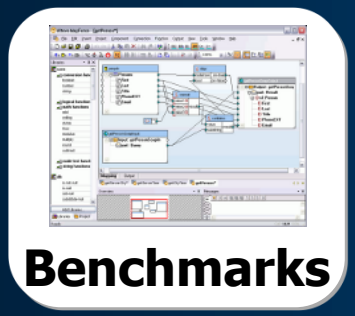
Schema Matching and Mapping Systems



From Usage To Evaluation

Angela Bonifati (Université Lyon 1 & CNRS Liris, BD Team)

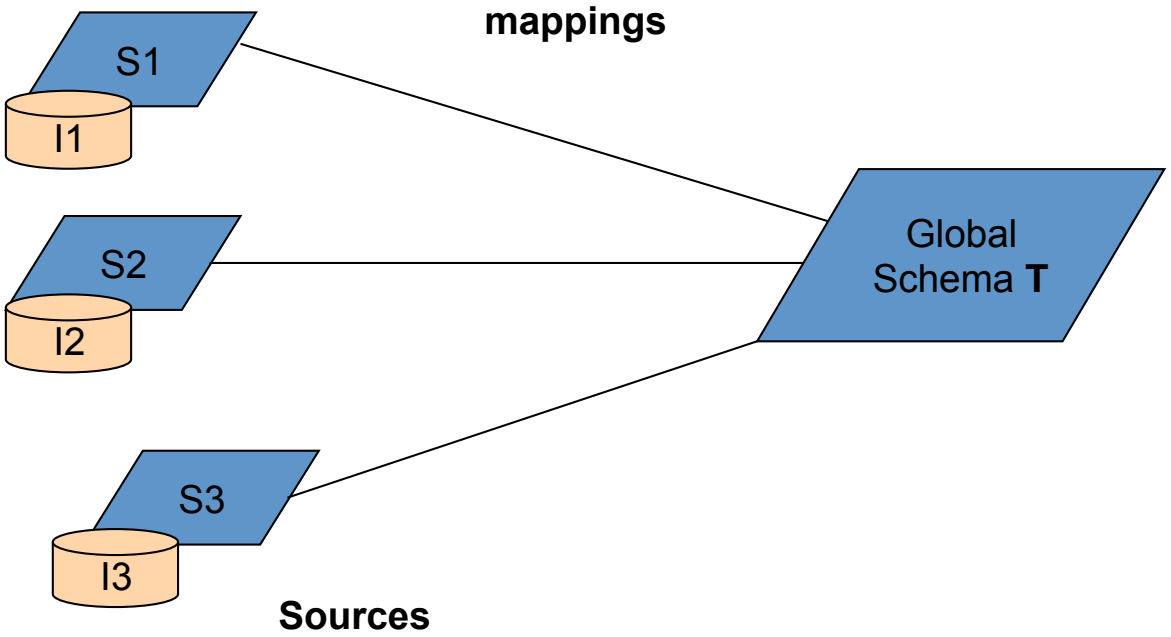
2ème Journée LSC du LIP, June 24th 2016

Introduction

- Data is inherently heterogeneous
  - Due to the explosion of online data repositories
  - Due to the variety of users, who develop a wealth of applications
    - At different time
    - With disparate requirements in their mind
- A fundamental requirement is to translate data across different formats
  - How data is transformed from one format to another is done through mappings

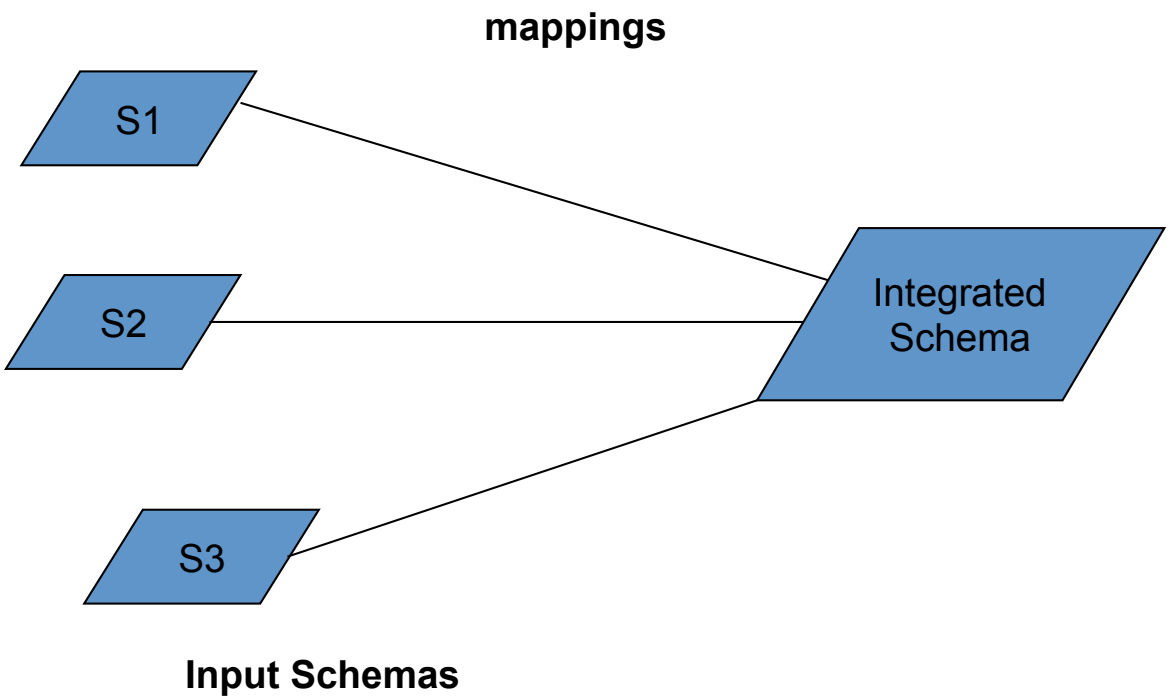
Mappings Are All Around

- Data integration [Lenzerini 2002]
  - to specify the relationship between local and global schemas



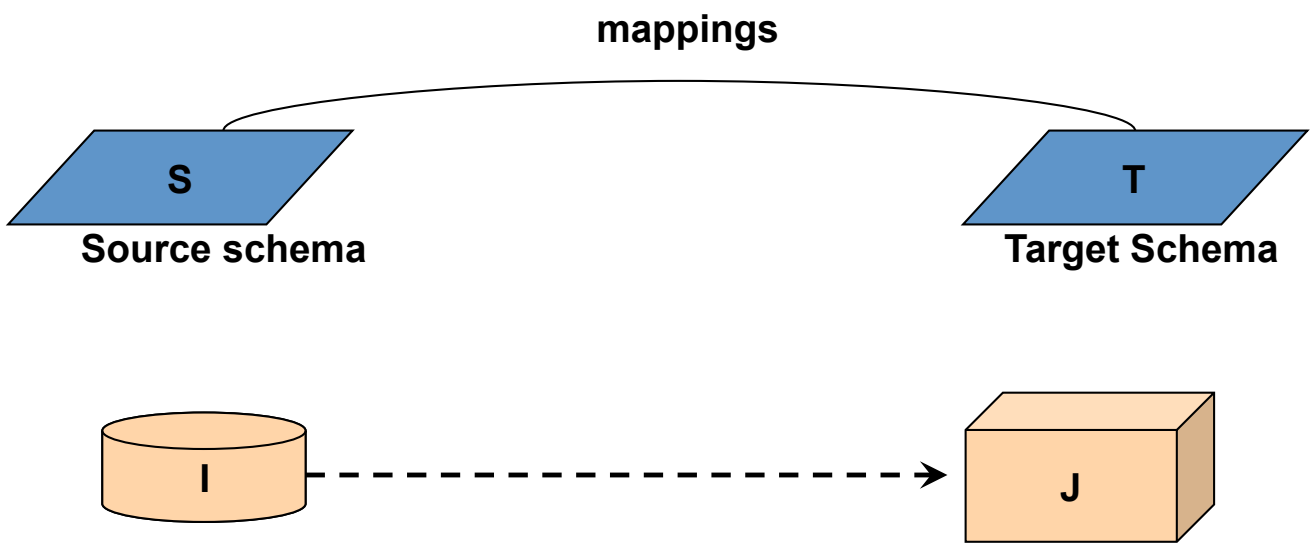
Mappings Are All Around

- Schema integration [Batini et al. 1986]
  - to specify the relationship between the input schemas and the integrated schemas



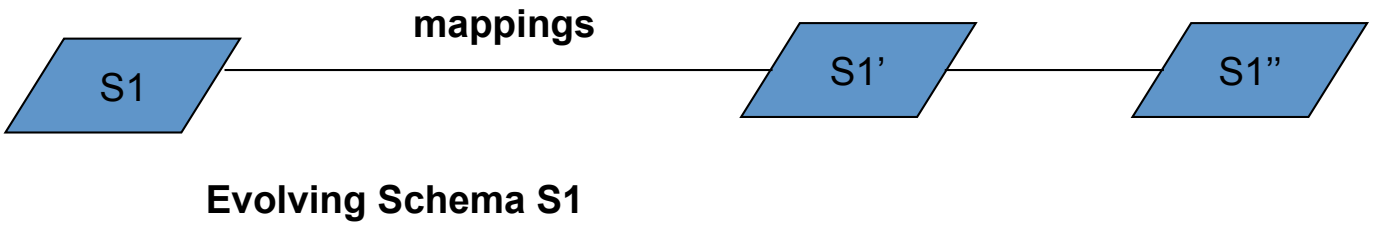
Mappings Are All Around

- Data exchange [Fagin et al. 2005]
  - to specify the relationship between source and target schemas



Mappings Are All Around

- Schema evolution [Lerner 2000]
  - to specify the relationship between the old and new version of an evolved schema



How Did It All Start

- One of the first systems to deal with this problem was developed at IBM in 1977:
  - EXPRESS (EXtraction, Processing and REStructuring System) [Shu et al. 1977] consists of two languages:
    - DEFINE that works as a DDL (Data Definition Language)
    - CONVERT that works as a DTL (Data Translation Language) and has a total of 9 operators, each of which receives as input a data file, performs the respective transformation and generates an output data file.
- EXPRESS required the users familiarity with the languages and was customized to only one model (hierarchical)
- After that, inter-model transformations were also studied [Tork-Roth et al. 1997] [Atzeni et al. 1997]

A Data Transfer Example

Source Instance

Projects

name	status
PIX	Active
E-services	Active
Clio	Inactive

Grants

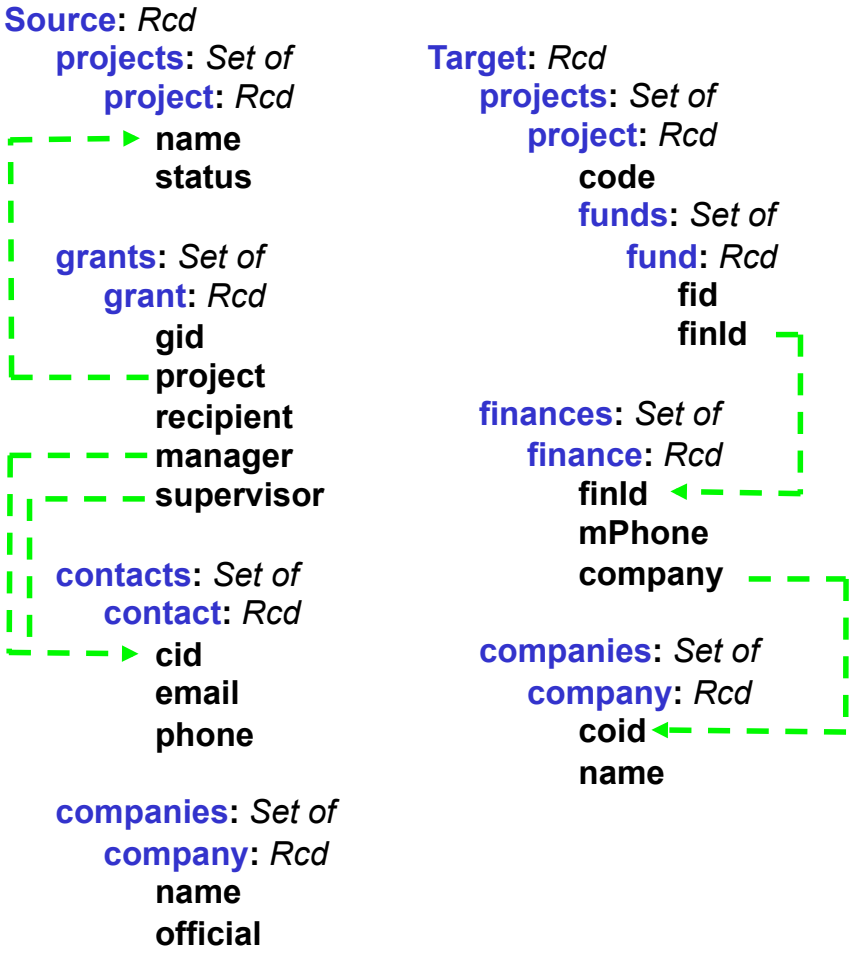
gid	project	recipient	manager	supervisor
g1	PIX	AT&T	Fernandez	Belanger
g2	PIX	AT&T	Shrivastava	Belanger
g3	E-services	Bell-labs	Benedikt	Hull

Contacts

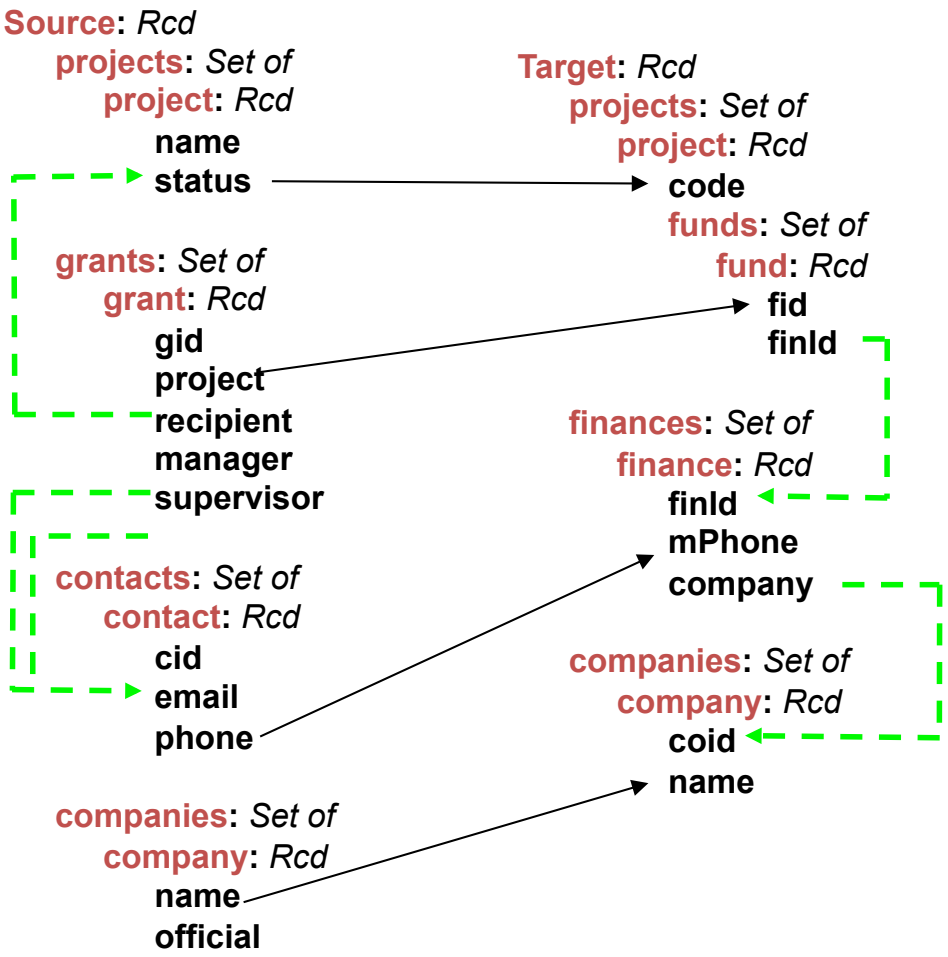
cid	email	phone
Benedikt	benedikt@research.bell-labs.com	5827766
Hull	hull@research.bell-labs.com	5824509
Shrivastava	divesh@research.att.com	3608776
Belanger	dgb@research.att.com	3608600
Fernandez	mff@research.att.com	3608679

Companies

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



Desired Target Instance



Target instance

Projects

code: E-services

Funds

fid	finId
g3	???

code: PIX

Funds

fid	finId
g1	???
g2	???

Finances

finId	mPhone	company
???	3608679	???
???	3608776	???
???	5827766	???

Companies

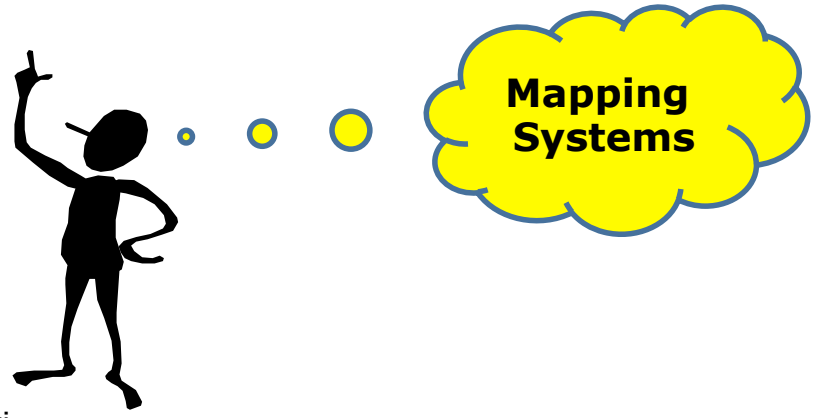
coid	name
Sk2(AT&T)	AT&T
Sk2(Lucent)	Lucent
???	???
???	???
???	???

The Needed Transformation Query

```
LET $doc0 := document("inputXMLfile")
RETURN
<T>  { distinct-values (
  FOR $x0 IN $doc0/S/grant, $x1 IN $doc0/S/project, $x2 IN $doc0/S/contact, $x3 IN $doc0/S/contact
  WHERE  $x2/cid/text() = $x0/manager/text() AND $x0/supervisor/text() = $x3/cid/text() AND $x0/project/text() = $x1/name/text()
  RETURN
    <project>
      <code>  {  $x0/project/text() }  </code>
      { distinct-values (
      FOR  $x0L1 IN $doc0/S/grant, $x1L1 IN $doc0/S/project, $x2L1 IN $doc0/S/contact, $x3L1 IN $doc0/S/contact
      WHERE $x2L1/cid/text() = $x0L1/manager/text() AND $x0L1/supervisor/text() = $x3L1/cid/text() AND
              $x0L1/project/text() = $x1L1/name/text() AND $x0/project/text() = $x0L1/project/text()
      RETURN
        <funding>
          <fid>  {  $x0L1/gid/text() }  </fid>
          <finId>  {  "Sk52(", $x0L1/gid/text(), ", ", $x0L1/project/text(), ")" }  </finId> </funding> ) }
    </project> )  }
   { distinct-values (
  FOR $x0 IN $doc0/S/grant, $x1 IN $doc0/S/project, $x2 IN $doc0/S/contact, $x3 IN $doc0/S/contact
  WHERE $x2/cid/text() = $x0/manager/text() AND $x0/supervisor/text() = $x3/cid/text() AND $x0/project/text() = $x1/name/text()
  RETURN
    <finance>
      <finId>  {  $x0/gid/text() }  </finId>
      <mPhone>  {  $x2/phone/text() }  </mPhone>
      <company>  {  "Sk46(", $x2/phone/text(), ", ", $x0/gid/text(), ")" }  </company> </finance> )  }
  { distinct-values (
  FOR  $x0 IN $doc0/S/grant, $x1 IN $doc0/S/project, $x2 IN $doc0/S/contact, $x3 IN $doc0/S/contact
  WHERE $x2/cid/text() = $x0/manager/text() AND  $x0/supervisor/text() = $x3/cid/text() AND  $x0/project/text() = $x1/name/text()
  RETURN
    <company>
      <coid>  {  "Sk46(", $x2/phone/text(), ", ", $x0/gid/text(), ")" }  </coid>
      <name>  {  "Sk49(", $x2/phone/text(), ", ", $x0/gid/text(), ")" }  </name> </company> )  }
  { distinct-values (
  FOR  $x0 IN $doc0/S/company
  RETURN
    <company>
      <coid>  {  "Sk93(", $x0/cname/text(), ")" }  </coid>
      <name>  {  $x0/cname/text() }  </name>  </company> )  }
```

The Road To Mapping Systems

- The design of data transformations has been a manual task for a long while
  - Designers had to be familiar with the language
  - As schemas became larger and more complex, the task became too laborious, time-consuming and error-prone
- The need of raising the level of abstraction and trying to automate the tasks was soon realized.
- The idea …



Generating Mapping

- Different techniques exist to generate mappings:
  - Manual, e.g.
    - by means of high-level mapping languages, such as [Bernstein et al. 2007]
    - by means of sophisticated user interfaces [Altova 2008]
  - Semi-automatic, e.g.
    - By means of designer guidance [Alexe2008]
    - Via advanced algorithms to do the reasoning instead of the mapping designer [Madhavan at al. 2001] [Popa et al. 2002][Do et al. 2002][Bonifati et al. 2008]

The First Step of a Mapping Task

Projects

name	status
PIX	Active
E-services	Active
Clio	Inactive

Grants

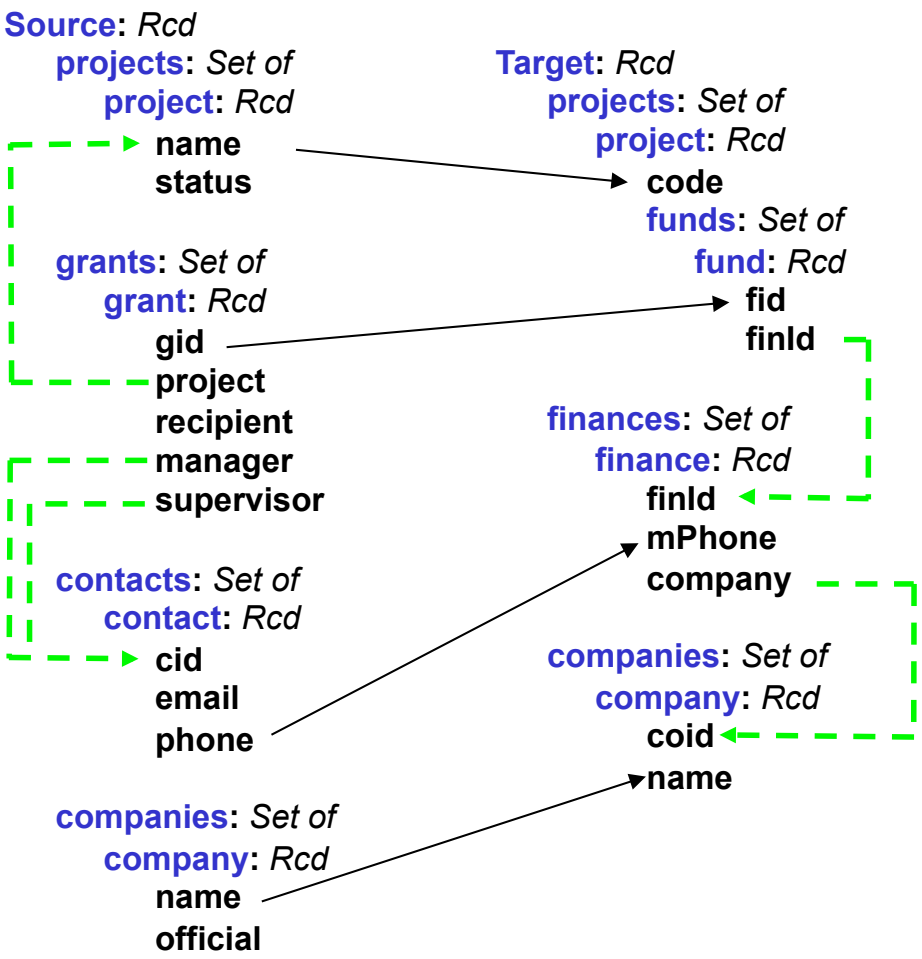
gid	project	recipient	manager	supervisor
g1	PIX	AT&T	Fernandez	Belanger
g2	PIX	AT&T	Shrivastava	Belanger
g3	E-services	Bell-labs	Benedikt	Hull

Contacts

cid	email	phone
Benedikt	benedikt@research.bell-labs.com	5827766
Hull	hull@research.bell-labs.com	5824509
Shrivastava	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



Matching

- Given two schemas as input, the source and the target schema, matching is a process that produces as output a set of matches or correspondences or (simply) lines between the elements of the two schemas

- A match is a triple <Es, Et, e>
  - Where Es is a set of elements of the source schema, Et is a set of elements of the target schema, and e specifies a simple relationship (equality or set inclusion) or complex relationship between element in Es and Et

The Matching Relationship e

- Depends on the cardinalities of Es and Et
- Depends on the semantics:
  - Can be a function
  - Can be an arithmetic operation
  - Can be a set-theoretic relation (e.g. ≡,overlaps)
  - Can be a conceptual modeling relationship (e.g. part-of, subclass-of)

Matching: An Alternative Definition

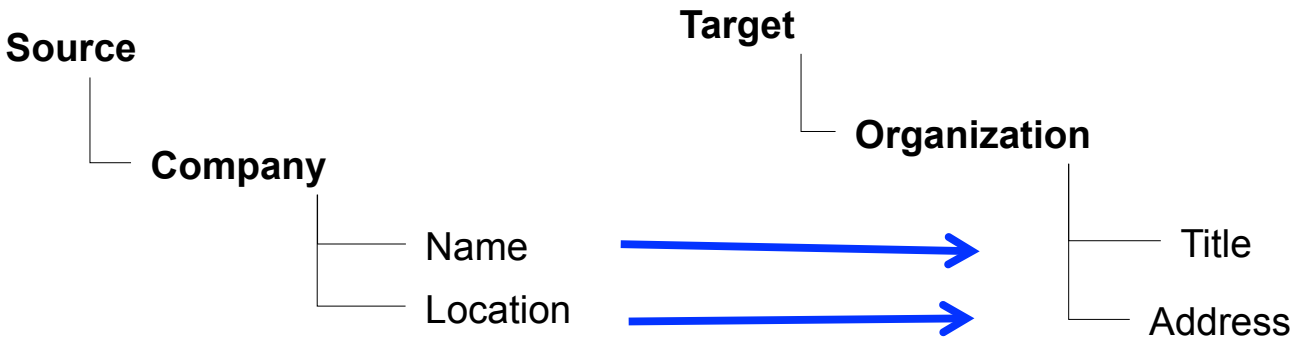
- The matching process [Euzenat et al. 2007] can be seen as a function f from a pair of schemas S and T, an optional input alignment A, a set of matching parameters p and a set of resources r:

$$A' = f(S, T, A?, p, r)$$

- Ultimately, an alignment is a set of correspondences between elements in S and elements in T

Matching Examples

- Simple relationship:
  - Name ≡ Title
  - Location ≡ Address



- Complex relationship:
  - speed = velocity x 2.237
  - speed x 0.447 = velocity
  - speed = concat(velocity x 2.237, ‘MPH’)
  - speed ≥ velocity

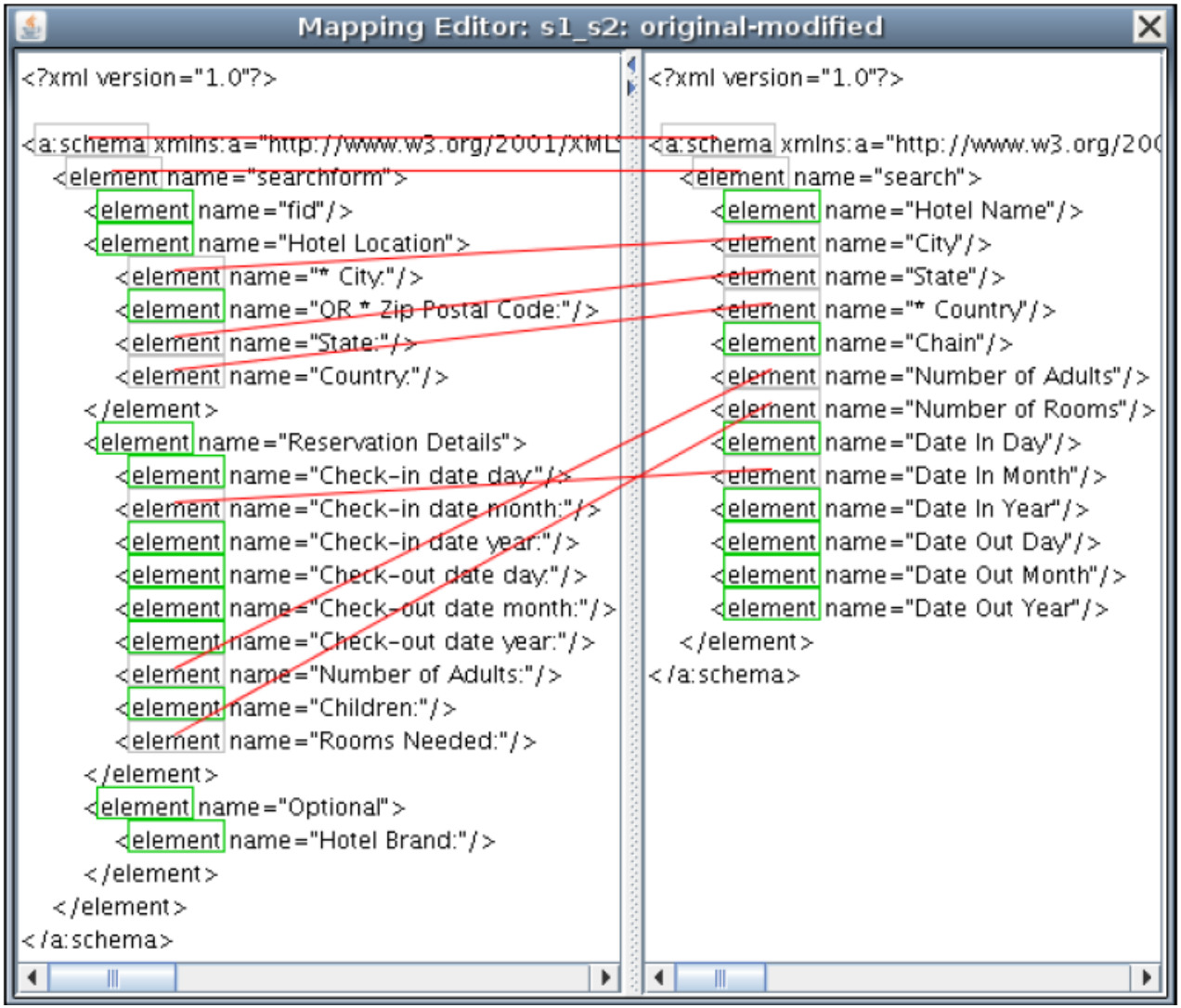
The matching process

- Can be roughly divided into three steps:
  - Pre-match: training of classifiers for machine learning-based matchers, matching parameters (weights, thresholds), adjustments of resources, such as thesauri and constraints
  - Match: the actual matching task
  - Post-match: the user may check and modify the displayed matches

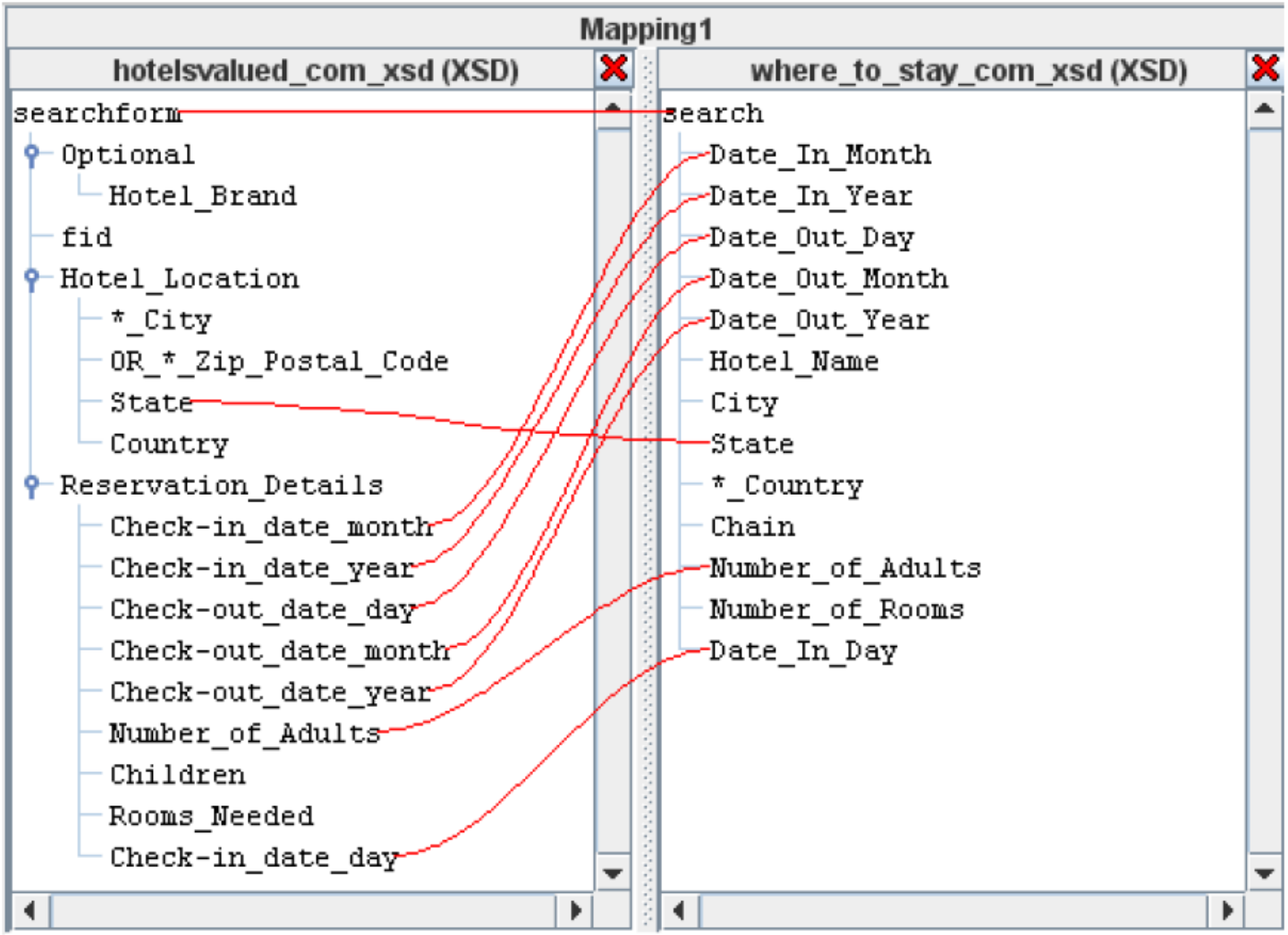
Some Schema Matchers

- Cupid [Madhavan et al. 2001] : based on structural and name similarity
- S-Match [Giunchiglia et al. 2004]: based on semantic closeness
- Coma++ [Aumueller et al. 2005]: based on matching reuse
- LSD [Doan et al. 2001]: based on data value analysis and machine-learning techniques
- iMap [Dhamankar et al. 2004]: suited for complex e expressions
- Similarity Flooding [Melnik et al. 2002]: based on graph similarity

Similarity Flooding



COMA++


Mapping1
hotelsvalued_com_xsd (XSD)
searchform
Optional
Hotel_Brand
fid
Hotel_Location
*_City
OR_*_Zip_Postal_Code
State
Country
Reservation_Details
Check-in_date_month
Check-in_date_year
Check-out_date_day
Check-out_date_month
Check-out_date_year
Number_of_Adults
Children
Rooms_Needed
Check-in_date_day
where_to_stay_com_xsd (XSD)
search
Date_In_Month
Date_In_Year
Date_Out_Day
Date_Out_Month
Date_Out_Year
Hotel_Name
City
State
*_Country
Chain
Number_of_Adults
Number_of_Rooms
Date_In_Day

Matchings Are Not Enough

Projects

name	status
PIX	Active
E-services	Active
Clio	Inactive

Grants

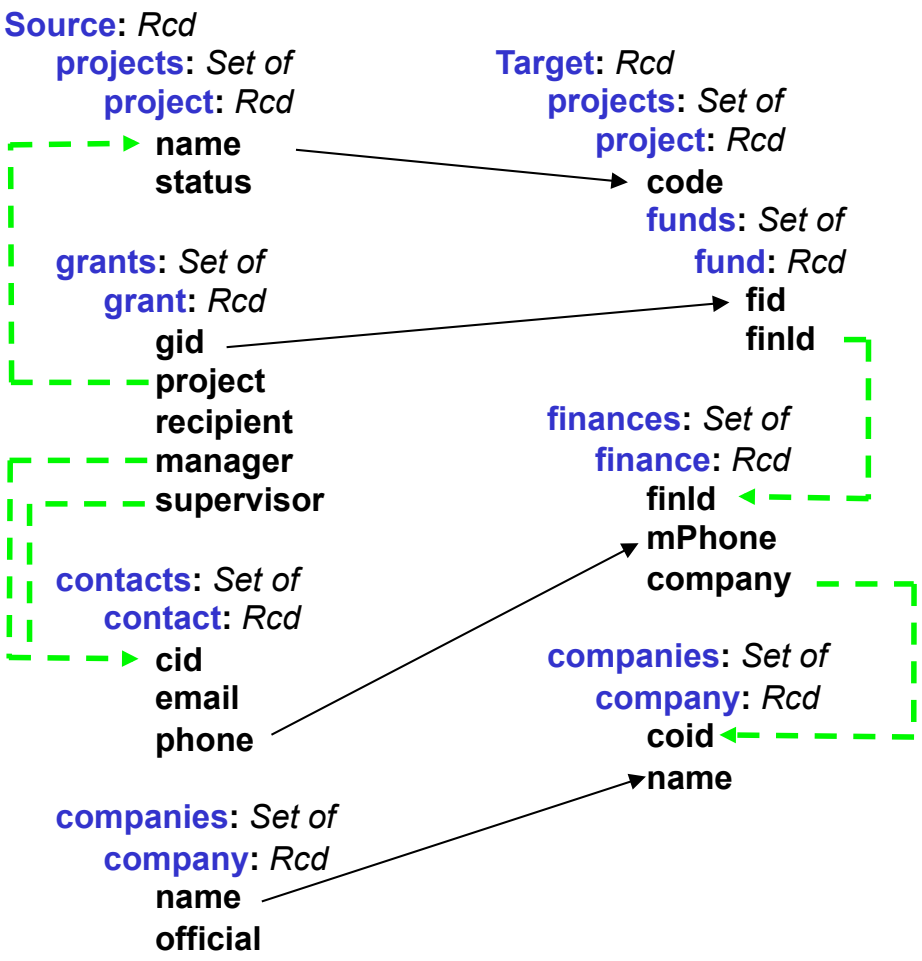
gid	project	recipient	manager	supervisor
g1	PIX	AT&T	Fernandez	Belanger
g2	PIX	AT&T	Shrivastava	Belanger
g3	E-services	Bell-labs	Benedikt	Hull

Contacts

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

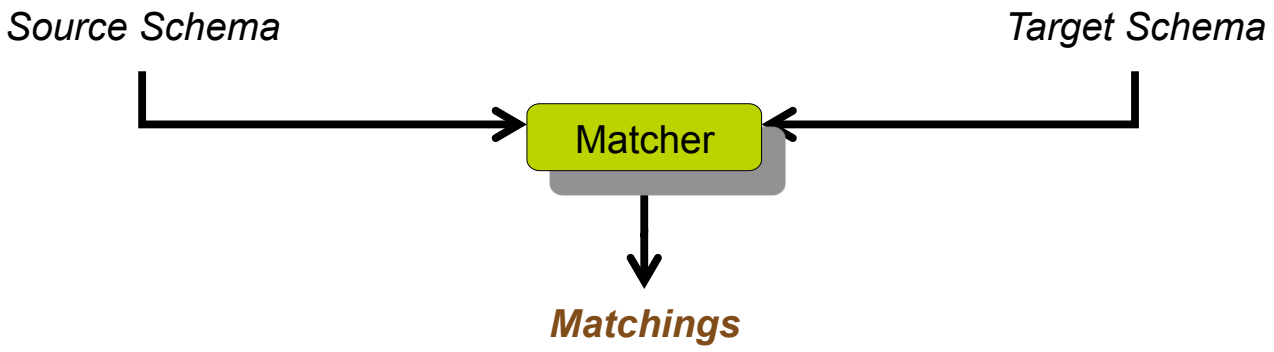
Companies

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



Cannot describe the full details of the transformation

The Mapping Generation Process



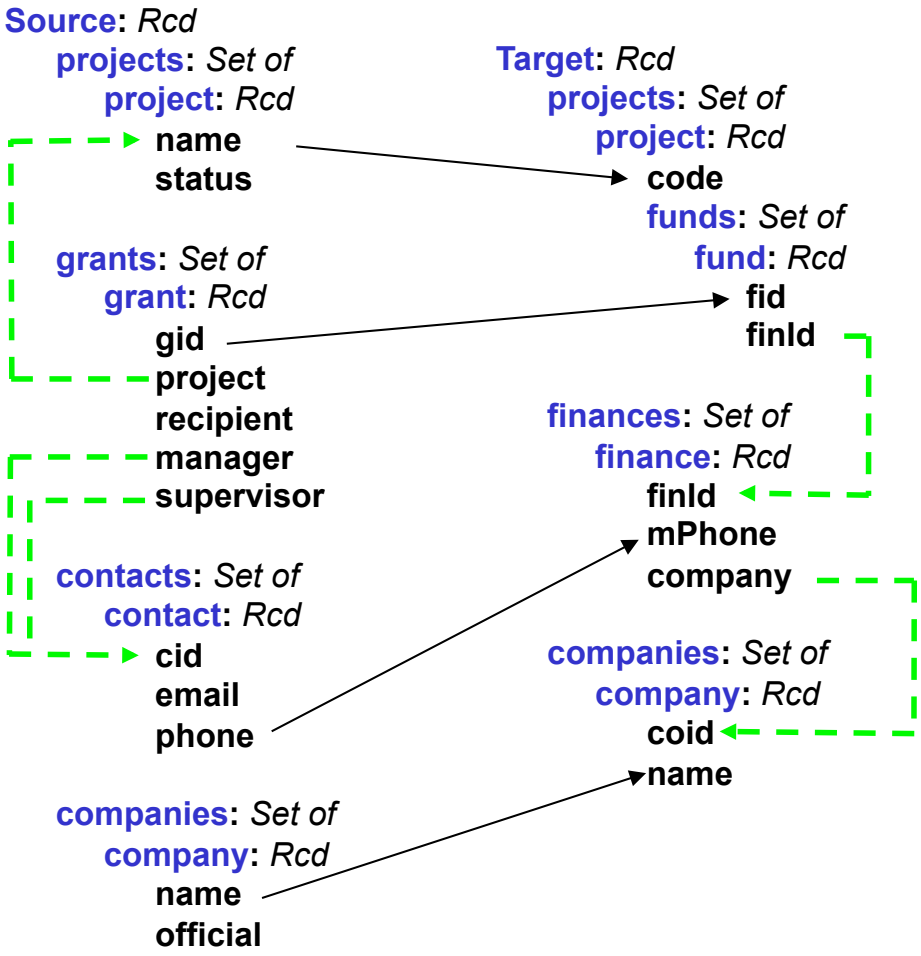
Matching is just the beginning of any mapping generation process

Mappings

- Given the source and the target schemas, mapping is a process that takes as input a set of matches between the elements of the two schemas and produces a relationship or constraint e that must hold between their respective instances

- In other words, a mapping is a triple <S, T, e>
  - Where S is the source schema, T is the target schema, and e specify a constraint that any instances adhering to S and T must satisfy or an executable statement to transform the instance of S into the instances of T

A Mapping Example



project(na,st), **grant**(gid,na,re,ma,su), **contact**(ma,em,ph) →

project(na,FUND), **fund**(gid,finId), **finance** (finId,ph,company)

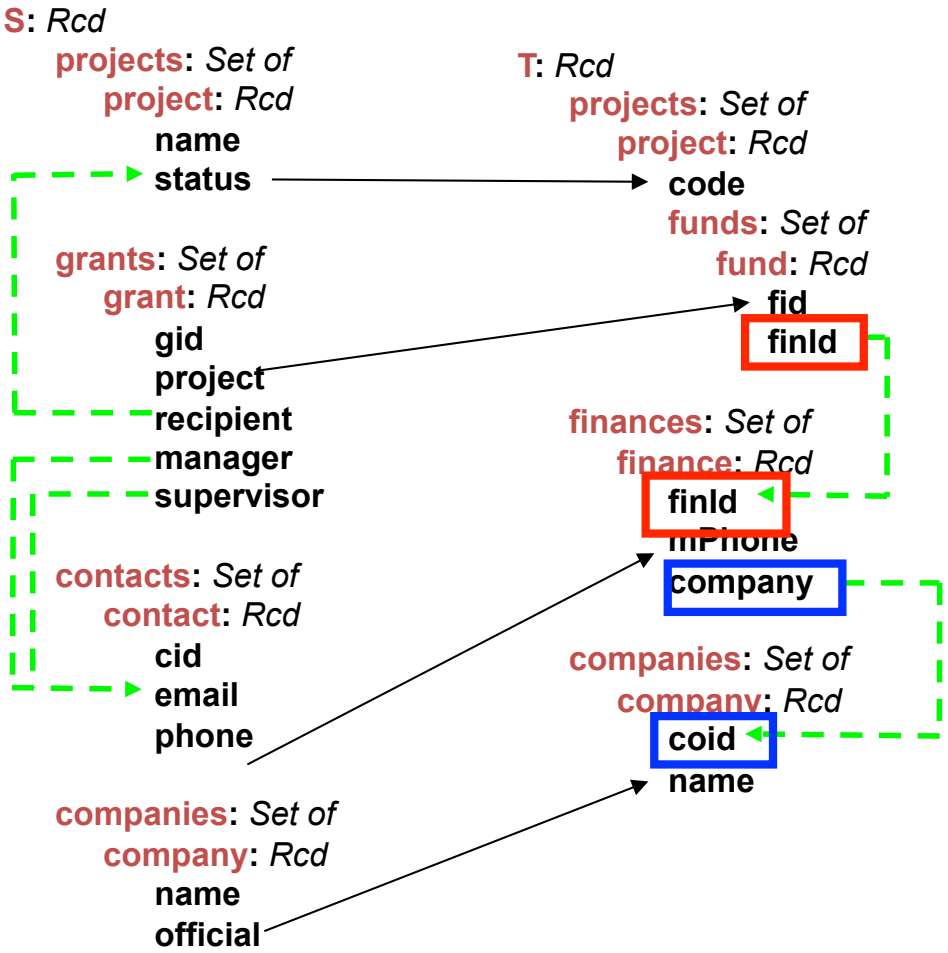
company(company, name)

Mappings & Instances

- Mappings are the basic ingredients of many tasks, such as information integration, P2P query answering, data exchange etc.

- In particular, mappings as inter-schema constraints may not be enough to fully specify a unique target instance
  - There may exist multiple target instances satisfying the mappings
  - Finding the best target instance is the goal of the data exchange problem [Fagin et al. 2005]
    - The mapping is converted into a executable transformation script to obtain that particular instance

A Data Exchange Example



Target instance

Projects

code: E-services

Funds

fid	finId
g3	???

code: PIX

Funds

fid	finId
g1	???
g2	???

Finances

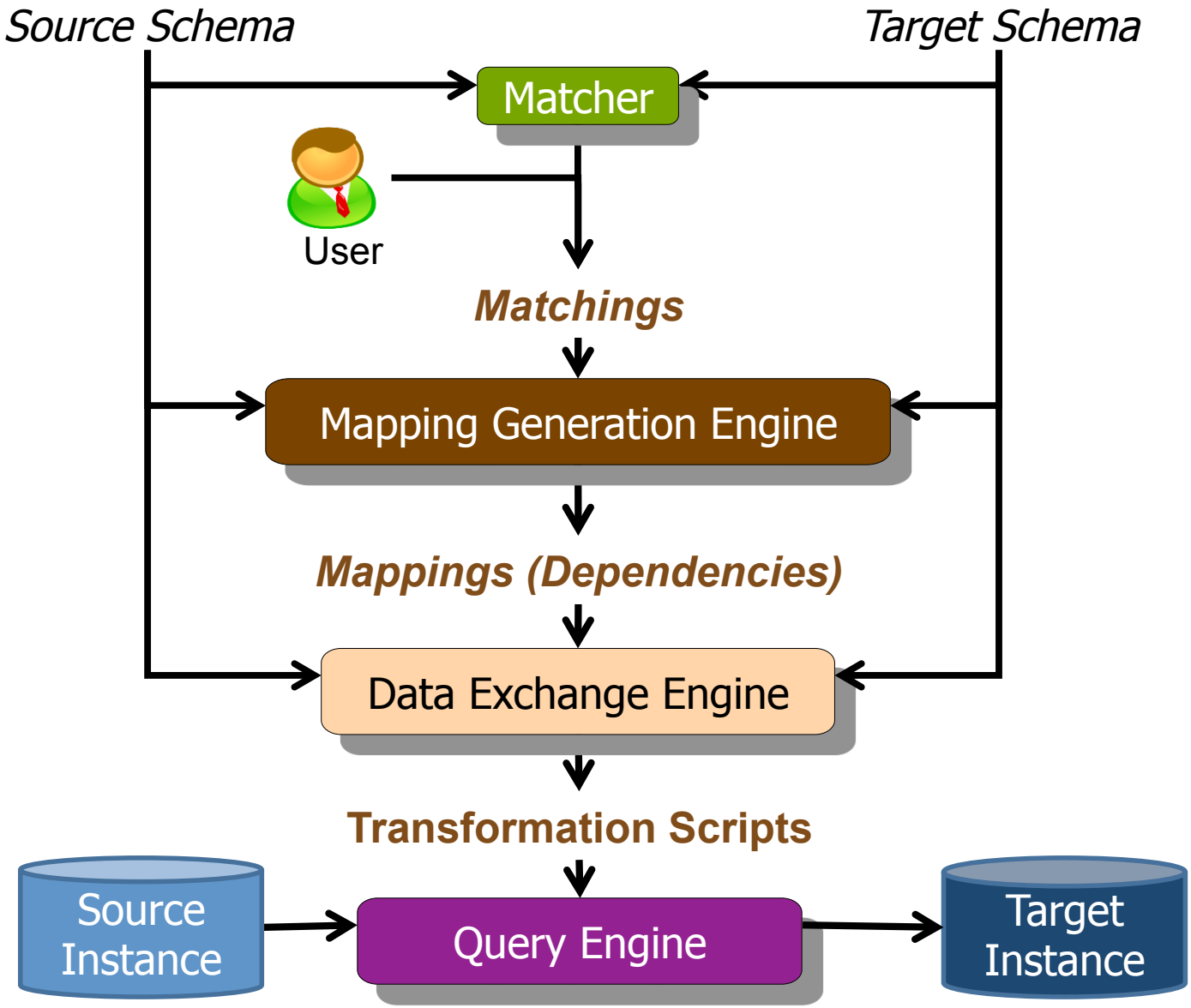
finId	mPhone	company
???	3608679	???
???	3608776	???
???	3608600	???

Companies

coid	name
???	AT&T
???	Lucent

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)

The Mapping Generation Process

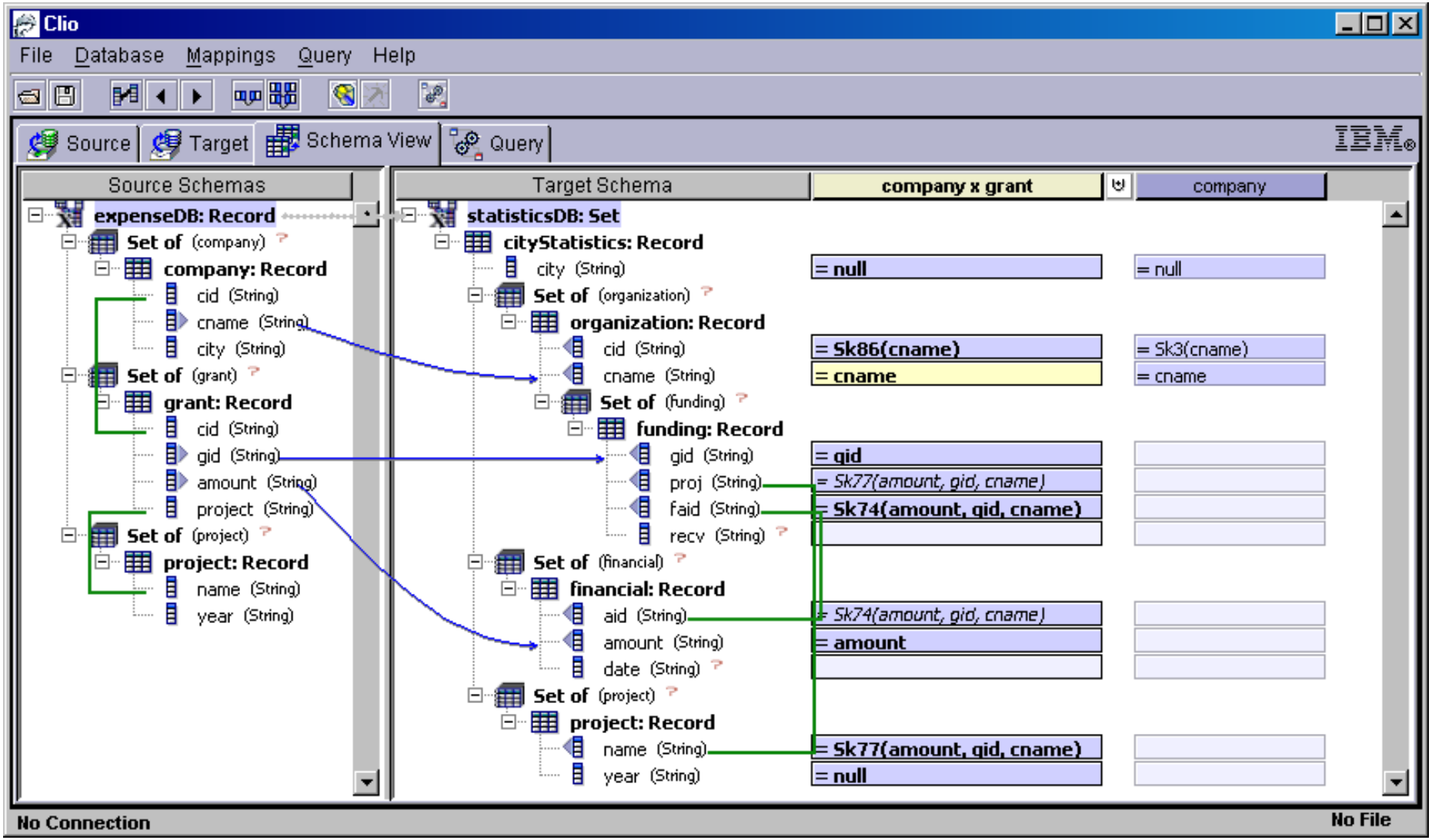


Research Prototype Systems

- Mapping generation and data exchange are separate tasks
  - Clio[Popa et al. 2002], HePToX[Bonifati et al. 2010], Spicy[Bonifati et al. 2008]
- Mappings Generation
  - the mappings are expressed as high-level assertions in a logical formalism
  - A mapping is a source-to-target tuple-generating dependency (or s-t tgd in short)

    $\phi(x) \rightarrow \exists\, \psi(x,y)$ where
  - $\varphi(x)$ ($\psi(x,y)$, resp.) is a conjunction of atoms over the source (target, resp.)
- Data exchange
  - The respective module transforms the high-level mappings into transformation scripts (in SQL or XQuery) to generate the target instance.

Clio


Clio
File Database Mappings Query Help
Source
Target
Schema View
Query
IBM
Source Schemas
expenseDB: Record
Set of (company)
company: Record
cid (String)
cname (String)
city (String)
Set of (grant)
grant: Record
cid (String)
gid (String)
amount (String)
project (String)
Set of (project)
project: Record
name (String)
year (String)
Target Schema
company x grant
company
statisticsDB: Set
cityStatistics: Record
city (String)
= null
= null
Set of (organization)
organization: Record
cid (String)
= Sk86(cname)
= Sk3(cname)
cname (String)
= cname
= cname
Set of (funding)
funding: Record
gid (String)
= gid
proj (String)
= Sk77(amount, gid, cname)
faid (String)
= Sk74(amount, gid, cname)
recv (String)
Set of (financial)
financial: Record
aid (String)
= Sk74(amount, gid, cname)
amount (String)
= amount
date (String)
Set of (project)
project: Record
name (String)
= Sk77(amount, gid, cname)
year (String)
= null
No Connection
No File

Spicy

SpicyGUI 0.7.7

File DataSource Map Spicy View Tools Window Help

View Best Mappings

Following is a list of the solutions that have
Select the ones you want to view

Rank	Quality
Solution #1	0.818
Solution #2	0.6222

univDBCourseDB.xml | Spicy - univDBCourseDB.xml

univDB
- departments
  - department [0..*]
    - name : (string)
    - {...}
      - staff [0..*]
        - id : (integer)
        - name : (string)
    - {...}
      - professor [0..*]
        - id : (integer)
        - name : (string)
        - position : (string)
- courses
  - course [0..*]
    - courseName : (string)
    - credits : (integer)
    - instructorId : (integer)
    - year : (integer)

0.0

0 0.5 1

courseDB
- course [0..*]
  - courseName : (string)
  - credits : (integer)
  - instructorName : (string)
  - year : (integer)

Output - Spicy

```
---------------- STARTING SEARCH WITH SIMILARITY=0.825
Combinations produced: 0
CANDIDATE MAPPING TASKS: 0
---------- STARTING SEARCH WITH SIMILARITY=0.82
Combinations produced: 2
CANDIDATE MAPPING TASKS: 2
---------- MAPPING TASK n. 1 of 2----------OVERALL QUALITY: 0.6222498914823225
---------- MAPPING TASK n. 2 of 2----------OVERALL QUALITY: 0.8179742044489537
Best quality: 0.8179742044489537
==========  SEARCH FINISHED. FOUND = 2 MAPPINGS ==========
==========
```

HepToX

Main

Query Translation

After moving from one Montreal to Boston, the patient Mary wants to query her problems and corresponding treatment descriptions for those problems administrated in hospital after 01/01/1990.

Source XQuery Set

- Doctor Scenario
- Patient Scenario
- Insurance Scenario
- Health Care Scenario

Target XQuery Set

- Doctor Scenario
- Patient Scenario
- Insurance Scenario
- Health Care Scenario

Original XQuery and its corresponding Tree Pattern Querie(s):

```
where $PName = "Mary"
  and $D > "01/01/1990"
  and $PatRef = $PID
return <Problem> $Prob </Problem>
    <Description> $TDesc </Description>
```

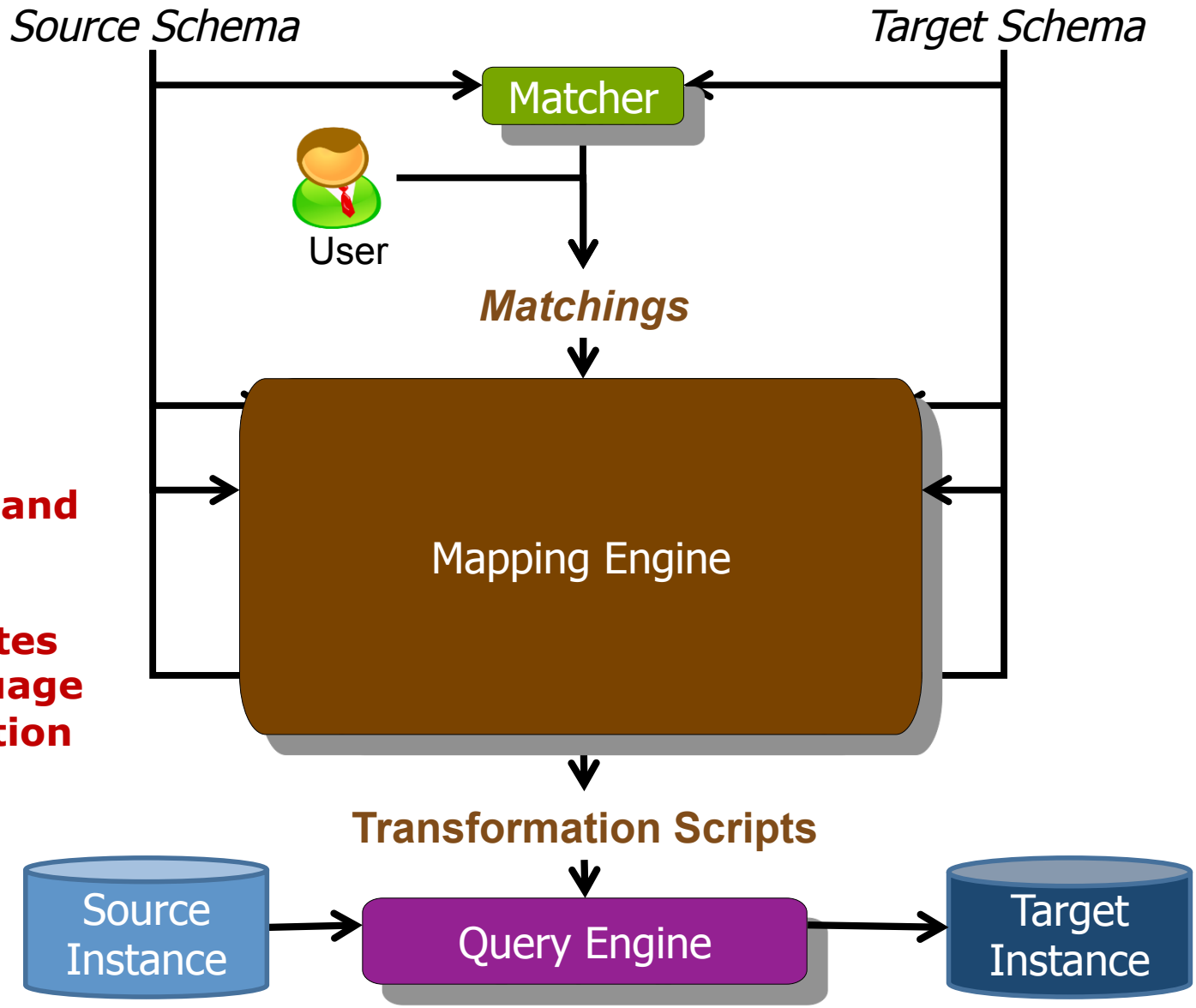
TP1:

Translated XQuery and its corresponding TPQ:

Show Final Result | Show Translation Process | Show Step for One Tree Pattern One Mapping Rule

An XQuery against the source DTD has been selected. Distinguished Nodes are displayed in RED. Please select a translation option above. Exit QT

Commercial Mapping Systems

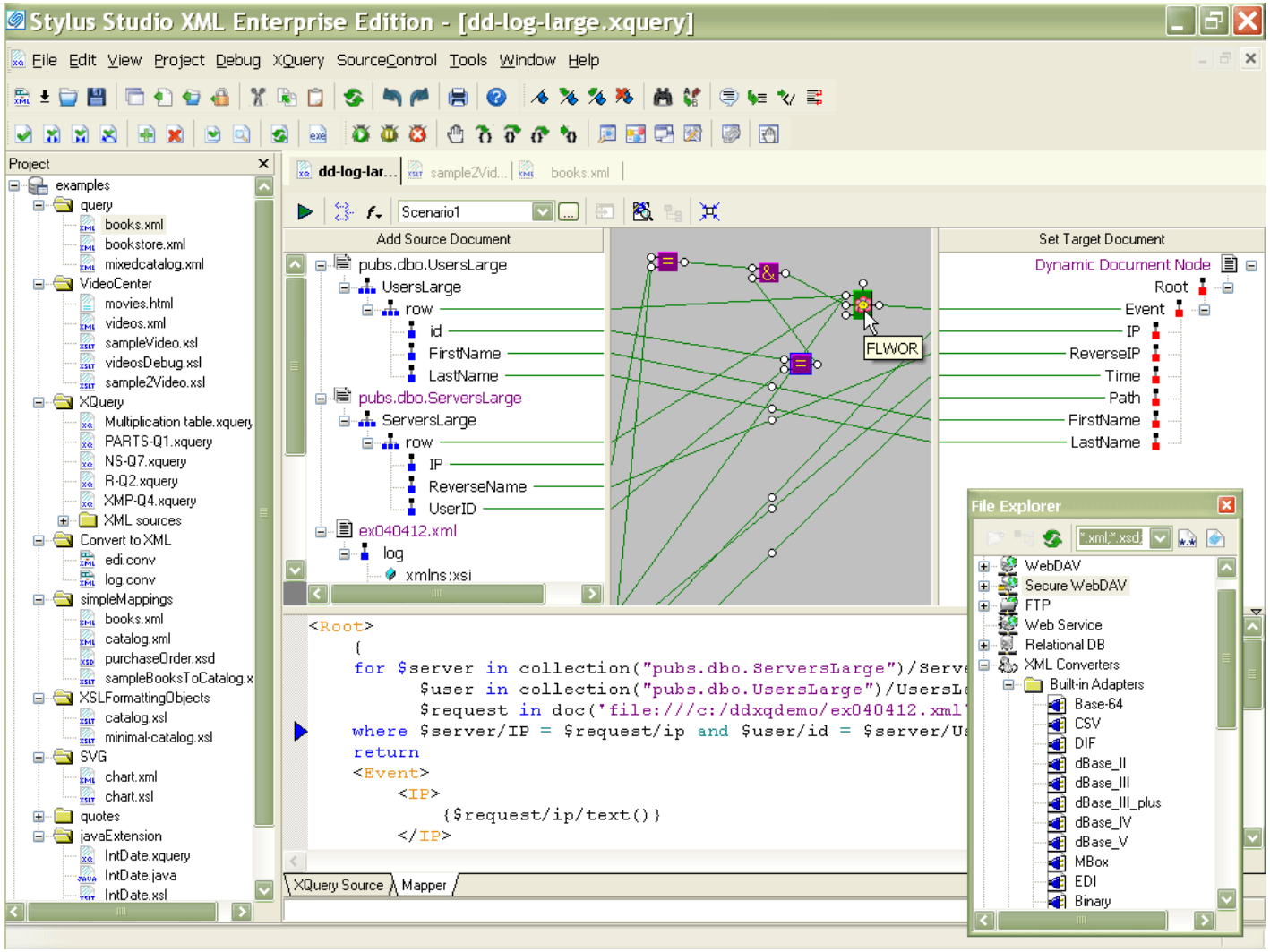


Mapping generation and Data exchange are merged in one. The system directly creates in some native language the final transformation script

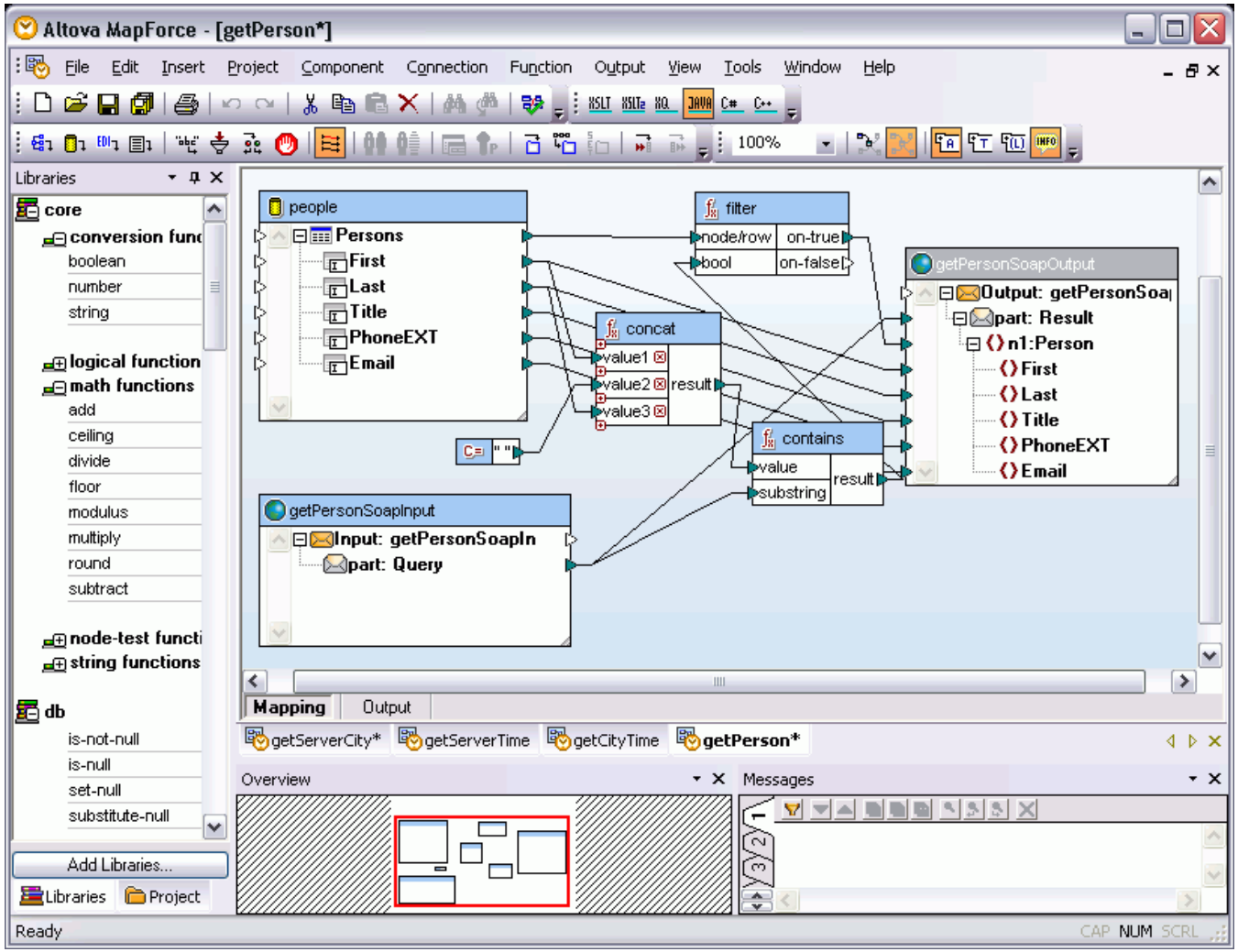
Popular Commercial Systems

- Altova Mapforce
- Stylus Studio
- IBM Rational Data Architect
- BizTalk mapper
- Adeptia
- BEA Aqualogic

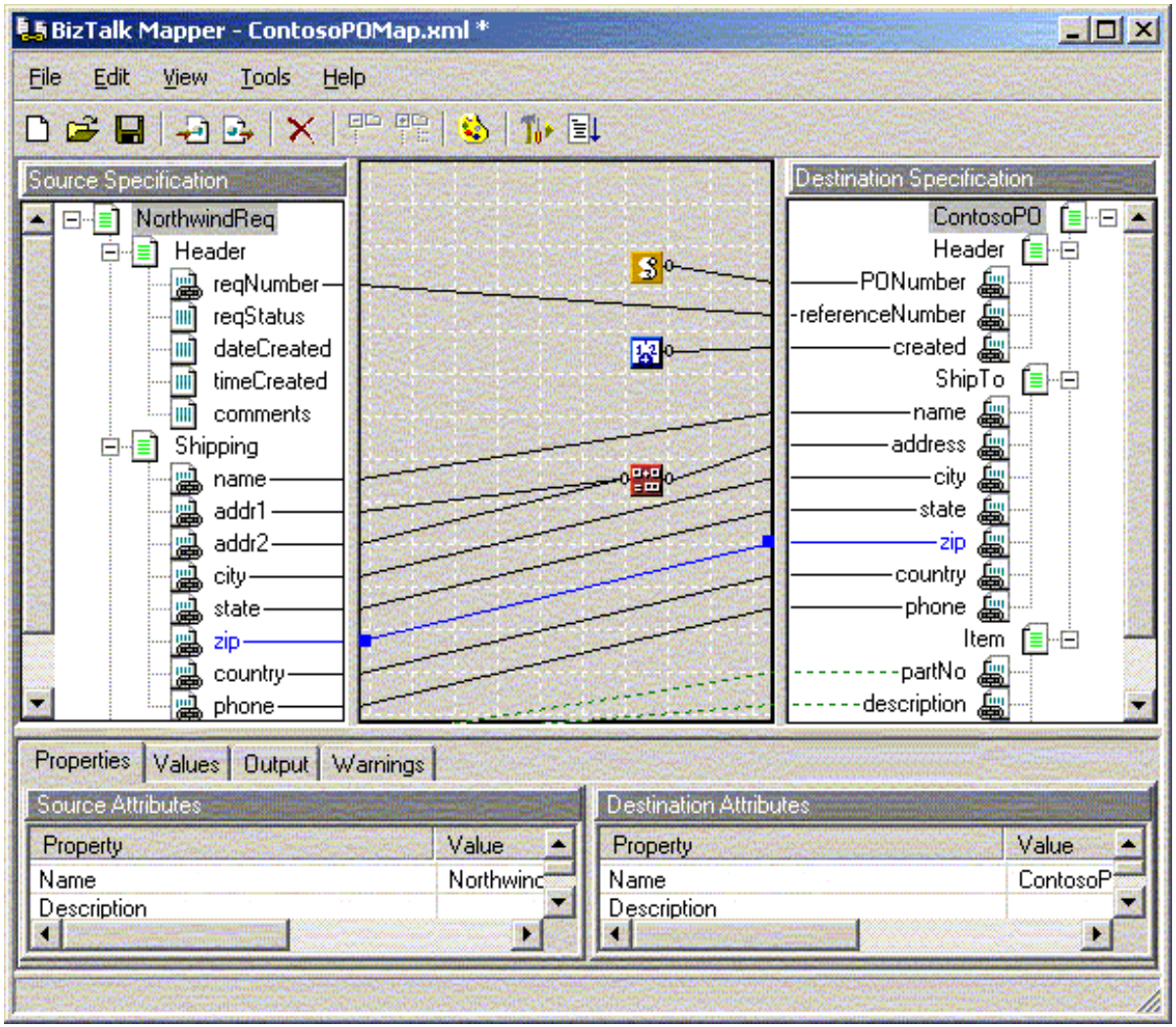
Stylus Studio



Altova MapForce

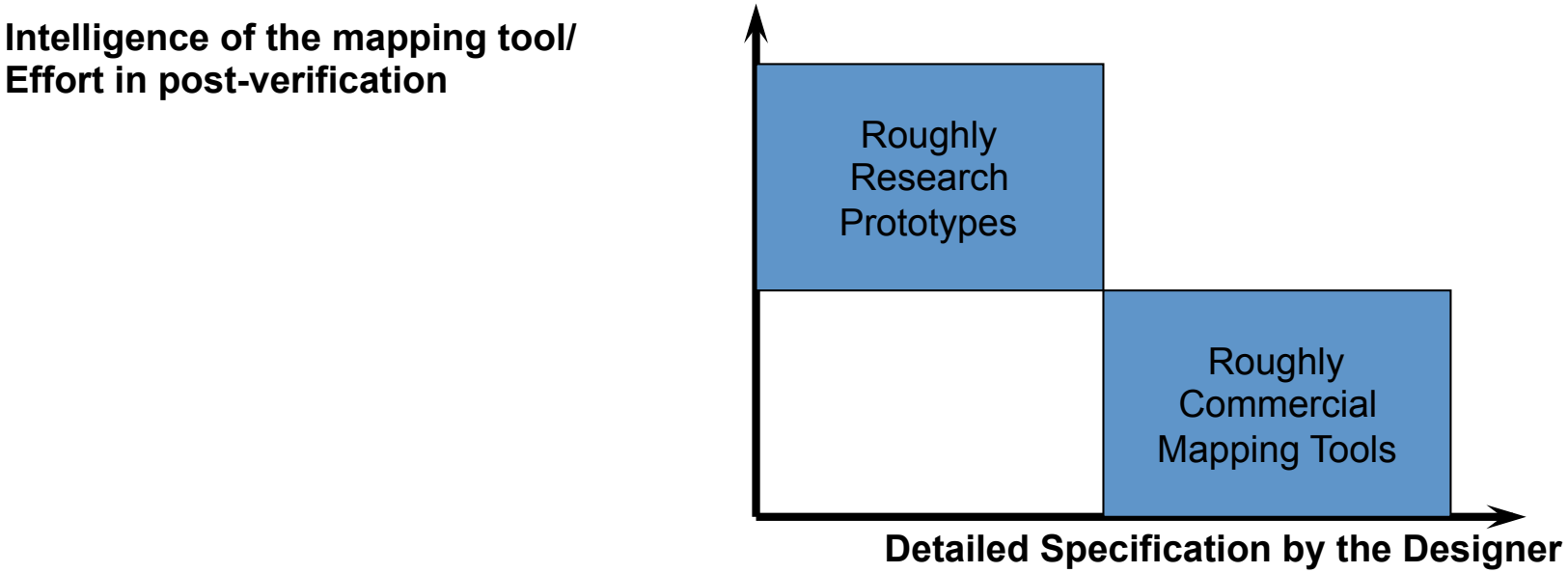


BizTalk Mapper



A Mapping Tool Categorization

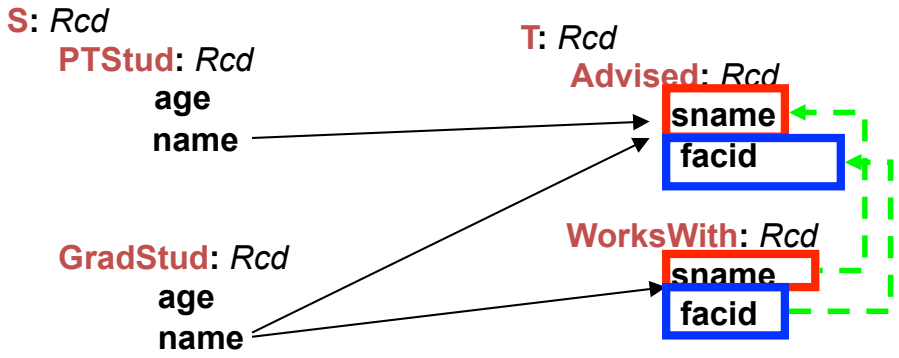
- All tools provide to the mapping designer:
  - A graphical representation of the two schemas
  - A set of graphical transformation constructs
    - The granularity and power of these constructs is a main factor of differentiation among the tools



Issues in Data Exchange

- When multiple target instances exist, how do we compute the best one?
- Is a given target instance better than another?
  - Universal solutions
    - Introduced in [Fagin et al. 2005]
    - These are the "most general" target instances, and also represent the entire space of solutions
    - Among the universal solutions, the smallest of all and the most compact one is called the "core"

Universal Core Instances



Source instance

PTStud

age	name
32	John
30	Ann

GradStud

age	name
27	Bob
30	Ann

A solution:

Advised

sname	facid
Bob	N3
Ann	N4
John	N1
Cathy	N1

WorksWith

sname	facid
Bob	N3
Ann	N4

Possible Target instances

A universal Solution:

Advised

sname	facid
Bob	N3
Ann	N4
John	N1
Ann	N2

WorksWith

sname	facid
Bob	N3
Ann	N4

The core:

Advised

sname	facid
Bob	N3
Ann	N4
John	N1

WorksWith

sname	facid
Bob	N3
Ann	N4

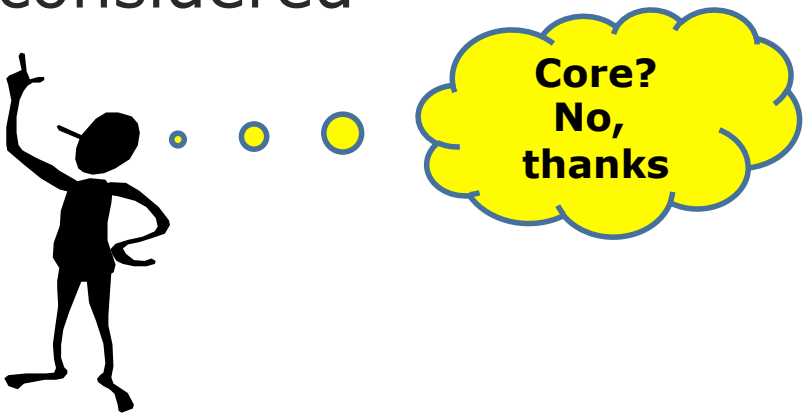
PTStud(x,y), → **Advised**(y,z)

GradStud (x,y) → **Advised**(y,z), WorksWith(y,z)

Commercial vs. Research Prototype Systems

- Whereas research prototypes (e.g. Clio, Spicy++) are tending to produce target instances that look more and more like the core

- Commercial tools leave the task to the users, who have to manually interact with sophisticated GUIs and write pieces of the transformation manually
  - No core definition is even considered



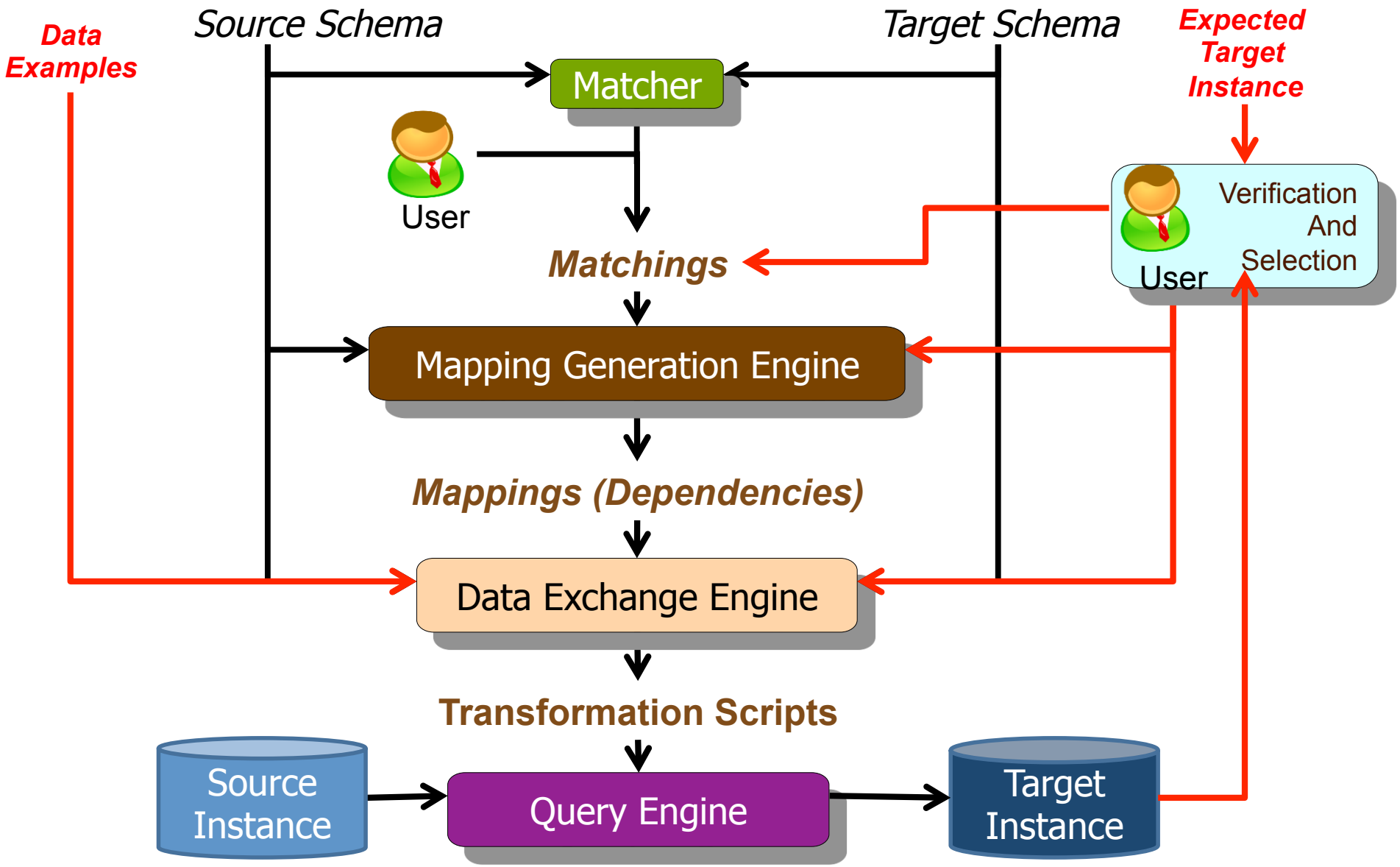
Mixing Matching and Mapping

- Matching and Mapping not always by separate tools
  - Clio has as an add-on a matcher based on attribute feature analysis [Naumann et al. 2002]
  - Bernstein's model management considers the matcher as a fully integrated and indistinguishable component
  - Spicy [Bonifati et al. 2008] has a matcher based on instance-based structural analysis

Limitations of Current Systems

- Manual approaches are not applicable to large-scale mapping tasks
- The user/developer has to become familiar with the mapping language and the user interfaces
- The outcome of the mapping process may not respect the user requirements and desired semantics (unsurprisingly!)
- Specifications may be incomplete and dependent of system peculiarities
- Thus, there is a need for a verification and guidance process

The Verification Process



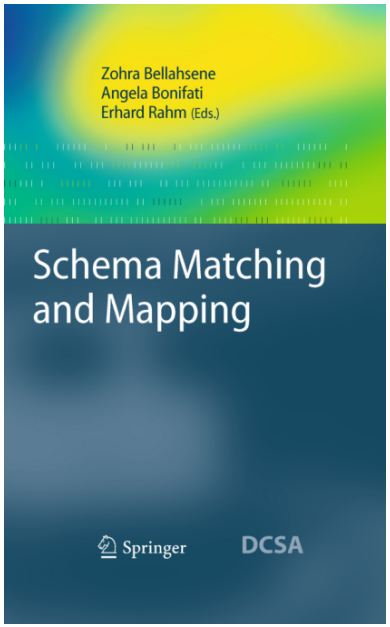
A-Posteriori Verification

- The main problem with matching and mapping is the dichotomy between the expected results and the generated answers
  - Some tools allow a post-verification
    - by using data examples
      - Tupelo [Fletcher et al. 2006], Muse[Alexe et al. 2008] Clio [Alexe et al. 2010]
    - by using an automatic instance comparison,
      - Spicy [Bonifati et al. 2008]
    - by means of manual user feedback
      - unfeasible for large-scale tasks
    - via debugging techniques
      - Routes [Chiticariu et al. 2006]

Conclusions and Ongoing work

- Interactive Mapping Refinement with User Data Examples
  - Independence of the graphical language
  - Approach driven by data examples that are simple to build for a generic user (few tuples/no core or universal data examples)
  - Obtained mappings have good properties (normalisation, absence of redundancy)
  - Joint work with U. Comignani (M2 Lyon1), E. Coquery and R. Thion
- Funding: ANR DataCert (2016-2020) & CNRS Mastodons MedClean 2016 & Palse Impulsion 2016

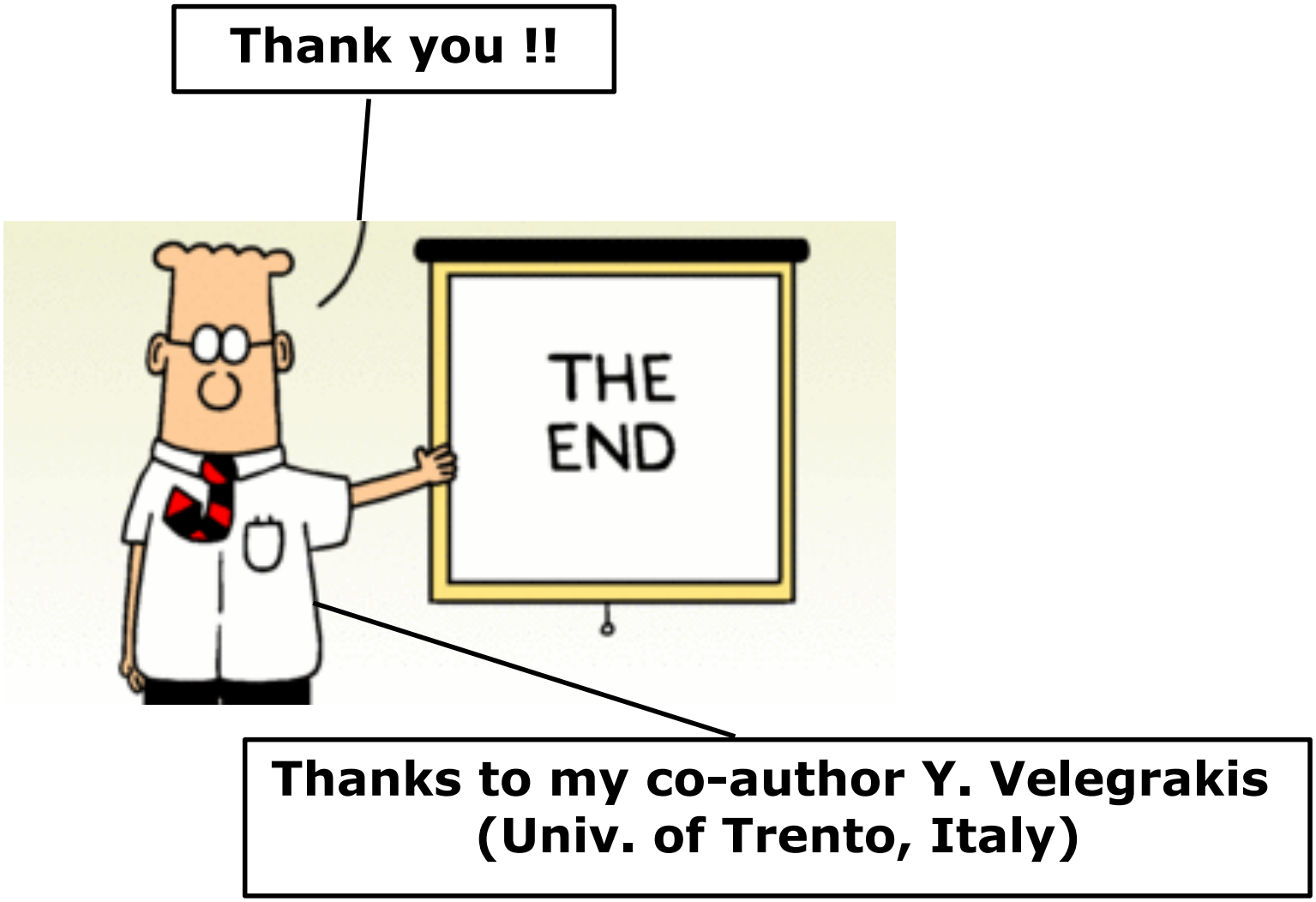
Main Sources



[Bellahsene et al. 2011] Z. Bellahsene, A. Bonifati and E. Rahm, *"Schema Matching and Mapping"*, *Springer-Verlag,* 2011

[Euzenat et al. 2007] J. Euzenat and P. Shvaiko, *"Ontology matching"*, *Springer-Verlag*, 2007

Thanks and Acknowledgements



List of References (cont'd)

- [Alexe et al. 2008] Alexe B, Chiticariu L, Miller RJ, Tan WC (2008) "Muse: Mapping Understanding and deSign by Example". In: ICDE, pp 10-19
- [Bonifati et al. 2008] Bonifati A, Mecca G, Pappalardo A, Raunich S, Summa G (2008) "Schema Mapping Verification: The Spicy Way". In: EDBT, pp 85-96
- [Lenzerini 2002] Lenzerini Maurizio "Data Integration: A Theoretical Perspective". In: PODS, pp 233-246
- [Miller et al. 2000] Miller RJ, Haas LM, Hernandez MA (2000) "Schema Mapping as Query Discovery". In: VLDB, pp 77-88
- [Rahm et al. 2001] Rahm E, Bernstein PA (2001) "A survey of approaches to automatic schema matching". VLDB Journal 10(4): 334-35
- [Velegrakis 2005] Velegrakis Y "Managing Schema Mappings in Highly Heterogeneous Environments". PhD thesis, University of Toronto

List of References (cont'd)

- [Altova 2008] Altova (2008) MapForce. Http://www.altova.com
- [Batini et al. 1986] Batini C, Lenzerini M, Navathe SB (1986) A Comparative Analysis of Methodologies for Database Schema Integration. ACM Comp. Surv. 18(4):323-364
- [Bernstein et al. 2007] Bernstein PA, Melnik S (2007) Model management 2.0: manipulating richer mappings. In: SIGMOD, pp 1–12
- [Do et al. 2002] Do HH, Rahm E (2002) COMA - A System for Flexible Combination of Schema Matching Approaches. In: VLDB, pp 610–621
- [Euzenat et al. 2007] Euzenat J, Shvaiko P (2007) Ontology matching. Springer Verlag, Heidelberg
- [Fagin et al. 2005] Fagin R, Kolaitis PG, Miller RJ, Popa L (2005) Data exchange: semantics and query answering. Theoretical Computer Science 336(1):89-124
- [Lerner 2000] Lerner BS (2000) A Model for Compound Type Changes Encountered in Schema Evolution. TPCTC 25(1):83–127
- [Popa et al, 2002] Popa L, Velegrakis Y, Miller RJ, Hernandez MA, Fagin R (2002)Translating Web Data. In: VLDB, pp 598–609

List of References (cont'd)

- [Alexe2010] Alexe B, Kolaitis PG, Tan W (2010b) Characterizing Schema Mappings via Data Examples. In: PODS
- [Aumueller2006] Aumueller D, Do HH, Massmann S, Rahm E (2005) Schema and ontology matching with COMA++. In: SIGMOD, pp 906–908
- [Bonifati et al. 2010] Angela Bonifati, Elaine Qing Chang, Terence Ho, Laks V. S. Lakshmanan, Rachel Pottinger, Yongik Chung: Schema mapping and query Translation in heterogeneous P2P XML databases. VLDB J. 19(2): 231-256 (2010)
- [Chiticariu et al. 2006] Chiticariu L, TanWC(2006) Debugging Schema Mappings with Routes. In: VLDB, pp 79–90
- [Dhamankar at al. 2004] Robin Dhamankar, Yoonkyong Lee, AnHai Doan, Alon Y. Halevy, Pedro Domingos: iMAP: Discovering Complex Mappings between Database Schemas. SIGMOD Conference 2004: 383-394
- [Doan et al. 2001] Doan A, Domingos P, Halevy AY (2001) Reconciling schemas of disparate data sources: A machine-learning approach. In: SIGMOD, pp 509–520