Data Schema Integration

Mustafa Jarrar

Birzeit University

Watch this lecture and download the slides

Online Courses : http://www.jarrar.info/courses

Thanks to Anton Deik for helping me preparing this lecture

Data Schema Integration

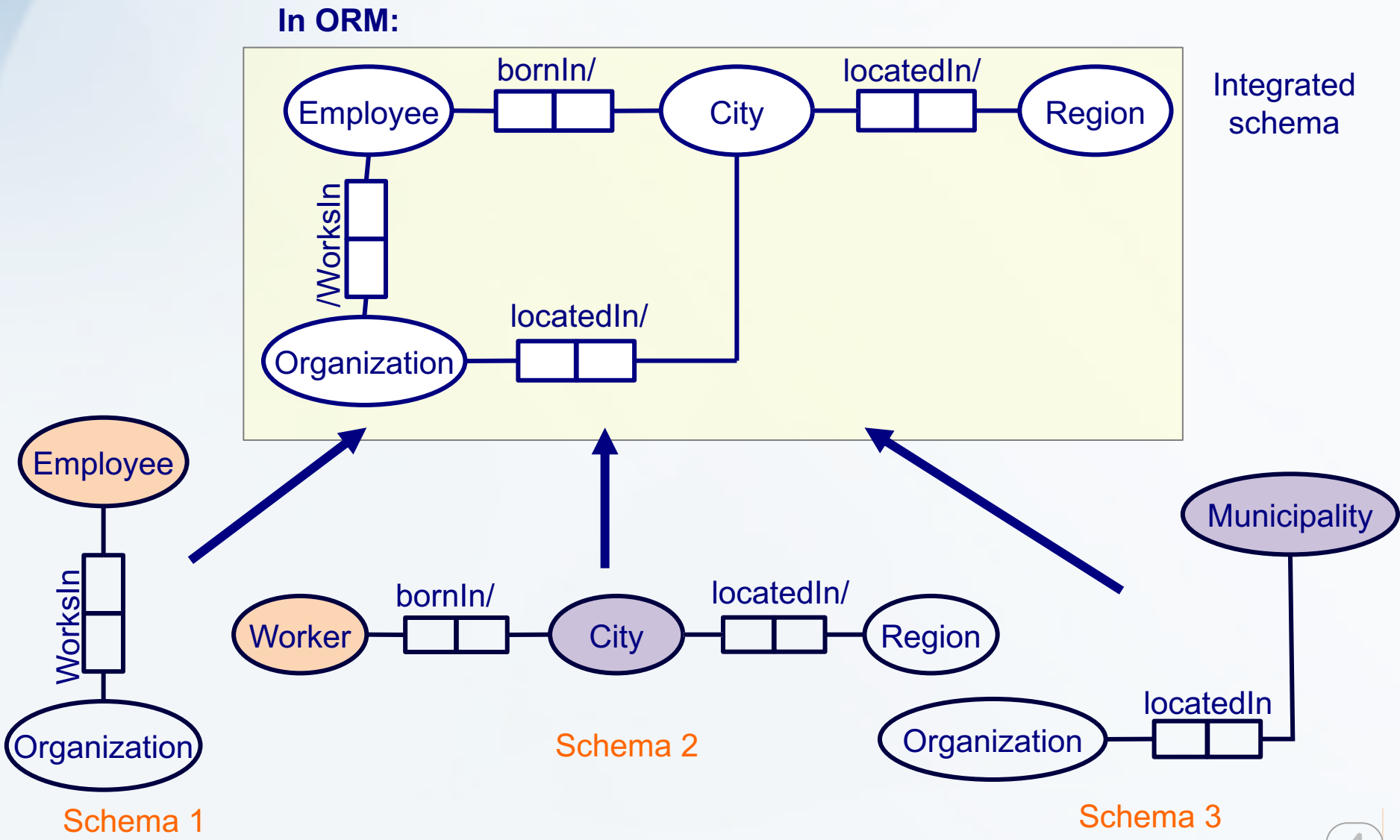
Part 1: Examples of Schema Integration Challenges

Part 2: Framework for Schema Integration
- Step 1- Schema Transformation
- Step 3- Schema Matching
- Step 3- Schema Integration

Part 3: Integration Process and Rules

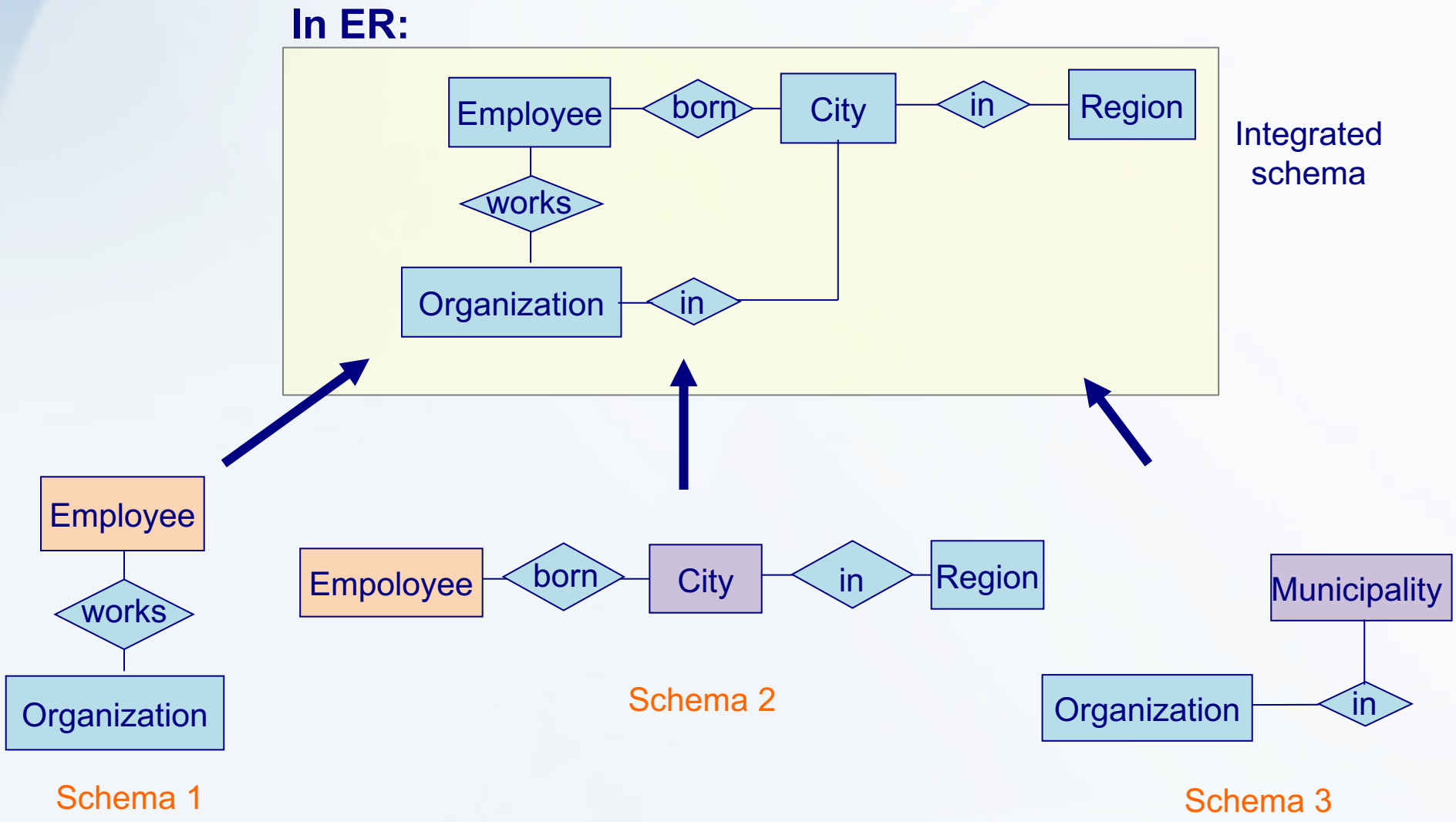
Keywords: Data Schema Integration, Integrated schema, heterogeneities, conflicts ,Transformation Rules, Matching Rules, Integration Rules, Schemas Transformation, Schemas Matching, Mappings, integration strategy, Framework, data model, homogeneization , Reverse Engineering, GAV and LAV Integration,Global As View ,Local As View

Data Schema Integration: A simple example



Data Schema Integration: A simple example

Source: Carlo Batini



Challenges of Data Schema Integration

Based on Carlo Batini [13]

Schema Integration has two major challenges:

1. Identification of all portions of schemas that are related to the same concept, in such a way to unify such different representations in the global schema.

2. Identification, analysis and resolution of the different types of conflicts (heterogeneities) in different schemas.

Data Schema Integration

Part 1: Examples of Schema Integration Challenges

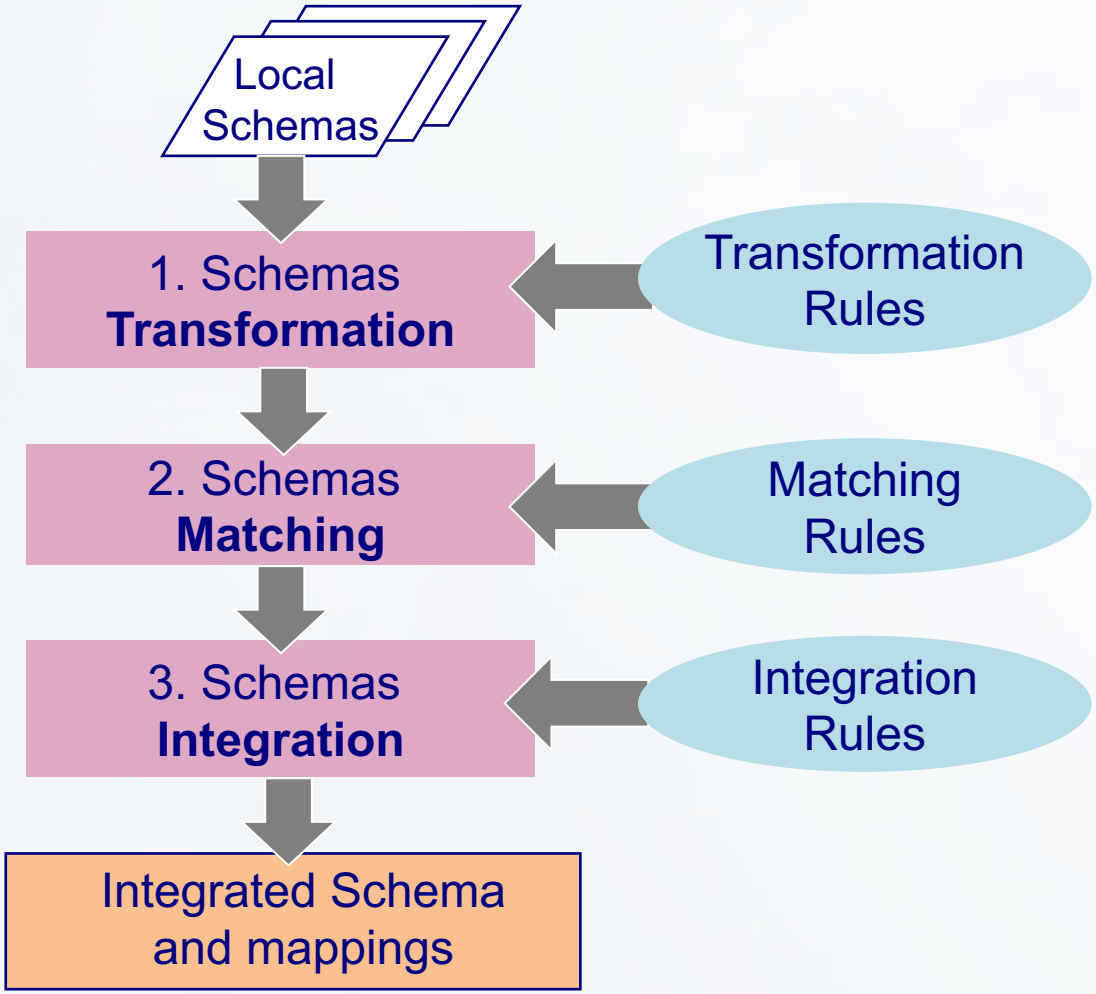
Part 2: Framework for Schema Integration

- Step 1- Schema Transformation
- Step 3- Schema Matching
- Step 3- Schema Integration

Part 3: Integration Process and Rules

Keywords: Data Schema Integration, Integrated schema, heterogeneities, conflicts ,Transformation Rules, Matching Rules, Integration Rules, Schemas Transformation, Schemas Matching, Mappings, integration strategy, Framework, data model, homogeneization , Reverse Engineering, GAV and LAV Integration,Global As View ,Local As View

Framework for Schema Integration



Source: Advances in Object-Oriented Data Modeling, M. P. Papazoglou, S. Spaccapietra, Z. Tari (Eds.), The MIT Press, 2000

Framework for Schema Integration

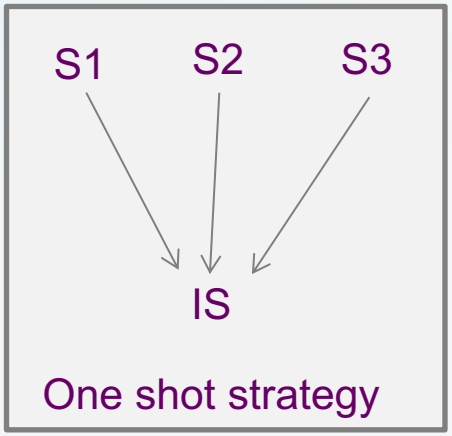
Define the integration strategy

If the number of local schemas to be integrated is large, the order of schema integration becomes important. There are several strategies.

Input: *n* source schemas

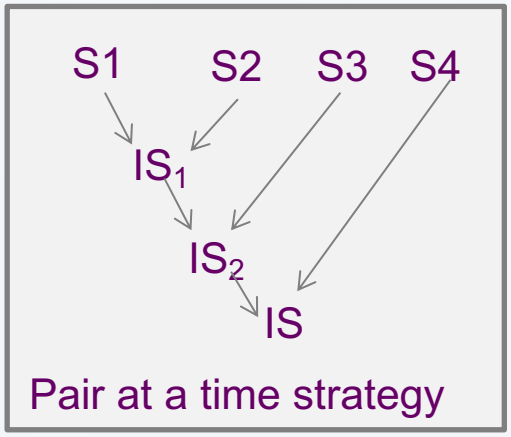
Output: *n* source schemas + integration strategies

Method used: heuristics



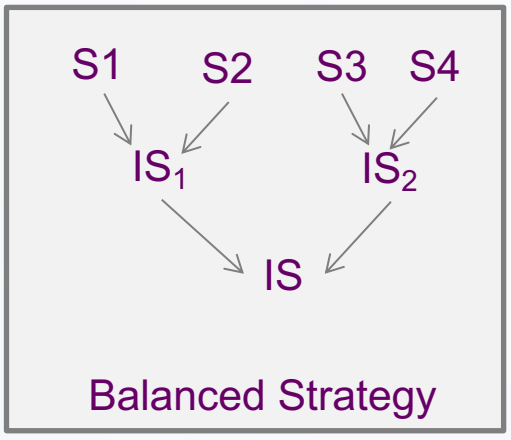
One shot strategy

- Efficient integration process
- Many correspondences between concepts have to be considered together.



Pair at a time strategy

- Priority to most relevant and stable schemas.
- The integration process is more efficient.



Balanced Strategy

- e.g.: Production, Marketing, Sales.
- Preferred when the cohesion among schemas is high.

...

Framework for Schema Integration

Source: Stefano Spaccapietra

Schema transformation (or Pre-integration)

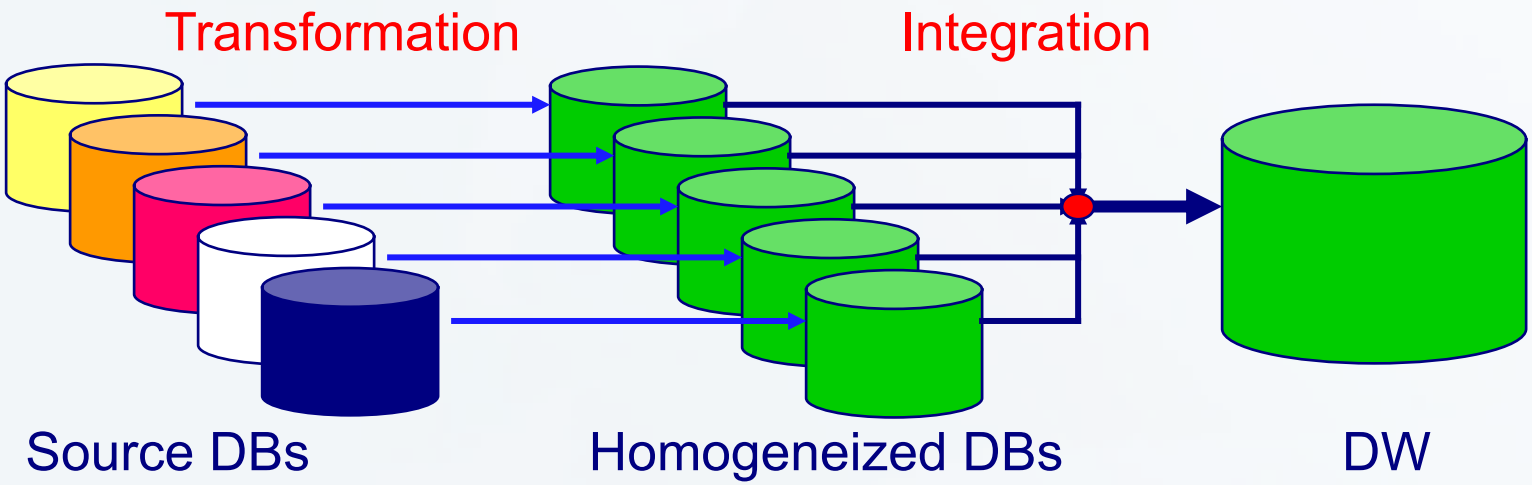
Input: *n* source schemas

Output: *n* source schemas homogeneized

Methods used: Model and Design Homogeneization

Reduce model heterogeneities as much as possible to make the sources more suitable for integration.

Goal: use a single, common data model and format.



Step 1: Schema Transformation

Schema Transformation involves:

- Data model homogenization
  - Describe all data sources using the same data model.
- Design homogeneization
  - Enforce standard design rules to reduce structural conflicts (e.g., normalization: one fact in one place).
- Reverse Engineering
  - Reverse engineer the schema from existing data (such as COBOL files, spreadsheets, legacy relational databases, legacy object-oriented databases).

Example of Design homogeneization (Normalization)

One Table:

R1 (#Student, Name, LastName, #Course, CourseName, Grade, Date)

Dependencies:

- #Student → Name, LastName
- #Course → CourseName
- #Student #Course → Grade, Date)

Normalized Into 3 Tables: one fact in one place:

R11 (#Student, Name, LastName)

R12 (#Course, CourseName)

R13 (#Student, #Course, Grade, Date)

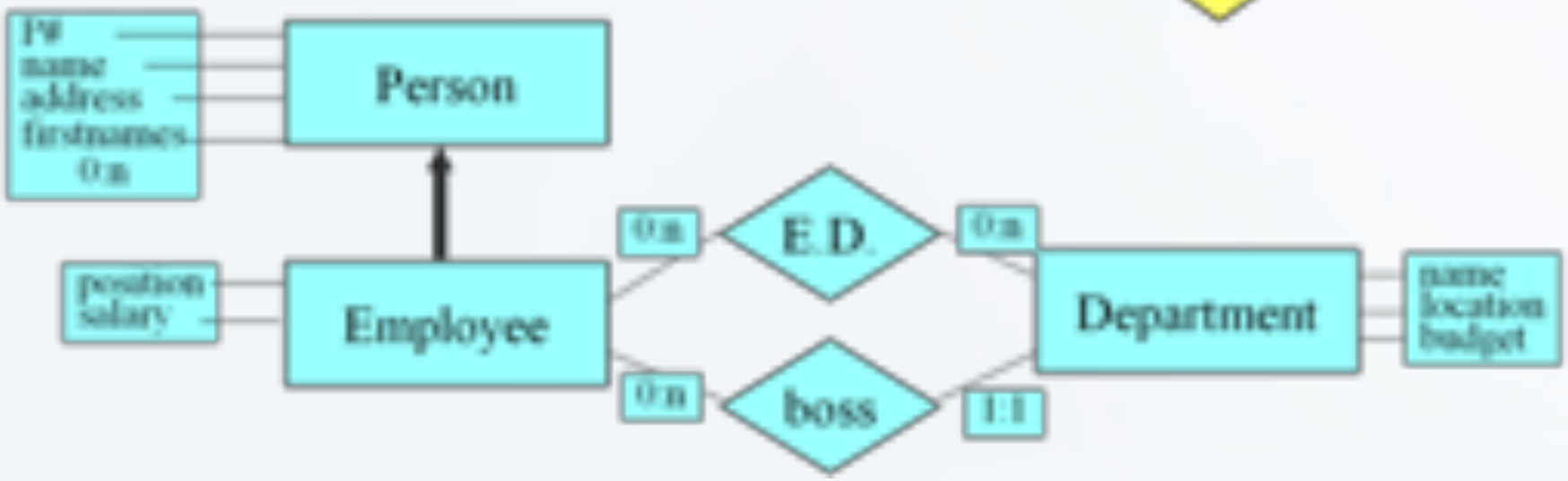
Example of Reverse Engineering

Source: Stefano Spaccapietra

Reconstructing a physical model from an existing schema

Person (P#, name, address)
Firstnames (P#, firstname)
Employee (E#, position, salary)
EmpDept (E#, department)
Department (name, location, boss, budget)

?



Step 2: Schema Matching

Schema matching (Correspondences investigation)

Input: *n* source schemas
Output: *n* source schemas + correspondences
Method used: techniques to discover correspondences

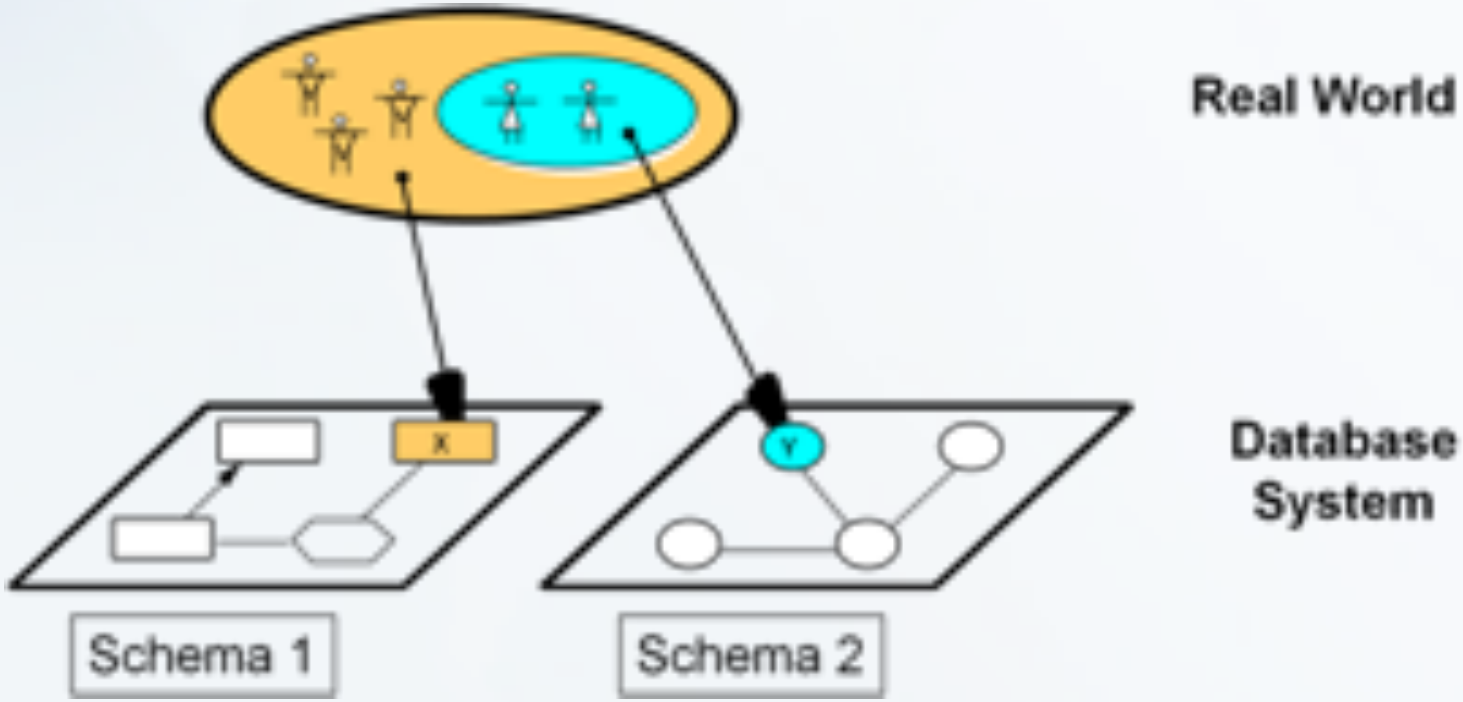
Correspondences relate (schema) elements which describe the same phenomena of the real world.

- This step aims at finding and describing all semantic links between elements of the input schemas and the corresponding data.
- By doing so, one matches between the schemas to be integrated.
- This step fixes the conflicts found in the schema.

Semantics of Correspondences

Source: Stefano Spaccapietra

Correspondences relate (schema) elements which describe the same phenomena of the real world.

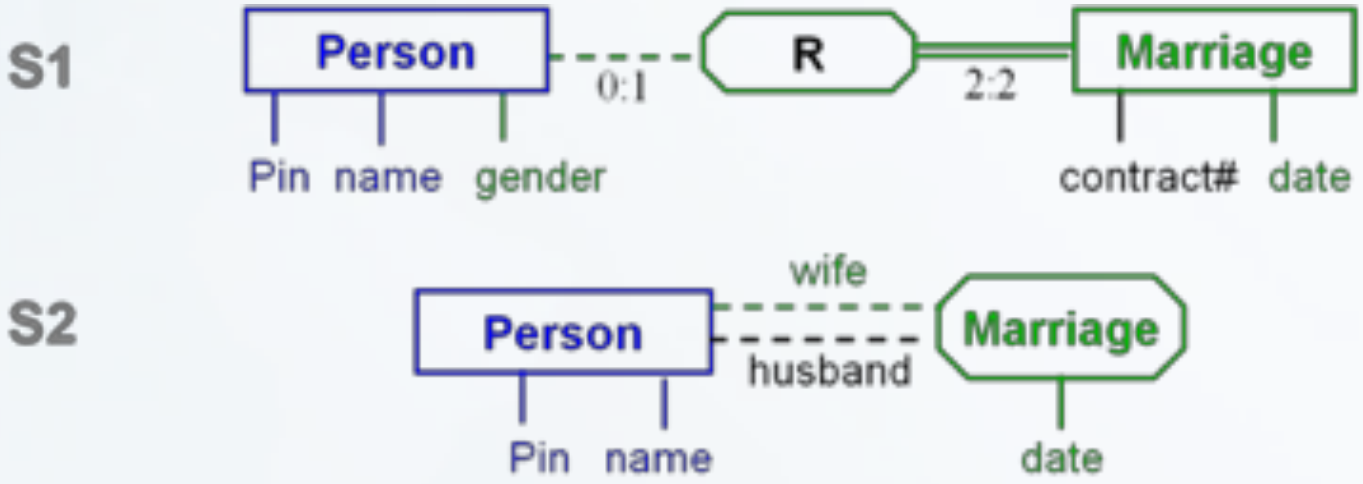


Asserting Correspondences

Source: Stefano Spaccapietra

Finding **matching correspondences** is done through the use of a rich language for expressing correspondences (**matchings**).

Example:




```
S1.Person ≡ S2.Person,
With Corresponding Identifiers: Pin,
With Corresponding Property: name
```

Automated Matching

- Fully automated matching is impossible, as a computer process can hardly make ultimate decisions about the semantics of data.
- But even partial assistance in discovering of correspondences (to be confirmed or guided by humans) is beneficial, due to the complexity of the task.
- All proposed methods rely on some similarity measures that try to evaluate the semantic distance between two descriptions.

Some state of the art matching systems

Cupid (Microsoft Research, USA)

FOAM/QOM (University of Karlsruhe, Germany)

OLA (INRIA Rhône-Alpes, France / University of Montreal, Canada)

S-Match (University of Trento, Italy)

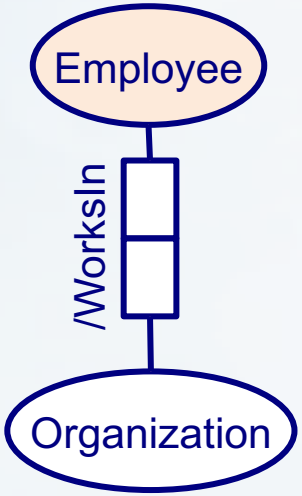
… many others

Examples of Correspondences

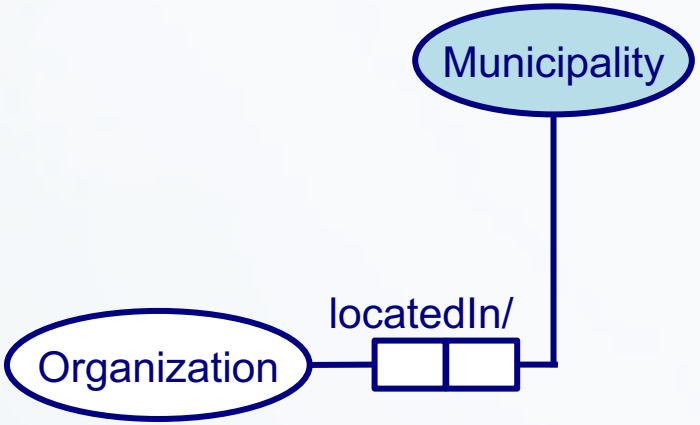
Source: Stefano Spaccapietra

Examples of Correspondences

Previous example



Schema 1



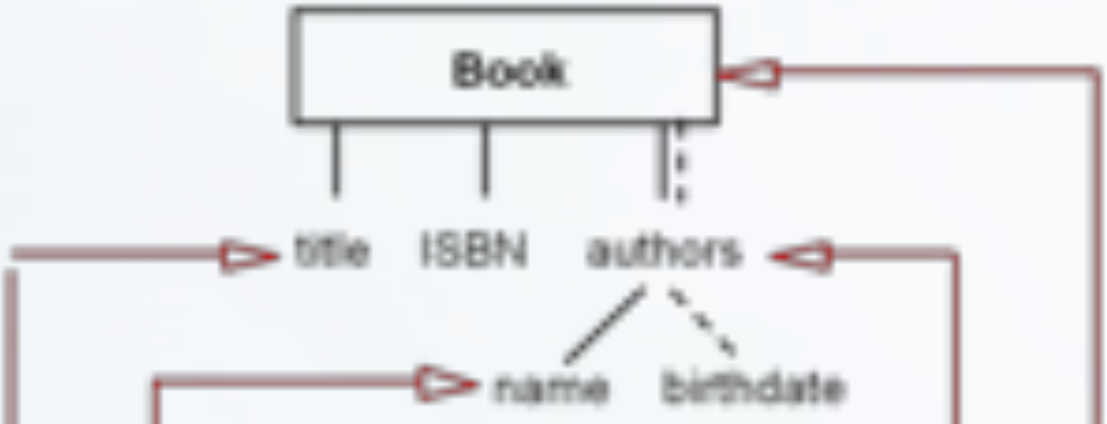
Schema 3

Schema 2

Examples of Correspondences

Source: Stefano Spaccapietra

- Schema 1



- Schema 2

Step 3: Schema Integration & Mapping Generation

Source: Carlo Batini

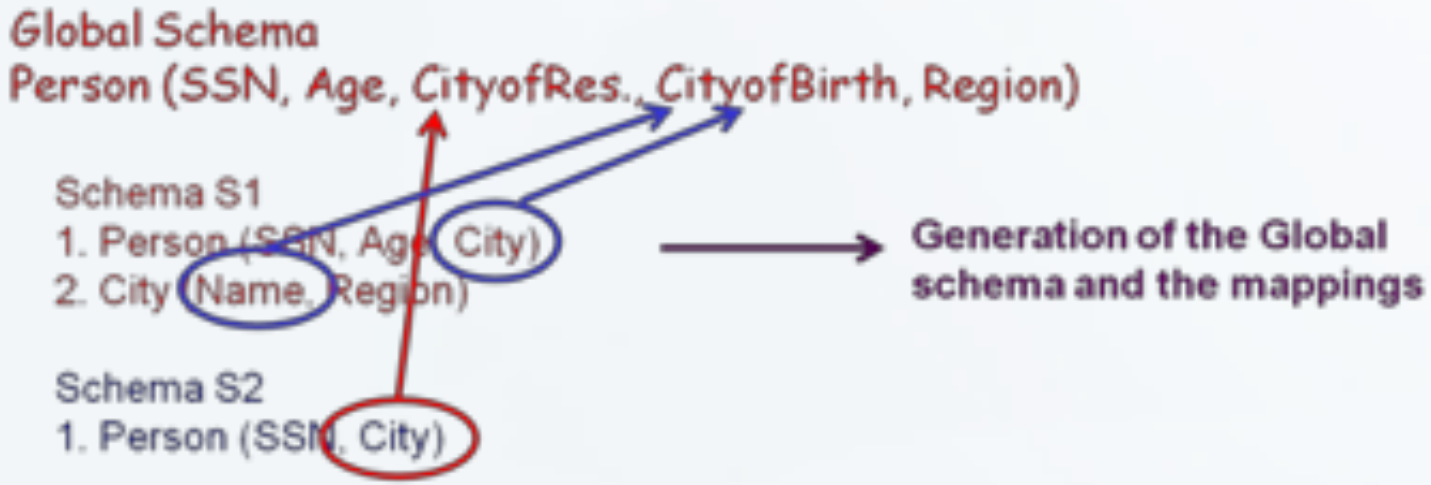
Schema integration and mapping generation

Input: *n* source schemas + correspondences

Output: integrated schema + mapping rules btw the integrated schema and input source schemas

Method used: New classification of conflicts + Conflict resolution transformations

GOAL: creating an Integrated Schema (IS) and the mappings to the local databases.



GAV and LAV Integration

Research has identified two methods to set up mappings between the integrated schema and the input schemas:

(1) **GAV (Global As View): proposes to define the integrated schema as a view over input schemas.**

GAV is usually considered simpler and more efficient for processing queries on the integrated database, but is weaker in supporting evolution of the global system through addition of new sources.

(2) **LAV (Local As View): proposes to define the local schemas as views over the integrated schema.**

LAV generates issues of incomplete information, which adds complexity in handling global queries, but it better supports dynamic addition and removal of source.

Data Schema Integration

Part 1: Examples of Schema Integration Challenges

Part 2: Framework for Schema Integration
- Step 1- Schema Transformation
- Step 3- Schema Matching
- Step 3- Schema Integration

Part 3: Integration Process and Rules

Keywords: Data Schema Integration, Integrated schema, heterogeneities, conflicts ,Transformation Rules, Matching Rules, Integration Rules, Schemas Transformation, Schemas Matching, Mappings, integration strategy, Framework, data model, homogeneization , Reverse Engineering, GAV and LAV Integration,Global As View ,Local As View

Integration Process

After we identified the correspondences (in the previous step), we now solve the conflicts:

One can distinguish between four types of conflicts:

- Structural conflicts
- Classification conflicts
- Descriptive conflicts
- Fragmentation conflicts

Examples of conflicts among related object types

- different classifications (sets of instances)
- different sets of properties
- different structures
- different coding schemes
- …

Integration Rules

Rules defining the strategy to solve conflicts

Example rules:

- If a class corresponds to an attribute, keep the class
- If the population of a class is included in the population of another class, build an is-a hierarchy

Integration rules depend on how you want the integrated schema to look like

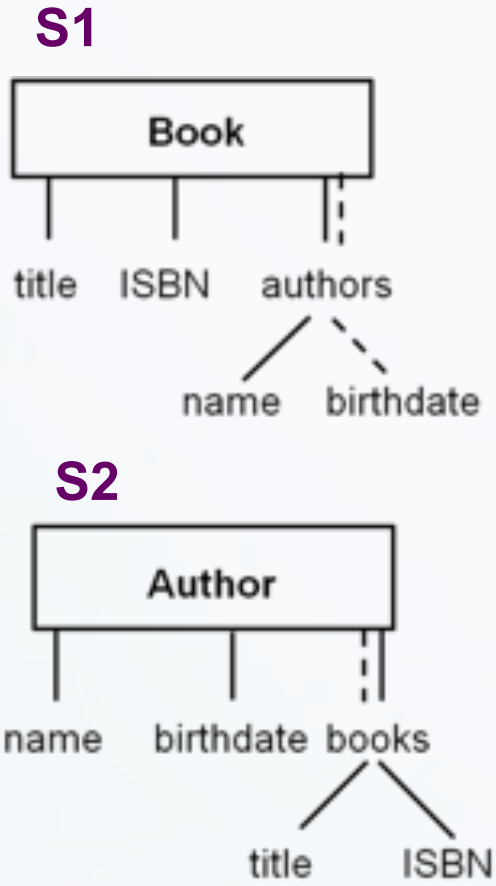
Structural Conflicts

Source: Stefano Spaccapietra

Different schema element types, e.g.: class, attribute, relationship

Library example:

- S1 : Book is a class
- S2 : books is an attribute of Author



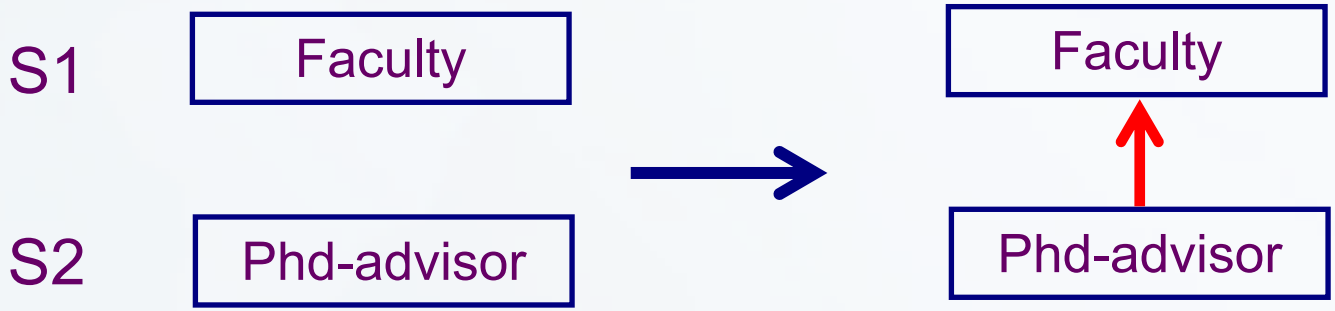
Conflict resolution :

Choose the less constraining structure

- Integrated Schema: Book is a class

Classification Conflicts

- Corresponding elements describe different sets of real world objects
  - S1.Faculty CONTAINS S2.PhD-advisor
- Conflict Resolution:
  - Generalization / Specialization hierarchy



  - Merging

Descriptive Conflicts

Corresponding types have different properties, or corresponding properties are described in different ways

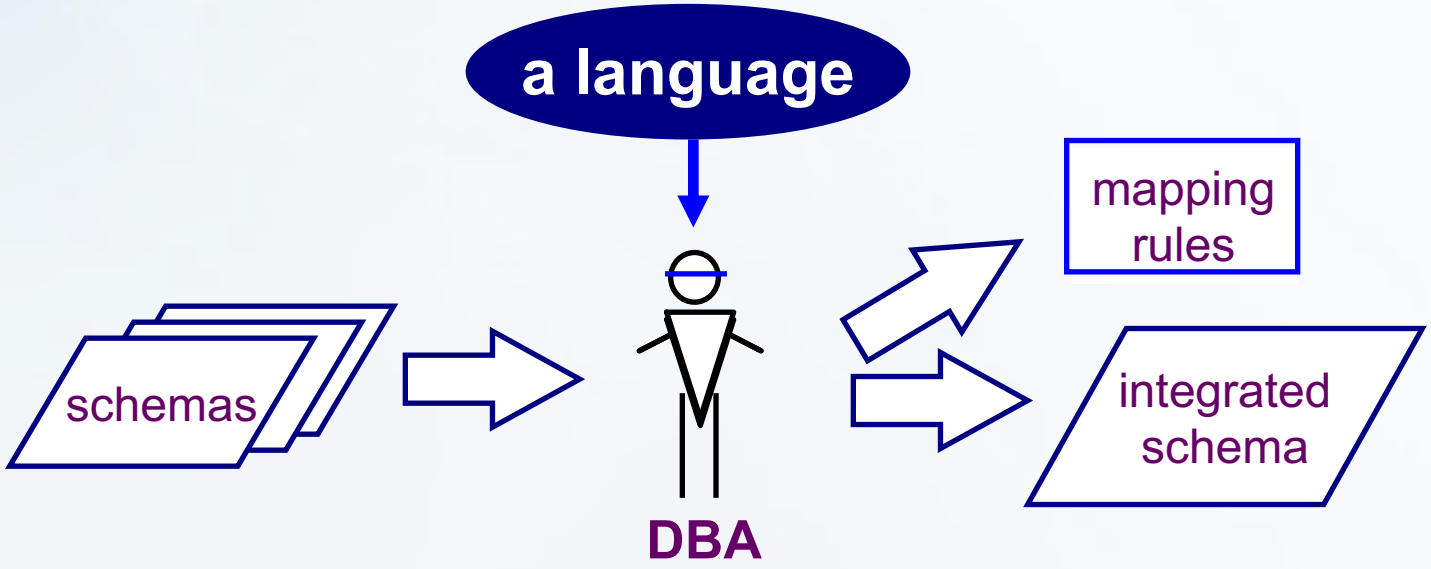
Object / Entity / Relationship type:

- Naming conflicts:
  - synonyms Worker , Employee
  - homonyms Highway (EU) , Highway (USA)
- Composition conflicts: different attributes and methods
  - Employee (E# , name , address)
  - Employee (E# , position , salary , department)

Integration Methods: Manual

Source: Stefano Spaccapietra

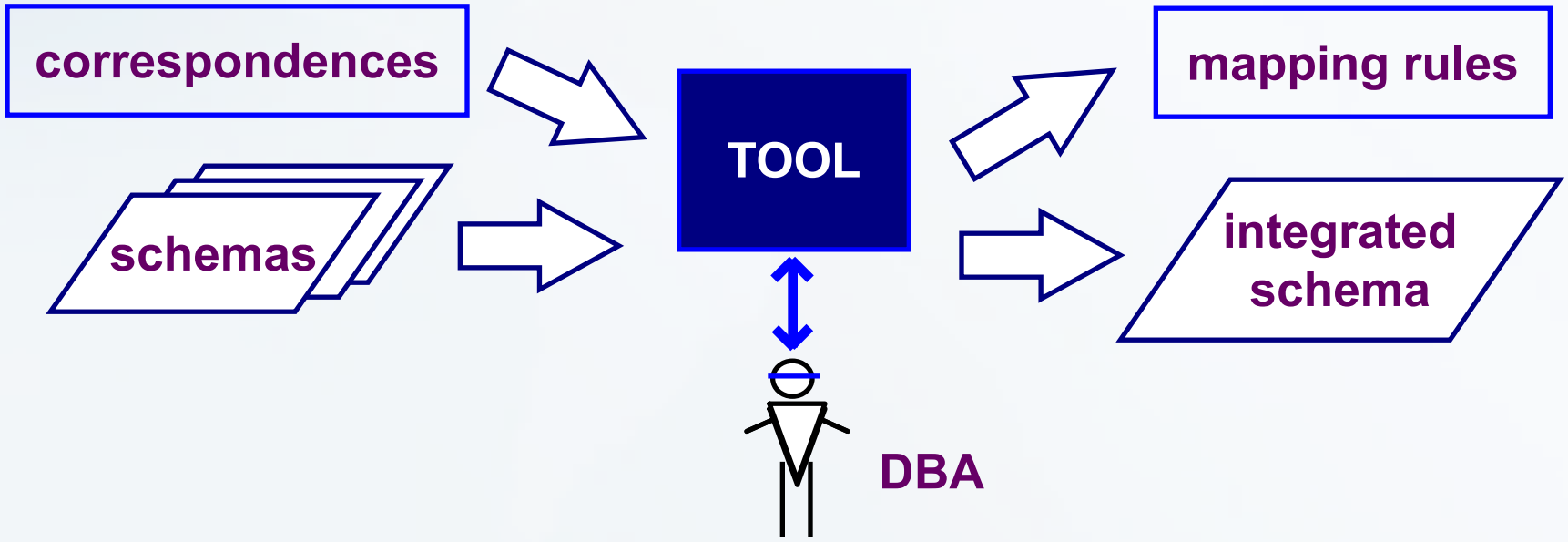
First method: manual integration " do it yourself "



Easy to implement , Flexible
BUT
time consuming for the DBA

Integration Methods: Semi-Automatic

Second method: semi-automatic integration
“ tell me about the problem, I will try to fix it “



Opens to visual CASE tools, integration servers
BUT knowledge acquisition can be painful

References

[1] Mustafa Jarrar, Anton Deik: The Graph Signature: A Scalable Query Optimization Index for RDF Graph Databases Using Bisimulation and Trace Equivalence Summarization. International Journal on Semantic Web and Information Systems, 11(2), 36-65,. April-June 2015

[2] Mustafa Jarrar, Anton Deik, Bilal Faraj: Ontology-Based Data And Process Governance Framework -The Case Of E-Government Interoperability In Palestine . In pre-proceedings of the IFIP International Symposium on Data-Driven Process Discovery and Analysis (SIMPDA'11). Pages(83-98). 2011.

[3] Mustafa Jarrar and Marios D. Dikaiakos: A Query Formulation Language for the Data Web. The IEEE Transactions on Knowledge and Data Engineering. IEEE Computer Society. Pages(783-798). Volume 24, Number 4, April 2012

[4] Paolo Ceravolo, Chengfei Liu, Mustafa Jarrar, Kai-Uwe Sattler: Special Issue on Querying the Data Web -Novel techniques for querying structured data on the web. The World Wide Web Journal. Volume(14), Issue (5-6). Springer. August 2011. ISSN:1573-1413.

[5] Anton Deik, Bilal Faraj, Ala Hawash, Mustafa Jarrar: Towards Query Optimization for the Data Web - Two Disk-Based algorithms: Trace Equivalence and Bisimilarity. Proceedings of the 3rd Palestinian International Conference on Computer and Information Technology (PICCIT 2010). 2010.

[6] Mustafa Jarrar, Marios D. Dikaiakos: Querying the Data Web: the MashQL Approach. IEEE Internet Computing. Volume 14, No. 3. Pages (58-670). IEEE Computer Society, ISSN 1089-7801. May 2010.

[7] Mustafa Jarrar, Marios D. Dikaiakos: Querying the Data Web: the MashQL Approach. IEEE Internet Computing. Volume 14, No. 3. Pages (58-670). IEEE Computer Society, ISSN 1089-7801. May 2010.Mustafa Jarrar and Marios D. Dikaiakos: A Data Mashup Language for the Data Web . Proceedings of LDOW, WWW'09. ACM. ISSN 1613-0073. (2009).

[8] Mustafa Jarrar and Marios D. Dikaiakos: MashQL: a query-by-diagram topping SPARQL -Towards Semantic Data Mashups. Proceedings of ONISW'08, part of the ACM CiKM conference. ACM. pages (89-96) ISBN 9781605582559.(2008).

[0] Mustafa Jarrar: Towards methodological principles for ontology engineering. PhD Thesis. Vrije Universiteit Brussel. (May 2005)

[10] Mustafa Jarrar, Luk Vervenne, Diana Maynard: HR-Semantics Roadmap- The Semantic challenges and opportunities in the Human Resources domain . Technical Report. The Ontology Outreach Advisory, Belgium. (OOA-HR/2007-08-20/v025). August 2007

[11] Lyndon Nixon, Malgorzata Mochol, Mustafa Jarrar, Stamatia Dasiopoulou, Vasileios Papastathis, and Yiannis Kompatsiaris: Prototypical business use cases. Deliverable D1.1.2 (WP1.1), The Knowledge Web Network of Excellence (NoE) IST-2004-507482, Luxemburg. January 2005.

[12] Peter Spyns, Daniel Oberle, Raphael Volz, Jijuan Zheng, Mustafa Jarrar, York Sure, Rudi Studer, and Robert Meersman: OntoWeb- a Semantic Web Community Portal. Proceedings of the 4th International Conference on Practical Aspects of Knowledge Management (PAKM 2002). Pages (189-200). LNCS 2569, Springer. ISBN: 3540003142. December 2002.

[13] Carlo Batini: Course on Data Integration. BZU IT Summer School 2011.

[14] Stefano Spaccapietra: Information Integration. Presentation at the IFIP Academy. Porto Alegre. 2005.

[15] Chris Bizer: The Emerging Web of Linked Data. Presentation at SRI International, Artificial Intelligence Center. Menlo Park, USA. 2009.