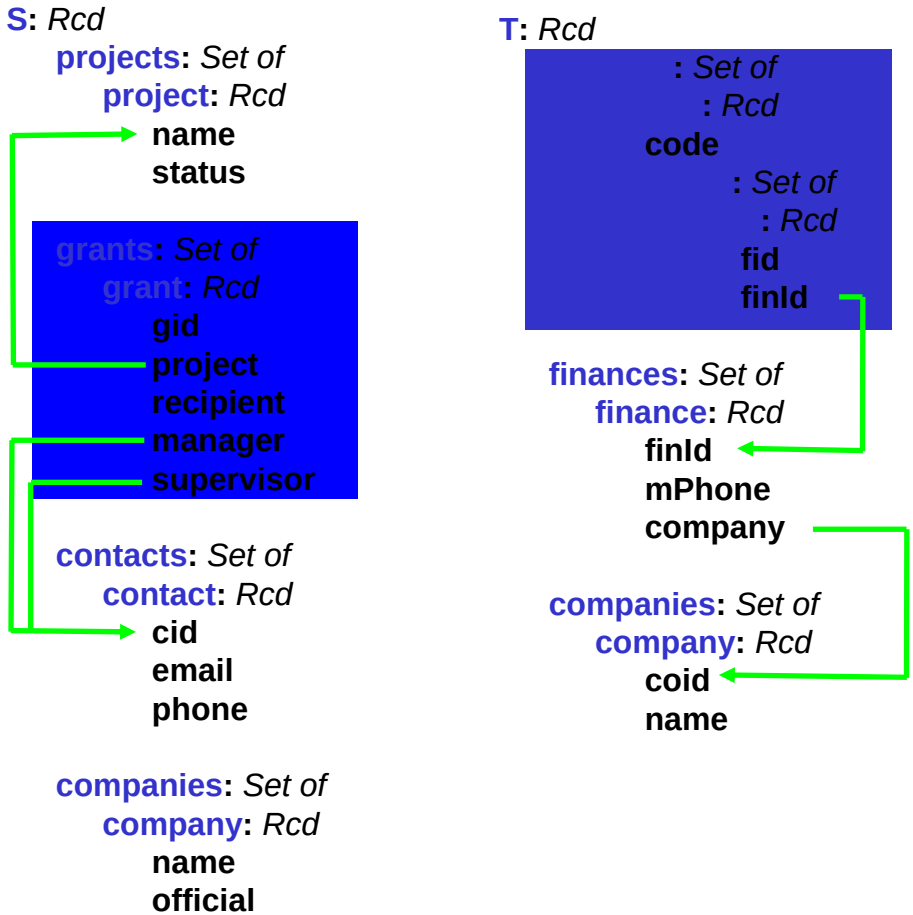
- Goal: Given correspondences
  - Discover intended data semantics
  - Generate mappings

PROJECT(na,st), **GRANT**(gid,na,re,ma,su), **CONTACT**(ma,em,ph) →

PROJECT(na,FUND), **FUND**(gid,finId), **FINANCE**(finId,ph,company)

Thanks to Yannis Velegrakis – University of Trento

Structural Associations



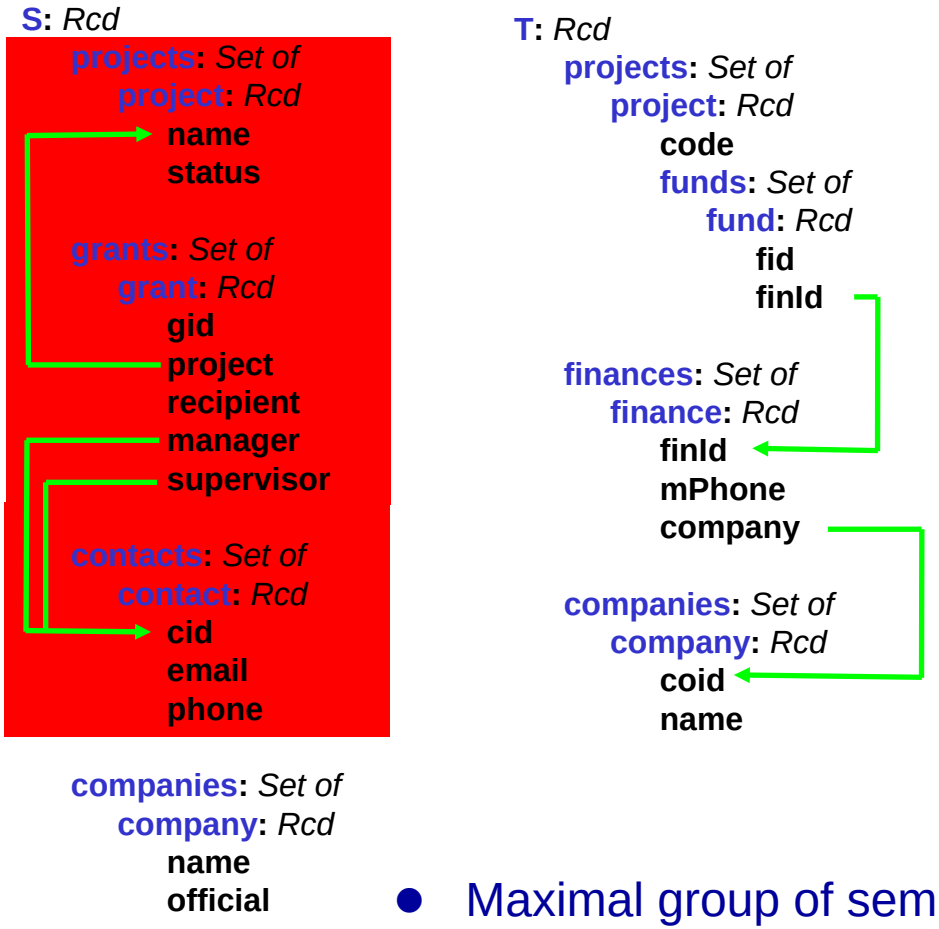
Grants

gid	project	recipient	manager	supervisor
g1	PIX	AT&T	Fernandez	Belanger
g2	PIX	AT&T	Srivastava	Belanger
g3	E-services	Bell-labs	Benedikt	Hull

- **Association:** A set of semantically related elements
  - Expressed as a query on the schema
- **Structural Association**: Association based on the structure alone

Thanks to Yannis Velegrakis – University of Trento

Logical Associations



Projects

name	status
PIX	Active
E-services	Active

Grants

gid	project	recipient	manager	supervisor
g1	PIX	AT&T	Fernandez	Belanger
g2	PIX	AT&T	Srivastava	Belanger
g3	E-services	Bell-labs	Benedikt	Hull

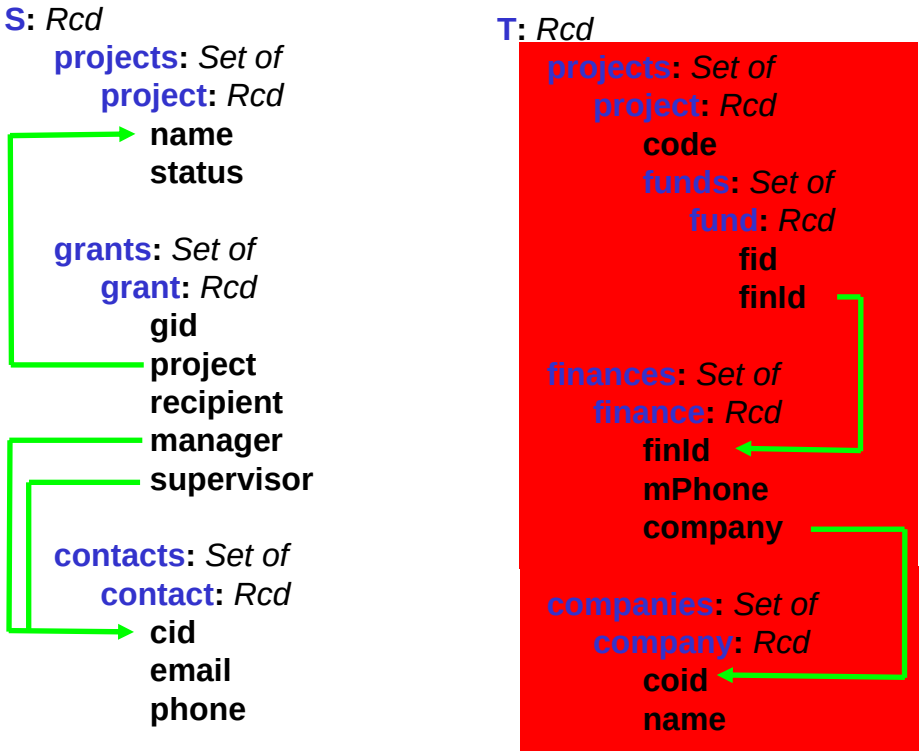
Contacts

cid	email	phone
Benedikt	benedikt@research.bell-labs.com	5827766
Hull	hull@research.bell-labs.com	5824509
Srivastava	divesh@research.att.com	3608776
Belanger	dgb@research.att.com	3608600
Fernandez	mff@research.att.com	3608679

- Maximal group of semantically related elements
- Generated by chasing structural associations
  - Relational chase [MMS79] extended for nested schemas [PT'99]
  - Adapted to deal with XML-Schema union types

Thanks to Yannis Velegrakis – University of Trento

Logical Association Generation



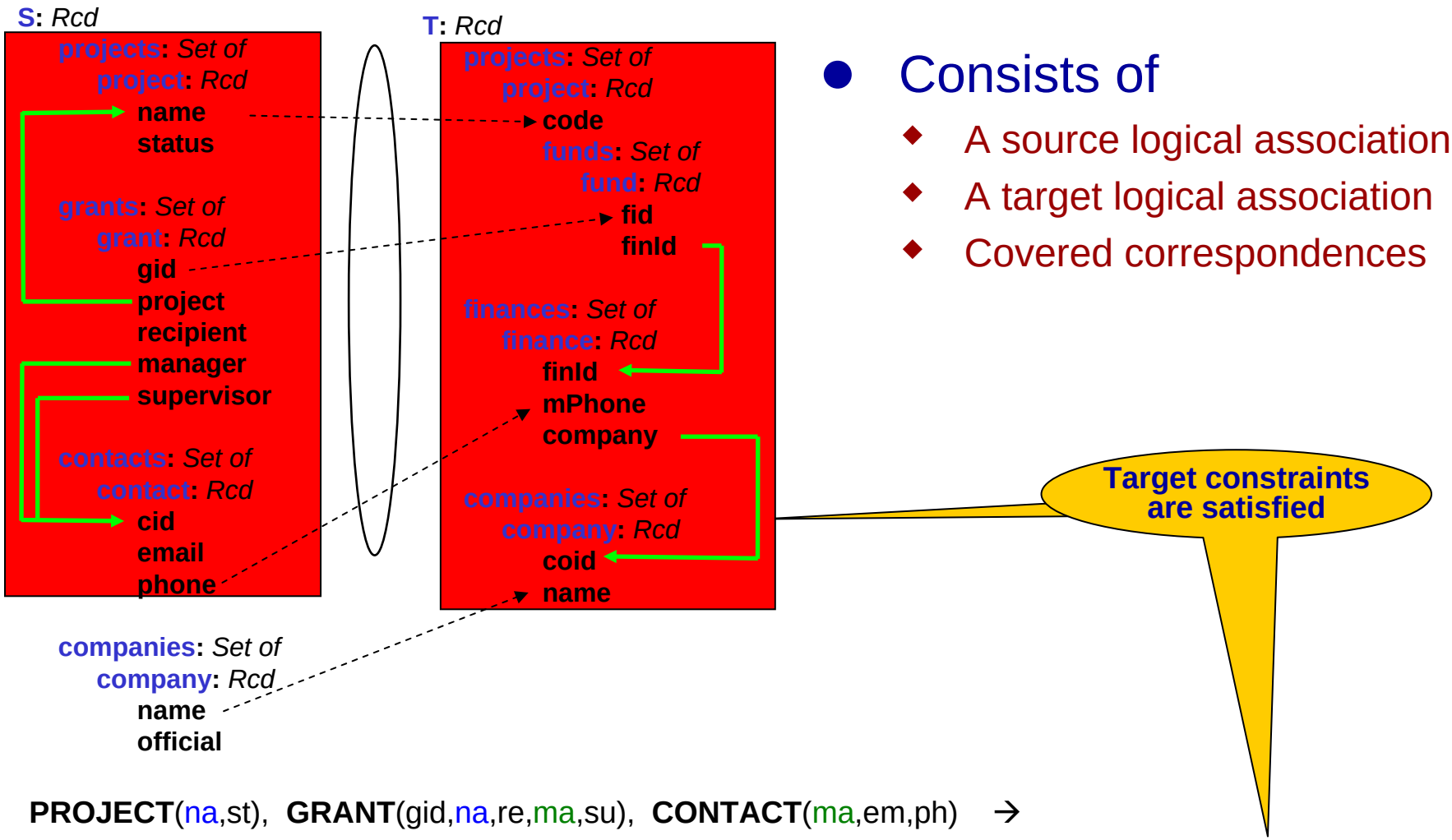
PROJECT(na,FUND), **FUND**(gid,finId) →**FINANCE**(finId,ph,company)

FINANCE(finId,ph,company) →**COMPANY**(company,name)

PROJECT(na,FUND), **FUND**(gid,finId), **FINANCE**(finId,ph,company),**COMPANY**(company,name)

Thanks to Yannis Velegrakis – University of Trento

Semantically Complete Mappings



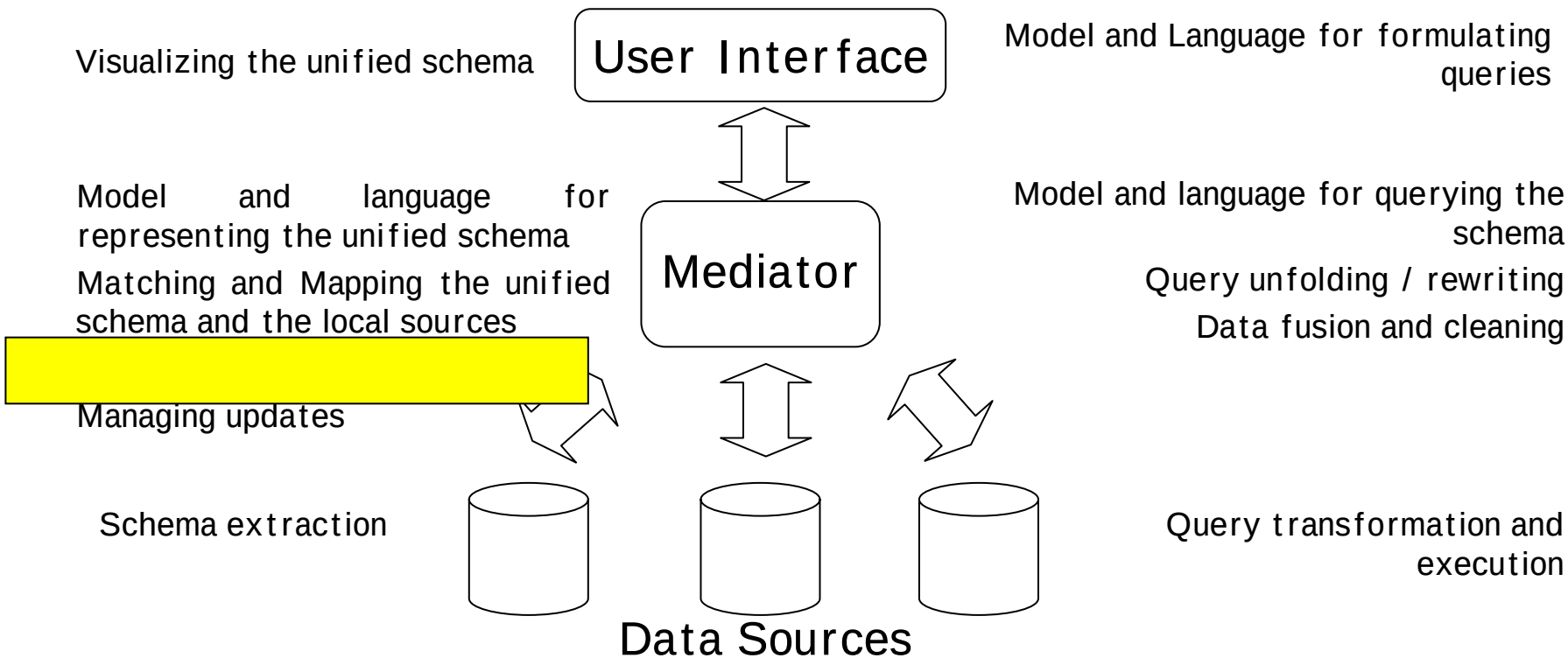
- Consists of
  - A source logical association
  - A target logical association
  - Covered correspondences

PROJECT(na,st), **GRANT**(gid,na,re,ma,su), **CONTACT**(ma,em,ph) →

PROJECT(na,FUND), **FUND**(gid,finId), **FINANCE**(finId,ph,company), **COMPANY**(company,name)

Thanks to Yannis Velegrakis – University of Trento

Mediators – relevant challenges



Data Integration: Building the unified schema

- Merging schema by means of mappings
  - In [Pottinger 2008]
    - A set of requirements for a mediated schema are proposed
      - Completeness: All information in the source schema should be exposed in the mediated schema
      - Overlap preservation: Each of the overlapping elements specified in the input mapping is exposed in a mediated schema relation
      - Extended overlap preservation: Source-specific elements that are associated with a source's overlapping elements are passed through to the mediated schema
      - Normalization: Independent entities and relationships in the source schemas should not be grouped together in the same relation in the mediated schema
      - Minimality: If any element of the mediated schema is dropped then the mediated no longer satisfies (i) – (iv) above
    - A normal form for mediated schemas and mappings is defined
    - An algorithm that generates normal form schema is provided
- Creating clusters of similar relations/classes
  - MOMIS in [Bergamaschi 2001], Artemis is exploited [Castano 2001]

Mediators – relevant challenges

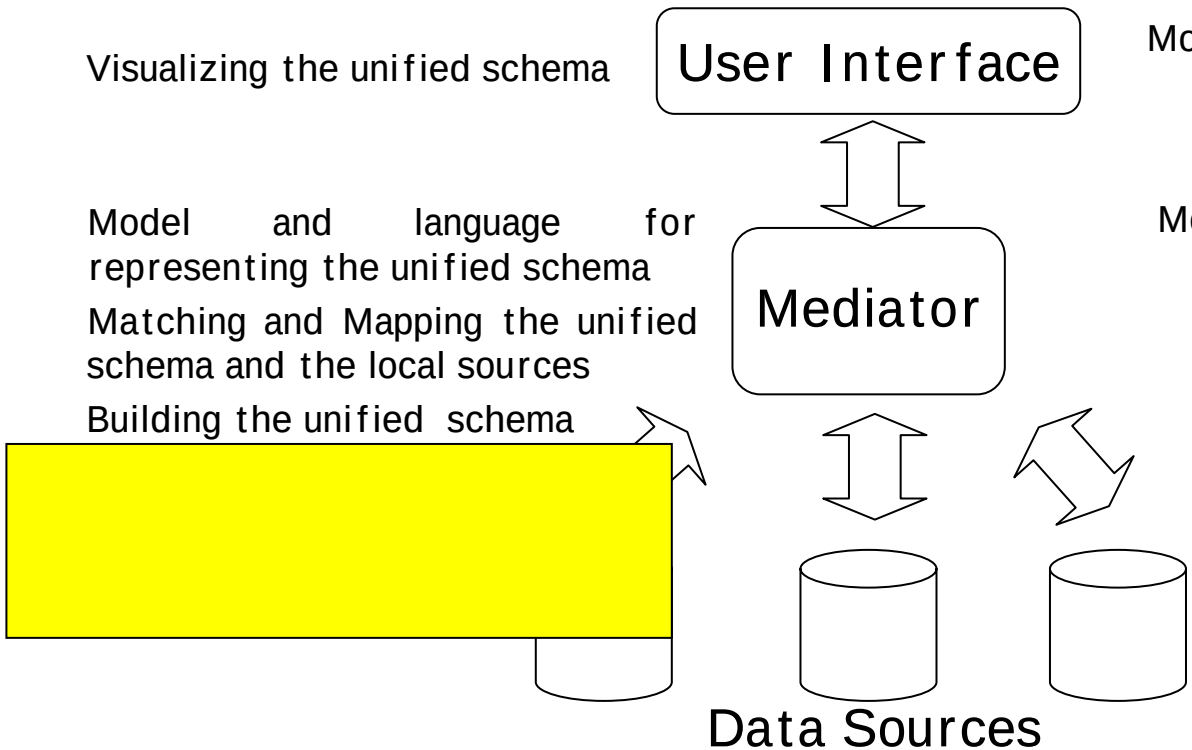
Publishing Phase

Visualizing the unified schema

Model and language for representing the unified schema

Matching and Mapping the unified schema and the local sources

Building the unified schema



Querying Phase

Model and Language for formulating queries

Model and language for querying the schema

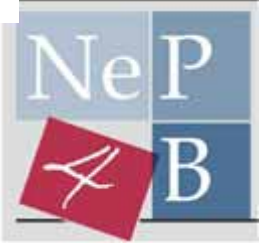
Query unfolding / rewriting

Data fusion and cleaning

Query transformation and execution

Managing updates in data integration systems

- Changes occur when
  - The data sources change
  - The users / applications using the integrated data sources need a different conceptualization
- For data integration systems following the LaV approach, managing dynamics means to add / to modify / to remove at the local source level
  - A change in a local data source does not affect the other sources
- For systems following the GaV approach, a change in a local data source may change the unified view
  - Consequently all the mappings between the global view and the local sources may in principle be affected by a source change
    - Evolution / versioning



Developed systems

Commercial Tools

Data Integration in the Enterprise

- Several papers analyzed the issues related to the deployment of the data integration technology in business environment [Halevy et al. 2005]
  - Scalability and performances
  - Robust applications
  - Role of the user
  - Integration with other tools middleware
  - Lack of explicit semantics in real database

Data Integration in the Enterprise

- A recent paper [Bernstein et al. 2008] described the kinds of information integration tools used in practice and reviewed the core technologies:
  - Information integration tool
    - Data Warehouse Loading (ETL tools)
    - Virtual Data Integration (EII tools)
    - Message Mapping (EAI tools)
    - Object to Relational Mappers
    - Document Management
    - Portal Management
  - Core technologies
    - XML
    - Schema standards
    - Schema mapping / matching
    - Keyword search
    - Information extraction
    - Dynamic Web Technologies (RSS)

Information Integration issues for Enterprises [Haas 2007]

- Information integration is a complex process, with the following major tasks:
  - **Understanding:** discovering relevant information and analysing it to assess quality and to determine statistical properties
  - **Standardization:** Determining the best way to represent the integrated information
    - Designing the "target" schema, deciding at the field level what the standard representation should be, and defining the terminology and abbreviations to use
    - Other rules specifying how to cleanse or repair data
  - **Specification:** the artifacts that will control the actual execution are produced. The techniques and technologies used for specification are intimately linked to the choice of execution engine(s)
  - **Execution:** Integration is obtained via materialization, federation and/or indexing
    - ***Materialization*** creates and stores the integrated data set; this may be thought of as eager integration. There are many techniques for materialization. *Extract/Transform/Load (ETL), Replication, Caching*
    - ***Federation*** creates a virtual representation of the integrated set, only materializing selected portions as needed; Federation is a form of mediation
    - ***Search*** creates a single index over the data (used for unstructured data)

Schema Mapping
Talend Open Studio

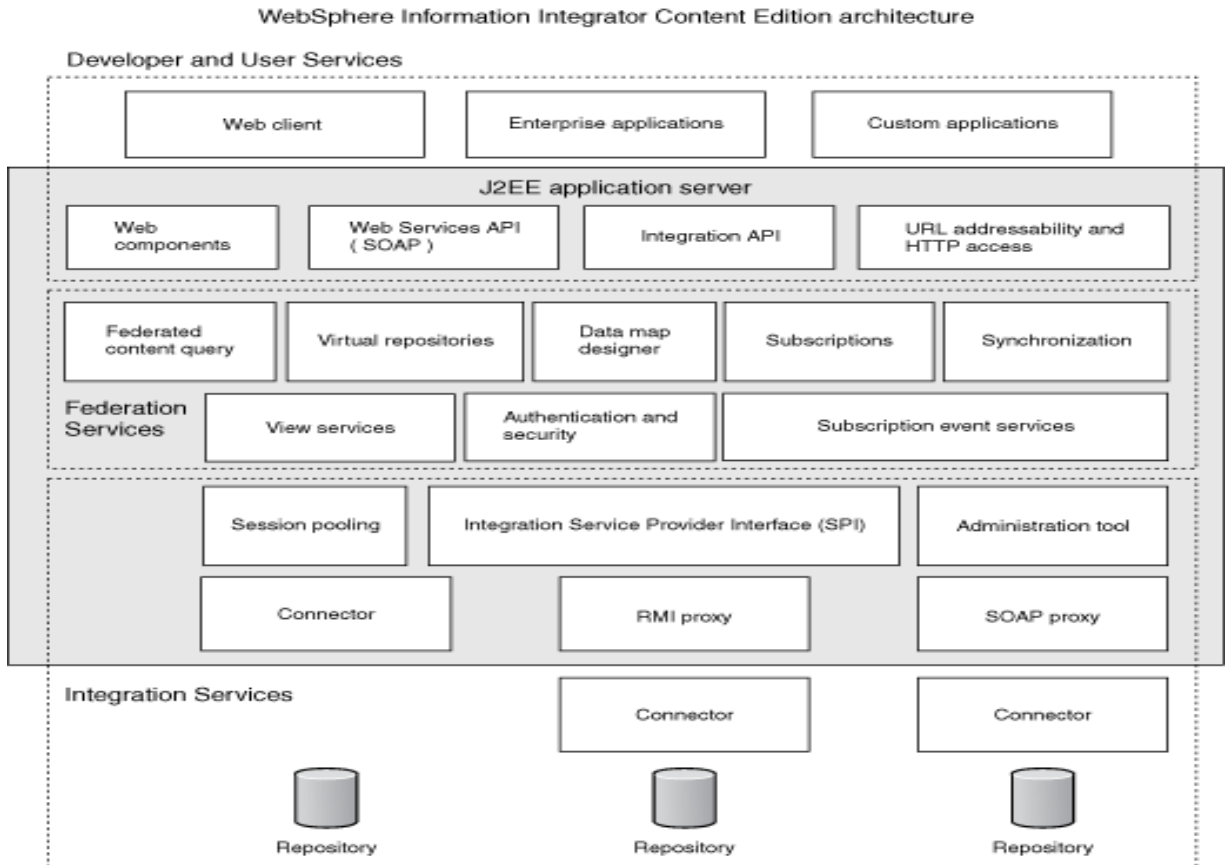
- The first **Open Source** data integration product
- Developed by Talend, a French start-up based in Paris
- Mainly an **ETL** tool, supports also an **ELT** approach
- The environment is based on **Eclipse**
- A **code-generating** product: it generates Java or Perl (plus SQL) code, deployable as required
- Easy to encapsulate and deploy a job as a **web service** or into third party applications

Talend Open Studio

- Typical **Graphical User Interface**: drag and drop onto a palette, a visual SQL builder, a source code editor, debugging facilities etc.
- **Data cleansing** and **data profiling** capabilities
- **Parallelisation** functions offered
- More than **400 connectors** are available, including application-specific connectors for *SAP, salesforce.com, SugarCRM, Microsoft Dynamics* and others
- Native support for more than **30 databases**, ODBC/JDBC can be used otherwise
- Access to several **file types**, including CSV, spreadsheets, unstructured data, zipped files and XML

WebSphere Information Integrator Content Edition 8.4

http://www-01.ibm.com/software/data/content-management/content-integrator/



WebSphere Information Integrator Content Edition 8.4 architecture

- *WebSphere Information Integrator Content Edition 8.4 (IICE)* is a software of data integration, developed by IBM and uses Java language

- This software is able to integrate enterprise applications with various types of content such as documents, images, audio, video, structured and semi-structured data stored in many heterogeneous environments

- In order to use all the features and services of this integration system is necessary to install it on an application server
  - WebSphere Application Server 6.1 (WAS)

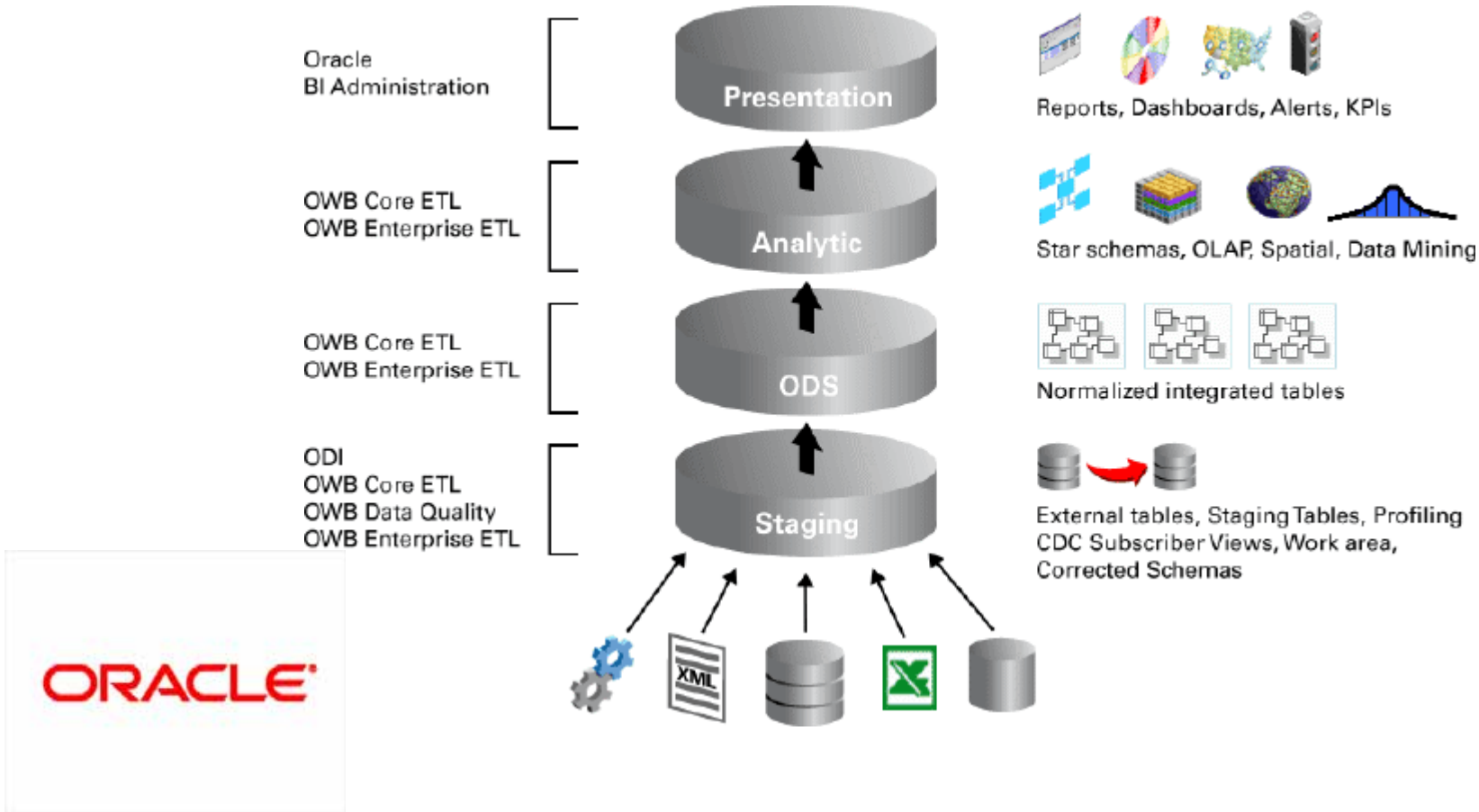
WebSphere Information Integrator Content Edition 8.4 architecture

- WebSphere Information Integrator Content Edition architecture consists of three main levels:
  - **Services for the developer and user** include components to view and use data (the content)
  - **Association Services** to combine and add values to the integrated content
  - **Integration Services** to provide efficient access to the content of the natives repositories

WebSphere Information Integrator Content

- Information Integrator enables the integration of heterogeneous structured and semi-structured data sources thanks to interfaces like ODBC, JDBC, etc.
  - The integration process produces a virtual database that communicates with the data sources through forms, one for each type of data sources
  - allows the user to query the virtual database by a command-line interface
  - The sources are examined by means of two functions: getContentLookup() which returns information concerning the documents and getContent() which returns the contents of the documents
  - It does not provide any mechanisms for resolving data conflicts

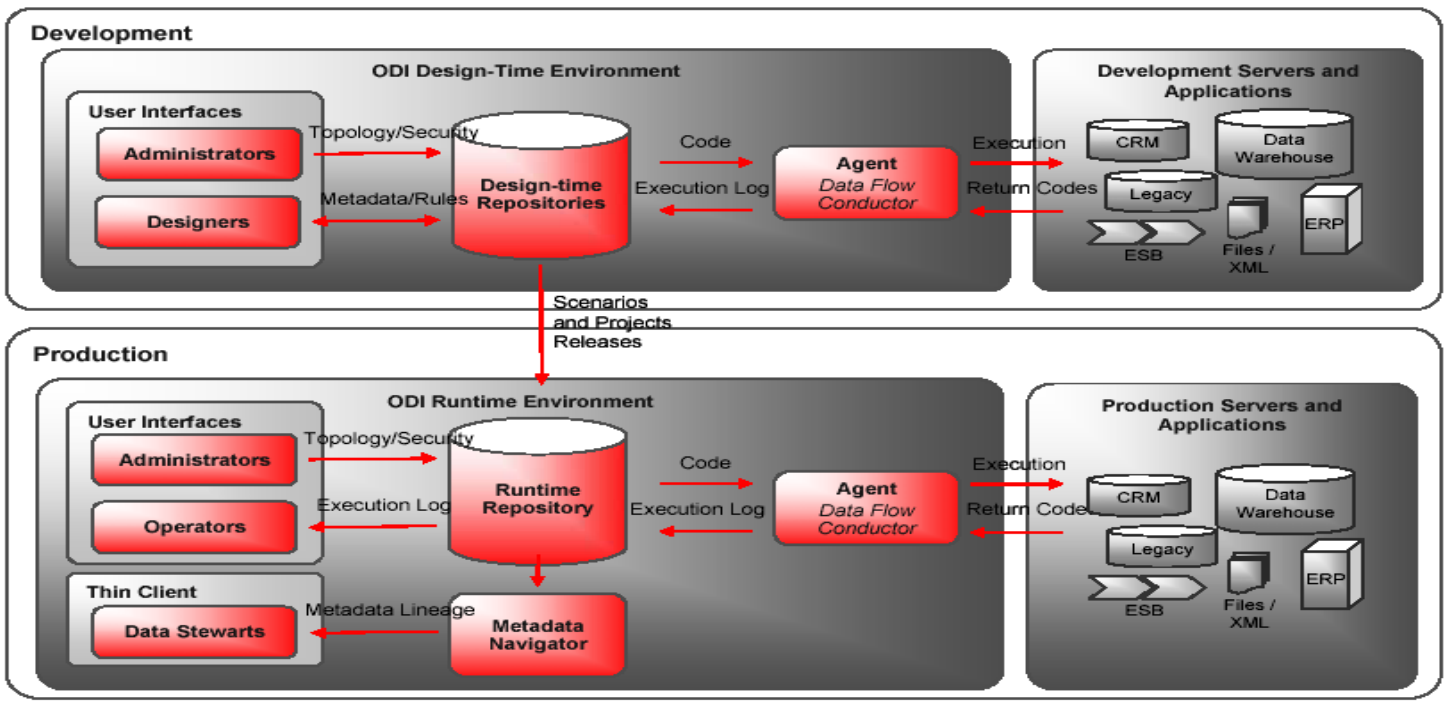
Oracle Data Integrator (ODI)

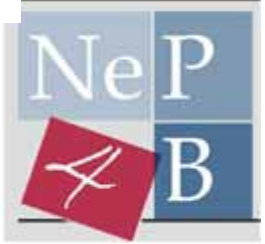


http://www.oracle.com/technology/products/oracle-data-integrator/index.html

Oracle Data Integrator (ODI) architecture

- *Oracle Data Integrator* is a data integration system that allows you to develop solutions for advanced datawarehouse with an innovative ETL approach, performing transformations on the database target
- Oracle Data Integrator has a modular architecture and its repositories are accessible in client-server mode by Java components



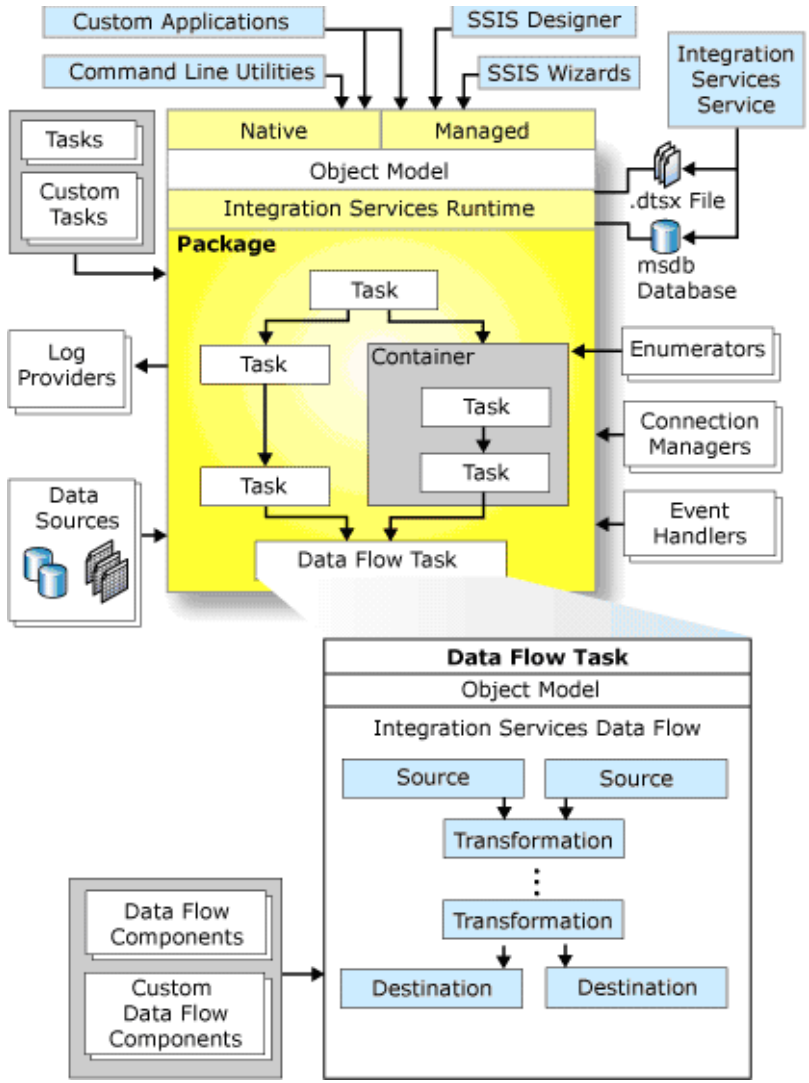


Oracle Data Integrator (ODI) architecture

- Oracle Data Integrator (ODI) has 4 modules that can be installed on any platform supporting the Java Virtual Machine
  - The **Designer** is the main module of the integration system. It defines rules for processing data
  - The **Operator** is designed to manage and control the integration flow, report errors by the log file and the data involved
  - The **Topology Manager** defines the physical and logical architecture of the whole integration system
  - The **Security Manager** is important to manage user profiles and access privileges
- Oracle Data Integrator provides access to the repository metadata from any web browser

Microsoft SQL Server 2005 Integration Services

http://www.microsoft.com/italy/server/sql/technologies/integration/default.mspx



Microsoft

Microsoft SQL Server 2005 Integration Services - architecture

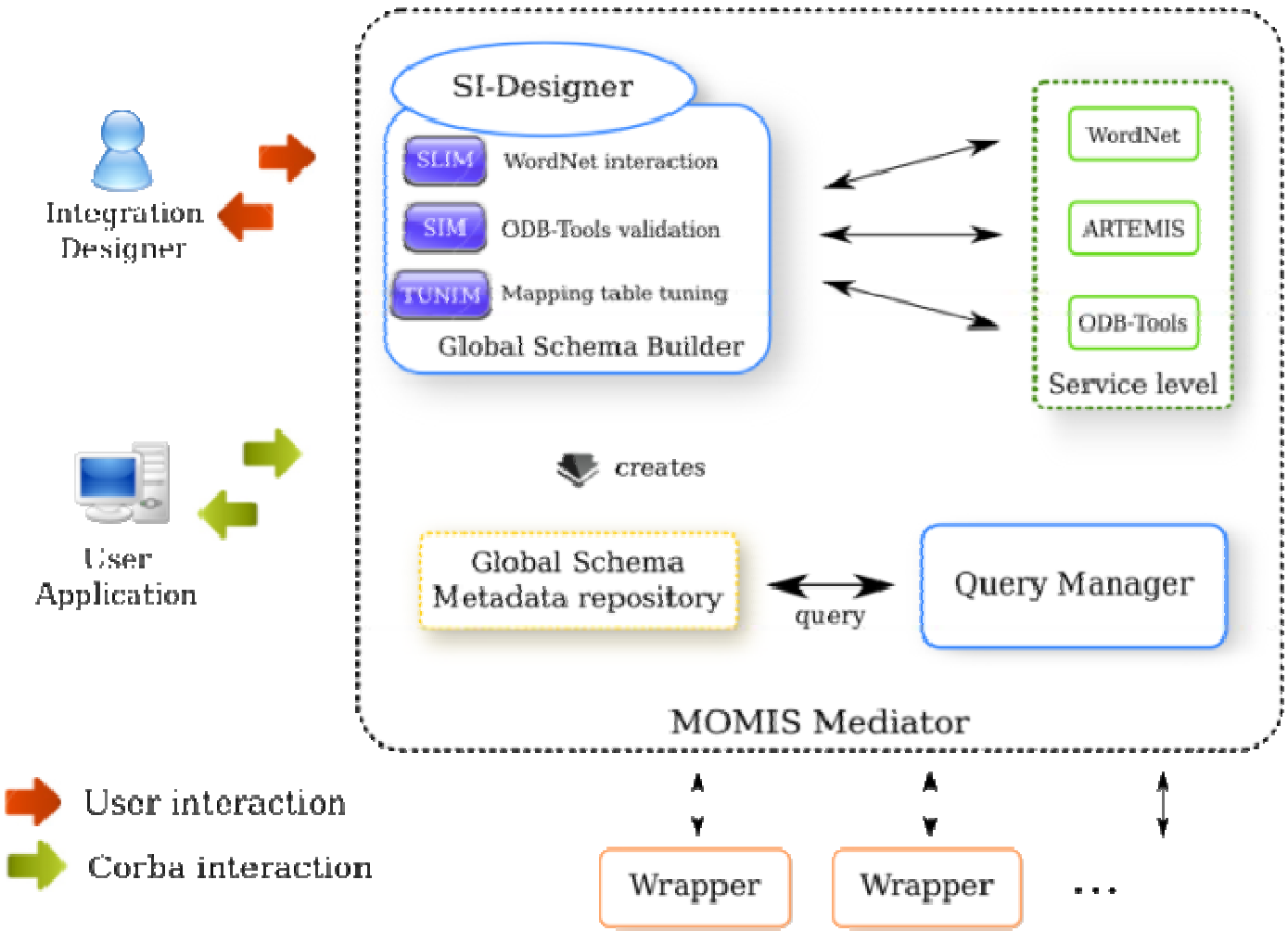
Microsoft SQL Server 2005 Integration Services (SSIS) is a scalable platform for creating data integration solutions at high performance.

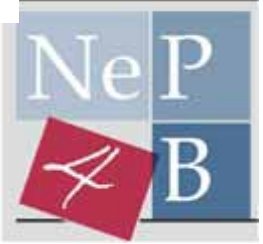
- *Integration Services:* deal with the monitoring of packages Integration Services execution and manages the storage of packages
- *The object model of Integration Services:* includes API for accessing to the tools, to the command-line utilities and to the customizable applications of Integration Service
- *Data Flow*: manages the transformations that modify the data and the destinations that load or make them available for other processes
- *The run-time:* saves the layout of the packages, runs the packages and provides support for registration, configuration, connections and transactions

MOMIS

- The MOMIS (Mediator envirOnment for Multiple Information Sources) is a framework to perform information extraction and integration from both structured and semi-structured data sources developed by the DBGroup atUNIMORE
  - An object-oriented language, with an underlying Description Logic, called ODL-I3, derived from the standard ODMG is introduced for describing schemata
  - Information integration is performed in a semi-automatic way, by exploiting the knowledge in a Common Thesaurus and the ODL-I3 descriptions with a combination of clustering and Description Logics techniques
  - This integration process gives rise to a virtual Global Schema of the underlying sources
- Tool-supported techniques to construct the Global Schema (GS)
- Local Schema Annotation w.r.t. a common lexical thesaurus (WordNet)
  - Semi-automatic generation of relationships among local schemata
  - Clustering techniques (to build the GS)
  - Semi-automatic generation of GAV mappings between the GS and the local schemata (Mapping Table)
  - Mapping Table refinement interface
  - GAV query manager
- An open source version under development by the startup DATARIVER at UNIMORE

MOMIS - architecture





Commercial Tools

Comparative analysis

Comparative Analysis

- The systems may be divided in two categories:

Virtual approach:

  - MOMIS → it supports a semi-automatic process for building the Global Schema
  - IICE → the designer creates the Global Schema of the attributes, the mapping process is manual (supported by a graphical interface)

- Data Warehouse approach: ETL and E-LT integration logic, we have a real DB
  - Microsoft SSIS and Oracle DI
    - the mappings between the local souces and the Global Schema are manually created by means of a GUI

Comparative Analysis

	Producer	Type of data	Approach	View	Query manager
MOMIS	DBGROUP-UNIMO**	Semi-structured and structured	Virtual Database (GAV)	Semiautomatic	Yes
IICE 8.4	IBM	Structured, semi-structured, multimedial data	Virtual Database	Manual (GUI)	NO*
Data Integrator	Oracle	Structured and semi-structured	Real DB (E - LT)	Manual (GUI)	Yes
Integration Services	Microsoft	Structured and semi-structured	Real DB (ETL)	Manual (GUI)	Yes

* allows you to query the virtual database by a command-line interface

** DATARIVER

Comparative Analysis by means of Thalia

www.cise.ufl.edu/research/dbintegrate/thalia

THALIA *(Test Harness for the Assessment of Legacy Information Integration Approaches)* [Hammer 2005] is a public tool used as benchmark to test integration systems

- It provides different data sources for testing the systems
- It has a set of 12 queries fo testing data integration systems. Each query evaluates as the system reacts to a particular syntactic and semantic heterogeneity
- It provides a function to obtain a final evaluation of the systems
- It is possible to divide the 12 Thalia queries into 3 main groups:
  1. Heterogeneity of attributes
  2. Missing data
  3. Structural heterogeneity

Heterogeneity of attributes

- Heterogeneity may exist between two semantically related attributes which belong to different sources. The following queries are provided:
  - ***Synonyms (query 1):*** attributes which have a different name but the same meaning (i.e. “instructor” or “lecturer”)
  - ***Simple mappings (query 2):*** related attributes which are in different schemas and differ one to each other due to mathematical transformations of their values. (e.g. time can be expressed in 0-24h or 0-12h)
  - ***Union types (query 3):*** *related* attributes in different schemas use different kinds of data types to represent the same information
  - ***Complex mappings (query 4):*** related attributes differ by a complex transformation of their values
  - ***Language Expressions (query 5):*** attribute names or values having the same meaning but expressed in different laguages (i.e. “Database” in English but “Datenbank” in German)

Missing data

- These are heterogeneities that are due to missing information (structure or value) in one of the schemas
  - ***Null values (query 6):*** the attribute (value) does not exist
  - ***Virtual attributes (query 7):*** information that is explicitly provided in one schema is only implicitly available in the other
  - ***Semantic incompatibility (query 8):*** a concept modelled by an attribute in a schema does not exist in the other schema

Structural heterogeneity

- Heterogeneities due to discrepancies in the way related information is modeled/represented in different schemas
  - ***The same attribute in different structure (query 9):*** the same attribute may be in different locations in different schemas
  - ***Handling sets (query 10):*** a set of values is represented using a single, set-valued attribute in one schema vs. a collection of single-valued attributes organized in a hierarchy in another schema
  - ***Attribute name does not define semantics (query 11):*** the name of the attribute does not adequately describe the meaning of the value that is stored there
  - ***Attributes composition (query 12):*** the same information can be represented by a single attribute or by a set of attributes organized in an hierarchical way

Results of the benchmark THALIA

We compared the integration systems on the basis of the criteria established in the THALIA benchmark

	MOMIS	IBM IICE	Oracle DI	Microsoft IS
	Heterogeneity of attributes			
Query 1	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆
Query 2	Yes ☆☆	Yes ☆	Yes ☆☆	Yes ☆☆☆
Query 3	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆
Query 4	Yes ☆☆☆	Yes ☆	Yes ☆☆☆	Yes ☆☆☆
Query 5	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆
	Lack of data			
Query 6	Yes ☆☆☆	Yes ☆☆	Yes ☆☆☆	Yes ☆☆☆
Query 7	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆	Yes ☆☆☆
Query 8	Yes ☆☆☆	Yes ☆☆	Yes ☆☆☆	Yes ☆☆☆
	Structural heterogeneity			
Query 9	Yes ☆☆☆	No*	Yes ☆☆☆	Yes ☆☆☆
Query 10	Yes ☆☆☆	No*	Yes ☆☆☆	Yes ☆☆☆
Query 11	Yes ☆☆☆	Yes ☆	Yes ☆☆☆	Yes ☆☆☆
Query 12	Yes ☆☆	Yes ☆	Yes ☆☆	Yes ☆☆

**These query can be solved by making views of the data sources*

The number of stars shows the complexity of the implementation. One star: difficult operation. Three stars simple task

Results of the benchmark THALIA (2)

- It is not possible to implement the queries 9 and 10 in IICE because it is not possible to define join relations among different data structures
  - The query may be indirectly solved by means of integration of views
- The solution of a query in IICE implies the implementation of a Java class
- The Microsoft and Oracle systems allow the user to use several functions for solving the problem on hand.
- The MOMIS system allows the user to formulate SQL 92 functions for managing data [Hammer et al., ICDE 2005]