UNIVERSITÄT LEIPZIG

Object Matching in P2P Data Integration

(From COMA to MOMA)

Erhard Rahm
D. Aumüller, H.H. Do, T. Kirsten,
H. Köpcke, S. Maßmann, A. Thor

http://dbs.uni-leipzig.de

Exchange and Integration of Data (EID06)
June 10, 2006

Matching objects in web sources

DBLP

```
@article{DBLP:journals/vldb/RahmB01,
 author = {Erhard Rahm and Philip A. Bernstein},
 title  = {A survey of approaches to automatic schema matching.}
 journal= {VLDB J.}, year   = {2001}, ...
```

Google Scholar

A survey of approaches to automatic schema **matching** - group of 25 »
EJ Rahm, PAJ Bernstein - The VLDB Journal The International Journal on Very Large
... In the next section, we summarize some example applica- tions of **schema match**
5 provides a classification of different ways to perform Match **automatically**. ...
Cited by 585 - Web Search

Information Fusion

ACM

A survey of approaches to automatic schema matching

Full text Pdf (196 KB)

Source **The VLDB Journal — The International Journal on Very Large Data Bases** archive
Volume 10 , Issue 4 (December 2001) table of contents
Pages: 334 - 350
Year of Publication: 2001
ISSN:1066-8888

Authors Erhard Rahm Universität Leipzig, Institut für Informatik, 04109 Leipzig, Germany; (e-mail: rahm@informatik.uni-leipzig.de)
Philip A. Bernstein Microsoft Research, Redmond, WA 98052-6399, USA; (e-mail: philbe@microsoft.com)

Publisher Springer-Verlag New York, Inc. Secaucus, NJ, USA

Additional Information: abstract citings index terms collaborative colleagues peer to peer

Duplicates within (integrated) sources

Duplicates due to
- Order of authors
- Extraction error (title)
- Different titles
- Typos (author name)

Additional authors (!)

Classification of data quality problems*

Data Quality Problems

Single-Source Problems

Schema Level

(Lack of integrity constraints, poor schema design)

- Uniqueness
- Referential integrity

…

Instance Level

(Data entry errors)

- Misspellings
- *Redundancy/duplicates*
- Contradictory values

…

Multi-Source Problems

Schema Level

(Heterogeneous data models and schema designs)

- Naming conflicts
- Structural conflicts

…

Instance Level

(Overlapping, contradicting and inconsistent data)

- Inconsistent aggregating
- Inconsistent timing

…

Object matching

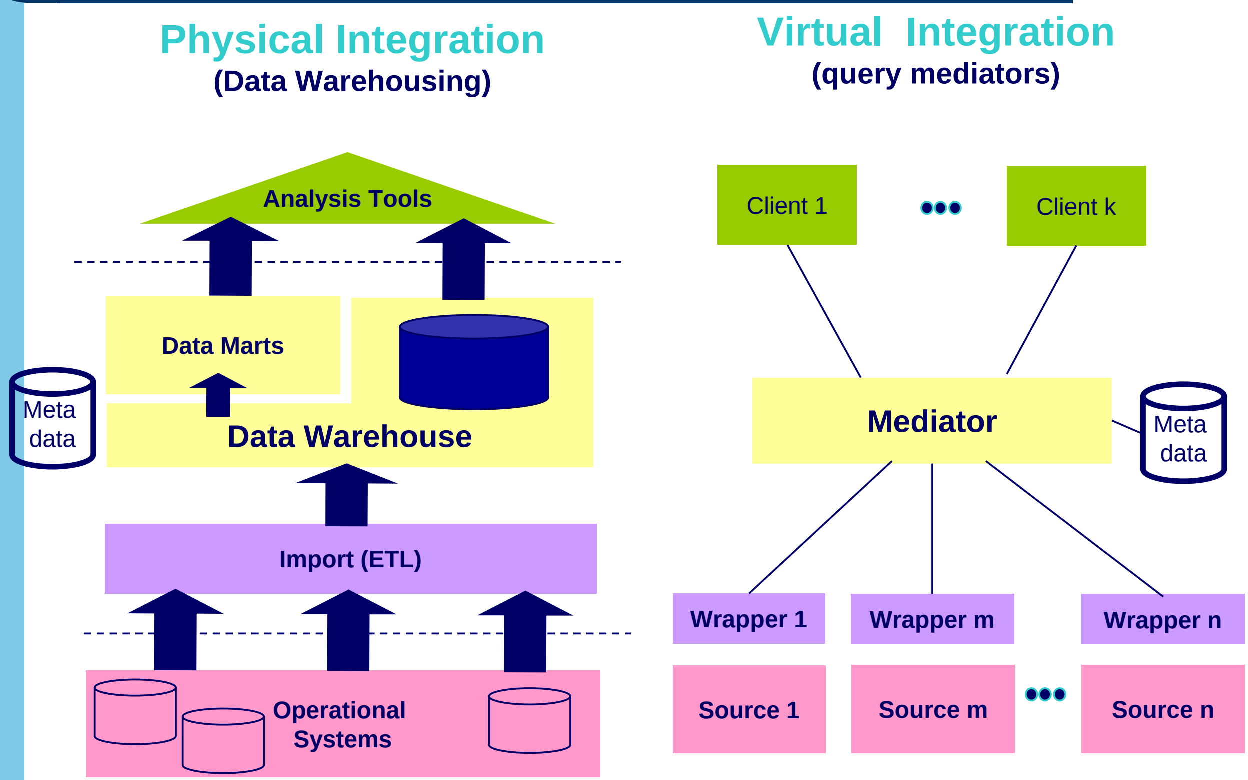
(entity resolution, duplicate detection, reference reconciliation …)

* E. Rahm, H. H. Do: *Data Cleaning: Problems and Current Approaches.* IEEE Techn. Bull. Data Eng., Dec. 2000

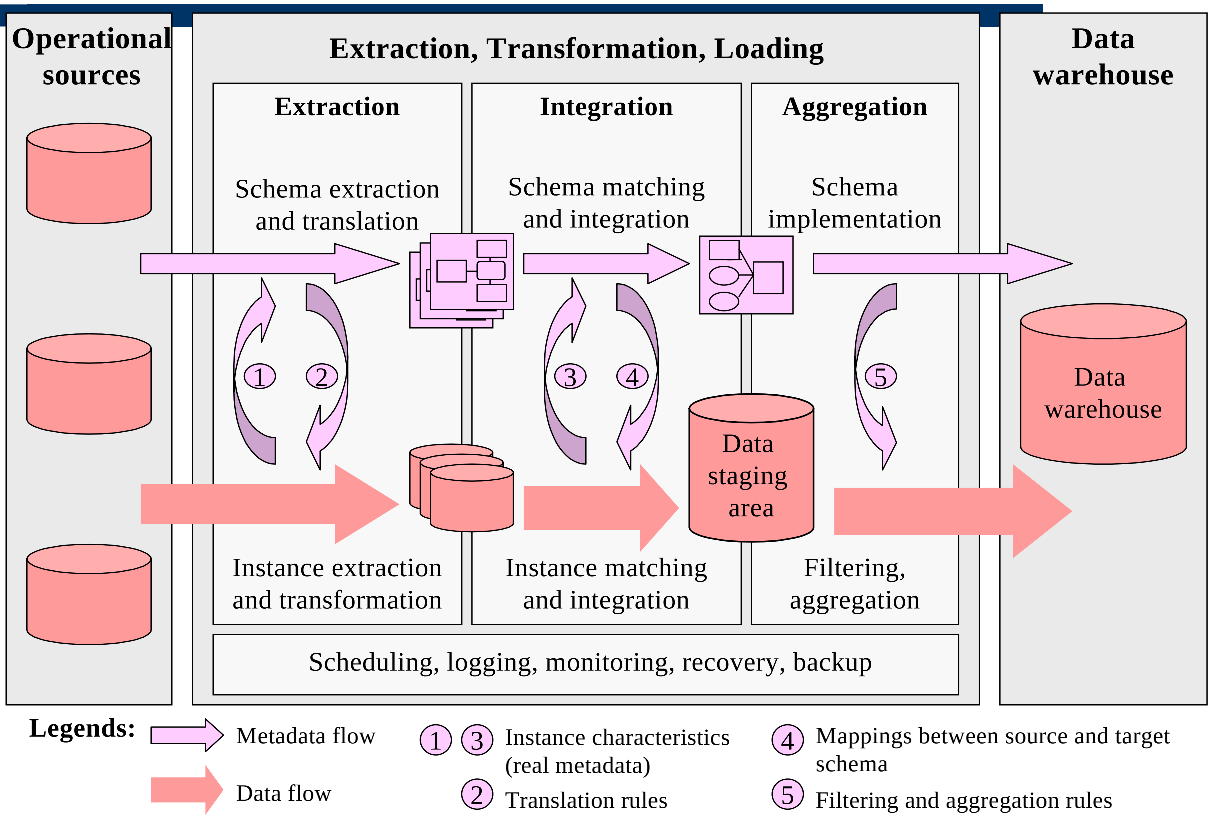
Agenda

- Motivation
- Data integration alternatives
  - Physiscal vs. virtual data integration
  - P2P-like data integration
  - Requirements for mapping-based data integration
- The iFuice data integration system
  - Sources, mappings, domain model
  - Operators
  - Use case: citation analysis of database publications
- MOMA Framework for Mapping-based Object MAtching
  - Architecture
  - Match strategies
  - Evaluation
  - Self-Tuning
- Related Work
- Summary

Data integration: physical vs. virtual



ETL Process*

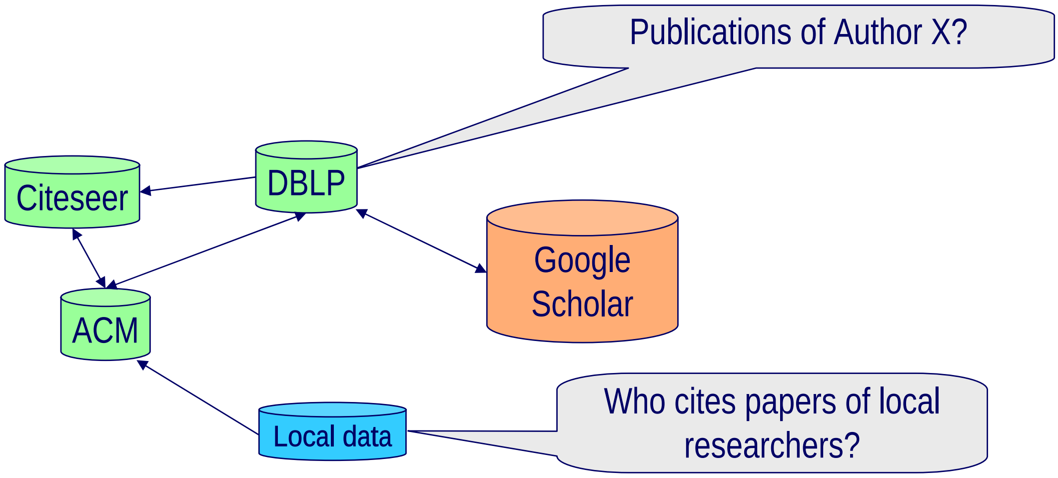


* E. Rahm, H. H. Do: *Data Cleaning: Problems and Current Approaches.* IEEE Techn. Bull. Data Eng., Dec. 2000

Data integration: physical vs. virtual (2)

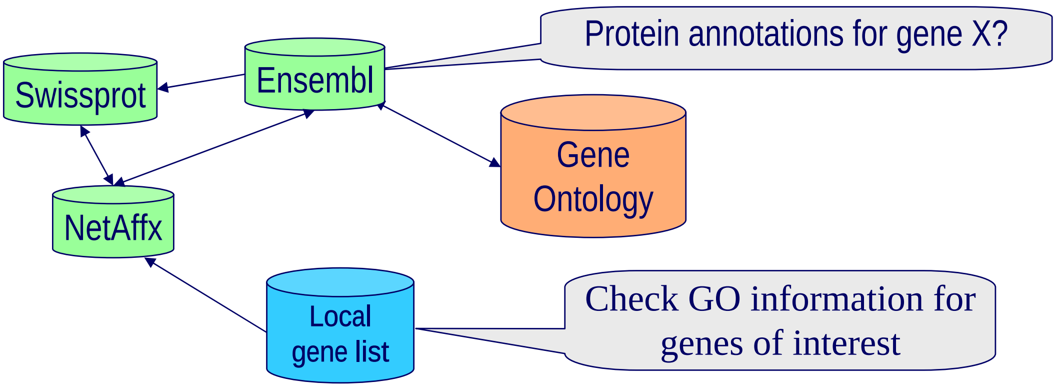
	Physical (Warehouse)	Virtual (Query mediators)
Schema integration	A priori	A priori
Instance data integration	A priori	At query runtime
Achievable data quality	+	o
Analysis of large data volumes	+	-
(HW) ressource requirements	-	o
Data freshness	o	+
Source autonomy	o	+
Scalability to many sources	-	-

P2P Integration: Bibliographic scenario

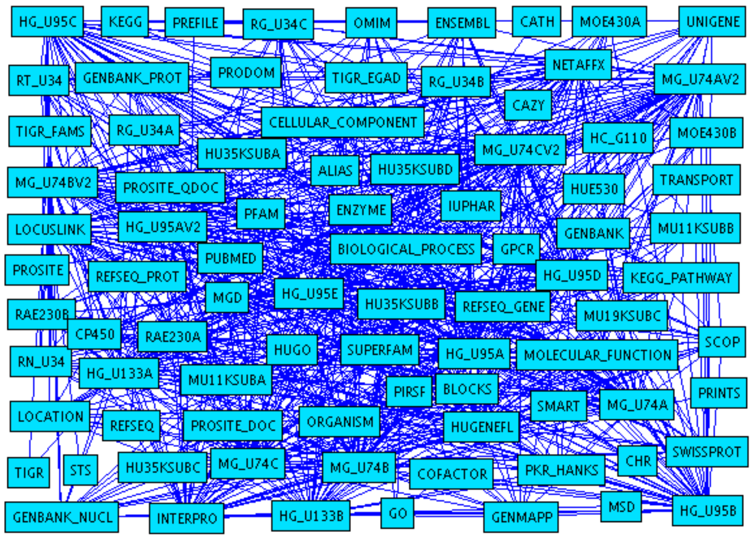


- Bidirectional mappings between data sources instead of global schema
- Queries refer to single source and are propagated to relevant peers
- Adding new sources becomes simpler
  - Support for local data sources

P2P: Bioinformatics scenario



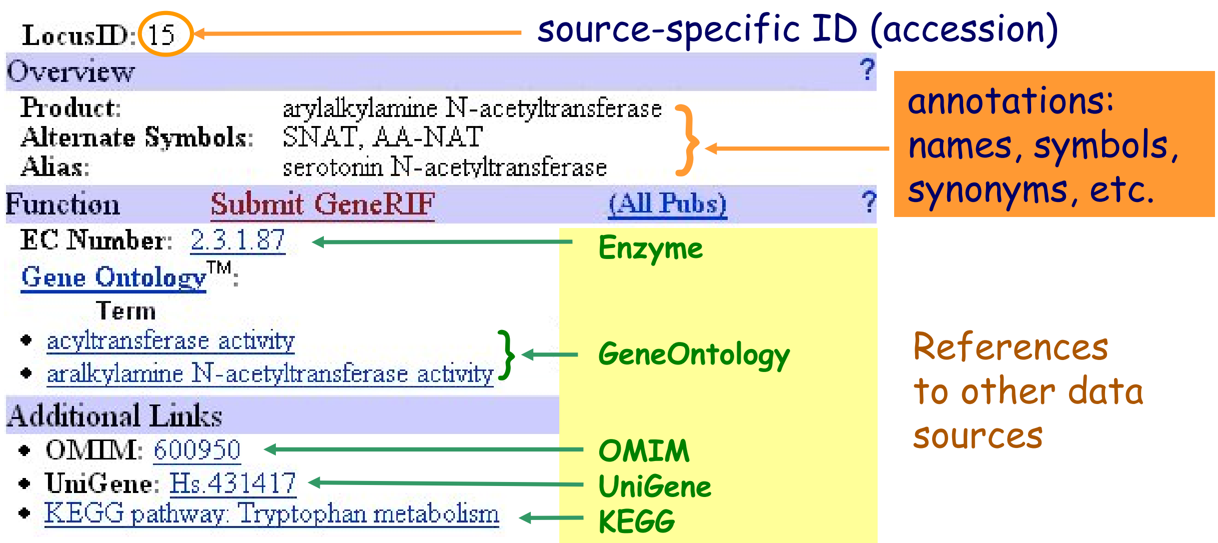
Data integration in bioinformatics



- many, highly connected data sources
- heterogeneous schemas, formats, semantics
- incomplete data sources
- frequent changes

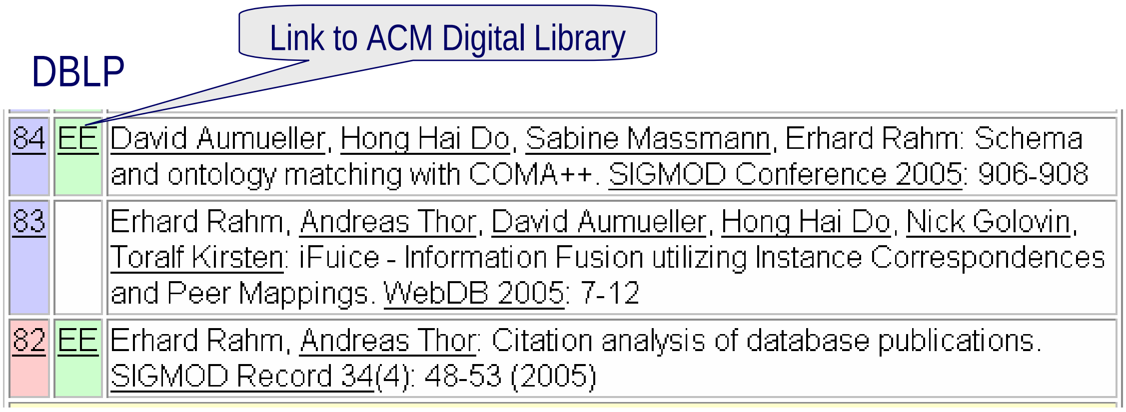
- global schema ???

Sample web data with cross-references

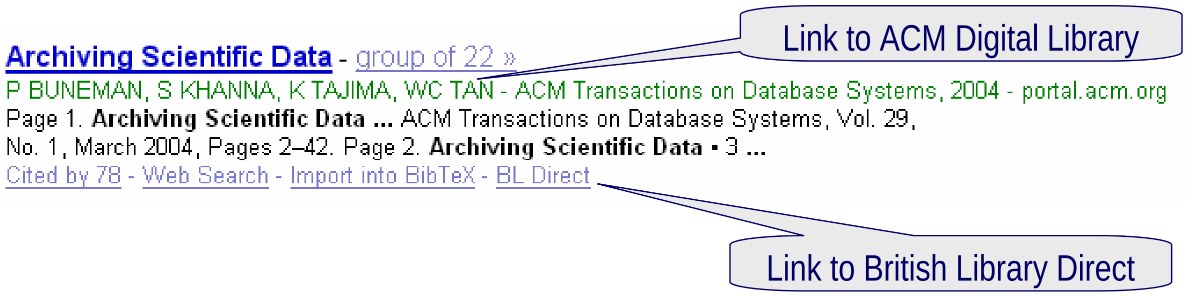


Bibliographic cross-references

- DBLP



- Google Scholar



Requirements for data integration (not only for bioinformatics)

- Utilization of instance-level cross-references (often manually curated, high quality data)
- Easy navigational and query access to many sources
- Support for ad-hoc analysis workflows
- Often no full transparency: users want to know from which sources data comes (data lineage / provenance)
- Need to integrate local (non-public) data
- Need for object matching:
  - data quality
  - create cross-references for information fusion

Solution: Mapping-based P2P-like data integration

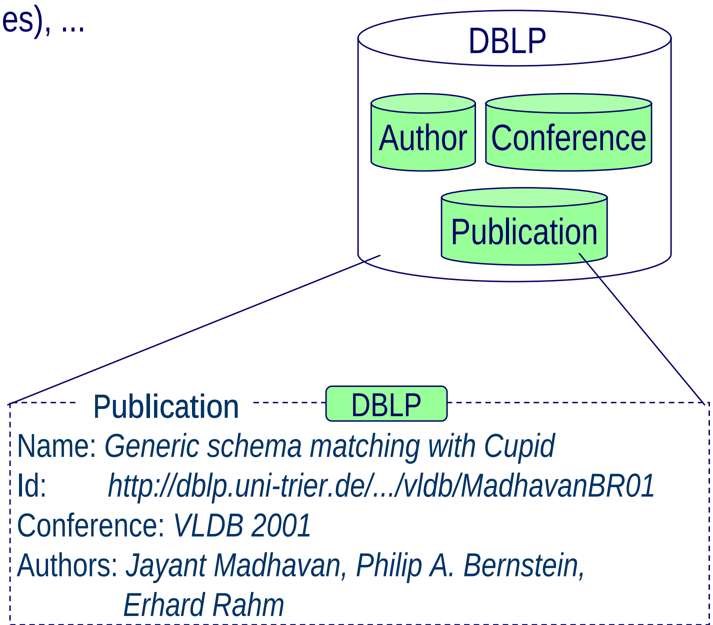
The iFuice approach*

- Information Fusion utilizing Instance Correspondences and Peer Mappings
- Generic infrastructure for data integration
  - Applicable to different application domains
  - Different types of sources (databases, files, …)
- Mapping-based (P2P-like) Integration
  - no global schema, but bidirectional mappings between data sources
  - utilization of existing cross-references / instance mappings
- Set-oriented operators for accessing data sources, traversing mappings, fusing data etc.
- Ontological domain model to categorize sources and mappings

*Rahm, E., et al.: *iFuice - Information Fusion utilizing Instance Correspondences and Peer Mappings*, Proc. 8th WebDB, Baltimore, June 2005

Data sources

- Physical data source (PDS)
  - Web data (DBLP), local data (files), ...
  - Splitted in logical data sources
- Logical data source (LDS)
  - Refers to one object type
  - Contains object instances
- Object instance
  - Refers to real world entity
  - Set of attributes
  - One attribute is id

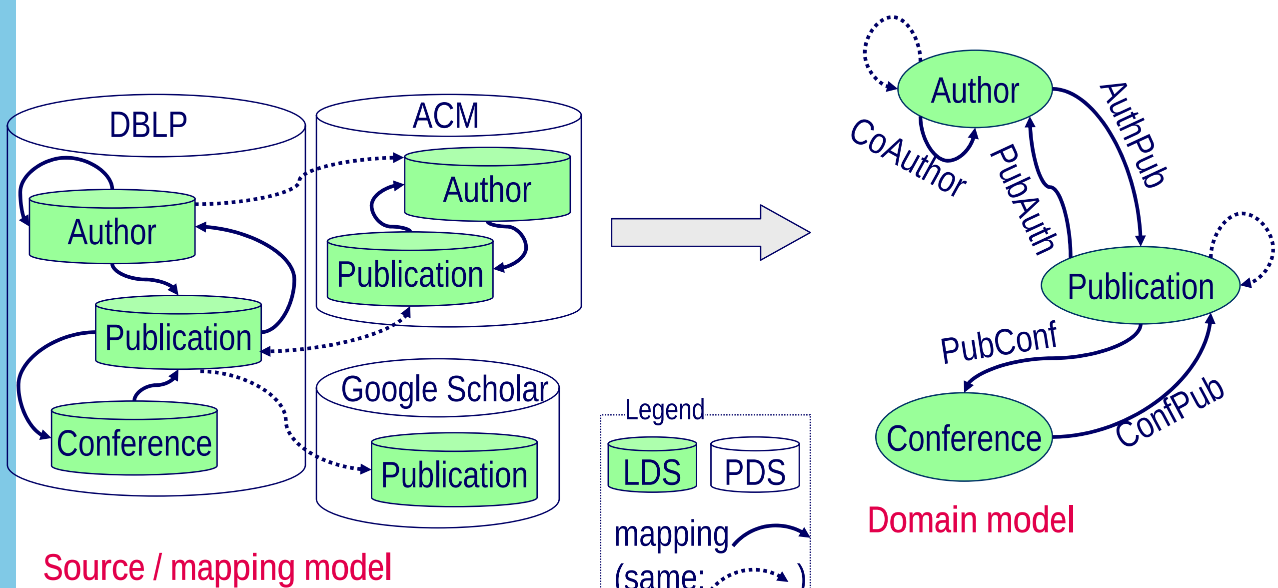


Mappings

- Directed binary relationship between LDS
- Mappings have a semantic mapping type
    - e.g., "publications of author“
- Kinds of mappings
    - Same mappings vs. association mappings
        - same = "equality" relationship typically between PDS
            - e.g., DBLP publication (id) → ACM publication (id)
    - Id mappings vs. query mappings
- Instance data: instance correspondences
    - Materialized: mapping tables
    - Determined on-the-fly: execution result (e.g., from web service or query)

Metadata model

- Used by mediator for mapping/operator execution
- Source/mapping model
- Domain model indicates available object types and relationships



Operators

- Query language capabilites + scripting support
- Set-oriented operators
  - Input: set of object or mapping instances
    + parameters / query specification
  - Output: set of object / mapping instances

$\Rightarrow$ Can be combined within scripts

Operators overview

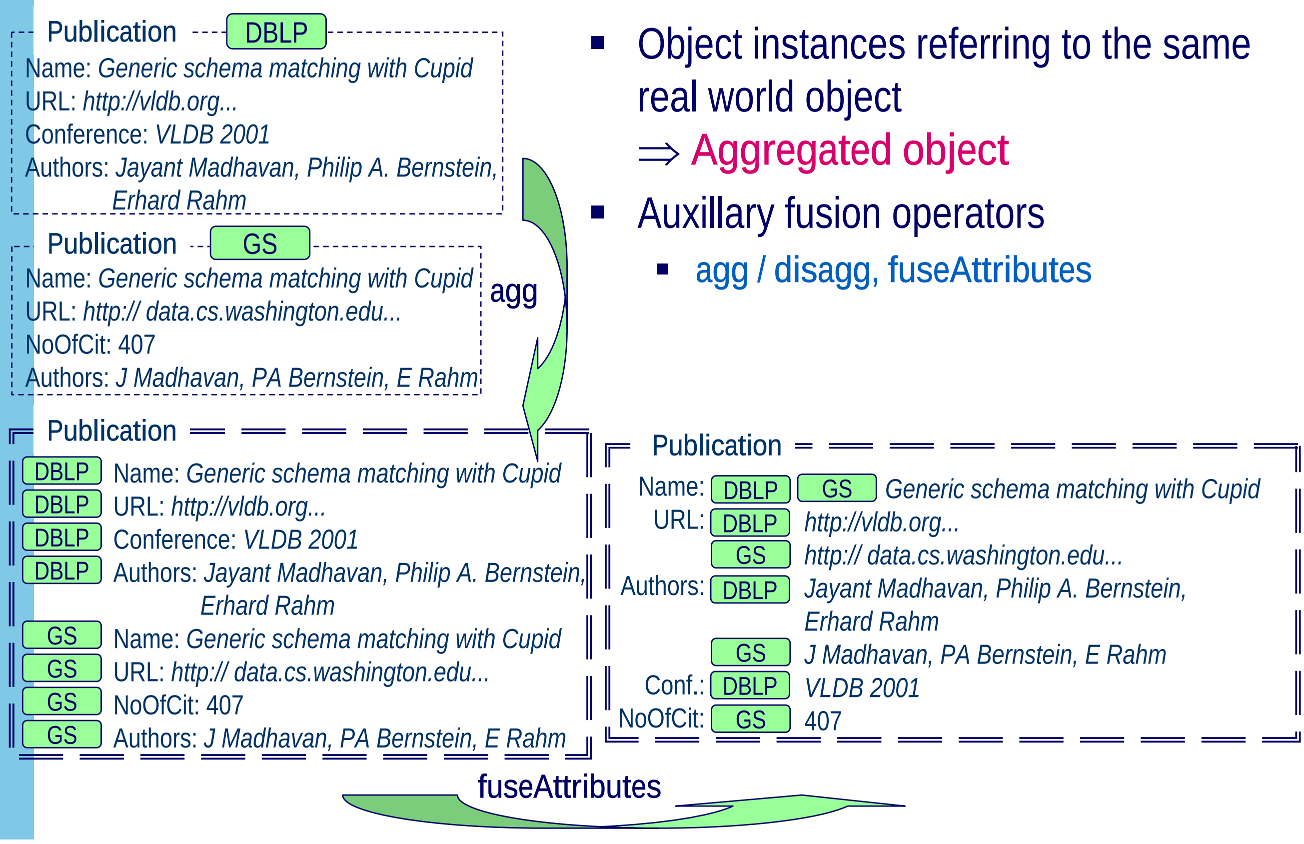
- Object instances (OI)
  - OI $\rightarrow$ OI: getInstances, traverse, traverseSame
  - Query $\rightarrow$ OI: queryInstances, queryMatch, attrTransf
- Aggregated objects (AO)
  - OI $\rightarrow$ AO: agg, disagg, fuseAttributes
  - AO $\rightarrow$ AO: aggregateSame, aggregateTraverse, aggregateMap
- Generic
  - union, diff, intersect
  - domain, range, compose

Operators for object instances

- queryInstances executes a query on a peer
  - *$S := queryInstances (Conf@DBLP, Series="SIGMOD")*
    returns all SIGMOD conferences from DBLP
- map executes a mapping
  - *map ($S, DBLP.ConfPubs)*
    returns all tuples (conference, publication)
- traverse returns the range of a mapping
  - *$P := traverse ($S, DBLP.ConfPubs)*
    returns all publications
- traverseSame "navigates" to corresponding objects of another physical source
  - *traverseSame ($P, GoogleScholar)*
    returns "equal" publications at GoogleScholar

Instance fusion

- Object instances referring to the same real world object
  ⇒ Aggregated object
- Auxillary fusion operators
  - agg / disagg, fuseAttributes

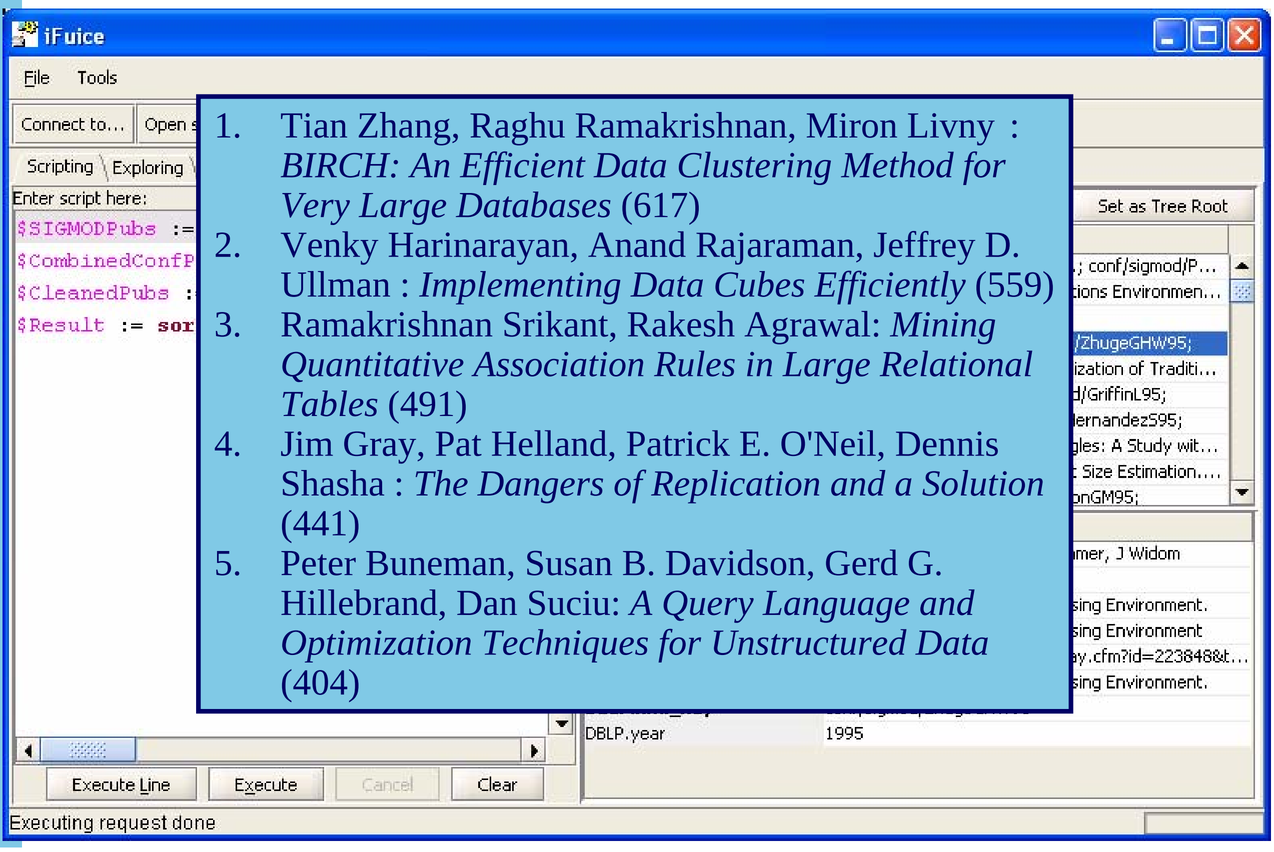


iFuice scripts

- Batch execution of operators
  - Store (intermediate) results in variables
- Scripts can be used by other scripts -> workflows
- Script example: SIGMOD test of time award

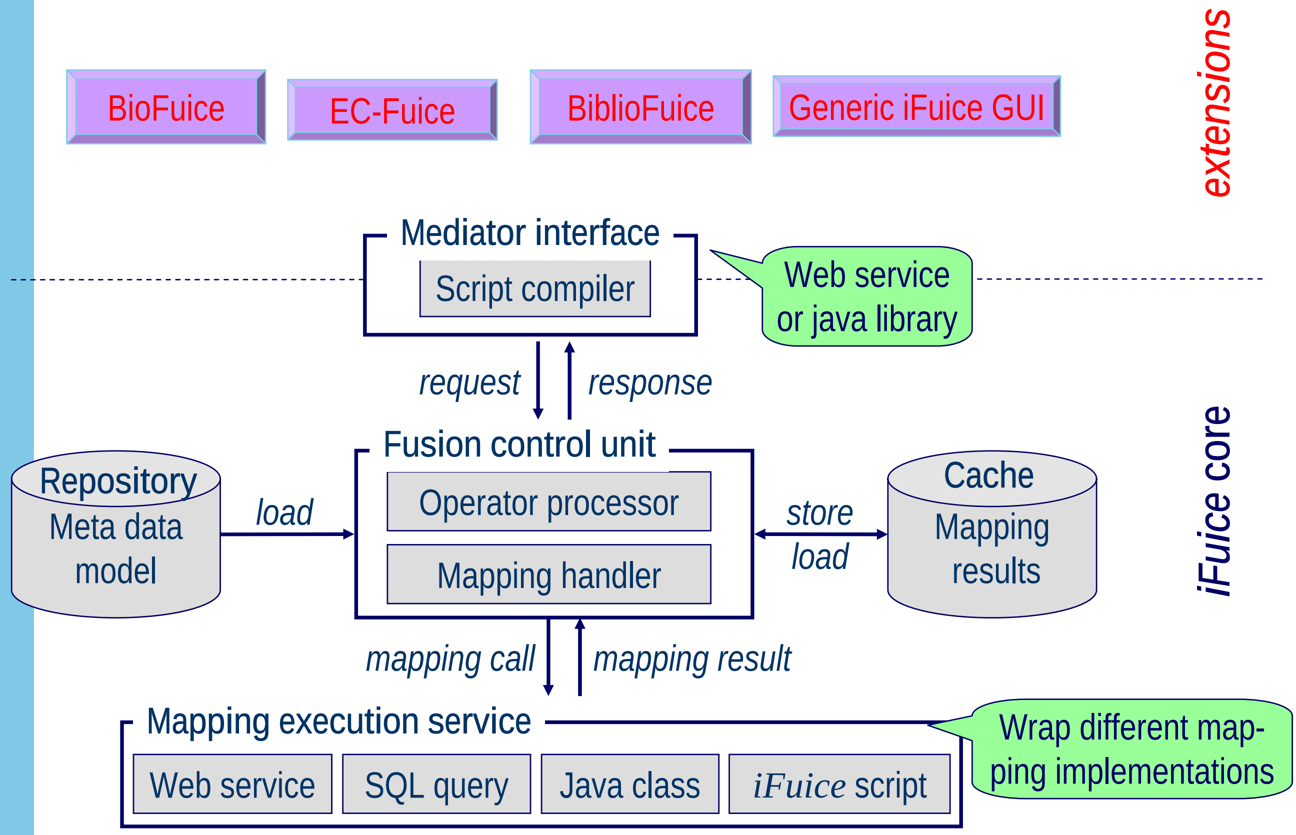
$*SIGMODPubs* := **queryTraverse** (LDS=DBLP.Conf, {Name="SIGMOD 1996"}, DBLPConfPubs)
$CombinedConfPub := **aggregateSame** (*$SIGMODPubs*, GoogleScholar)
$CleanedPubs := **fuseAttributes** (*$CombinedConfPub*)
$Result := **sort** ($*CleanedPubs*, "NoOfCitings")

Example: SIGMOD test of time award



1. Tian Zhang, Raghu Ramakrishnan, Miron Livny : *BIRCH: An Efficient Data Clustering Method for Very Large Databases* (617)
2. Venky Harinarayan, Anand Rajaraman, Jeffrey D. Ullman : *Implementing Data Cubes Efficiently* (559)
3. Ramakrishnan Srikant, Rakesh Agrawal: *Mining Quantitative Association Rules in Large Relational Tables* (491)
4. Jim Gray, Pat Helland, Patrick E. O'Neil, Dennis Shasha : *The Dangers of Replication and a Solution* (441)
5. Peter Buneman, Susan B. Davidson, Gerd G. Hillebrand, Dan Suciu: *A Query Language and Optimization Techniques for Unstructured Data* (404)

iFuice Architecture



Integration of bibliographic data*

- Citation analysis of scientific database publications
  - 10 years: 1994 – 2003
  - 2 conference series (SIGMOD, VLDB), 3 journals (TODS, VLDBJ, Sigmod Record)

dblp.uni-trier.de
COMPUTER SCIENCE BIBLIOGRAPHY
UNIVERSITÄT TRIER

- manually curated
- good data quality
- no citation counts

Google Scholar BETA

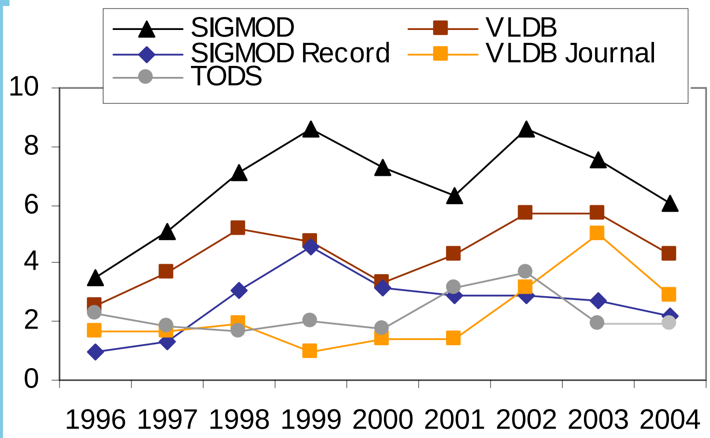
- automatic extraction of bibliographic data + citations from fulltext documents (PDF, PS files)
- data quality problems (duplicates, …)

PORTAL
HeBIS - Consortium

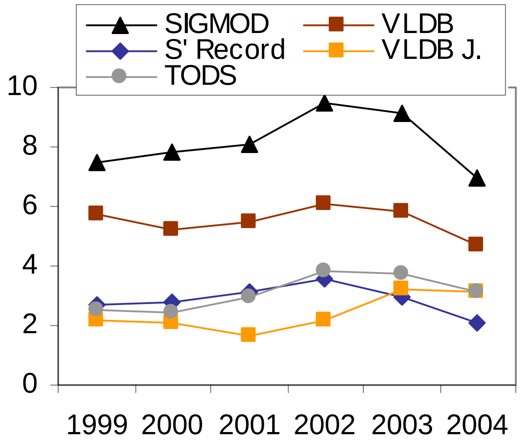
- ACM Digital Library
- includes
  - citations
  - author addresses / institutions

* Rahm, E., A. Thor: *Citation analysis of database publications*, ACM Sigmod Record, Dec. 2005

Integration Result: Impact Factors



2 years impact factors



5 years impact factors

- Journal impact factor JIF(X) = *average number of times articles from the journal published in the past two years X-1 and X-2 have been cited in year X*
- Adopted for conferences; extended to 5 years
- SIGMOD > VLDB > Journals

Aggregated Citation Frequencies

	Country	# Cit.	in %	# Pub.
1.	USA	51783	72.7	599
2.	Germany	4445	6.2	74
3.	Canada	3342	4.7	38
4.	France	2255	3.2	31
5.	Italy	2079	2.9	25
6.	Israel	858	1.2	6
7.	Japan	753	1.1	8
8.	Switzerland	699	1.0	13
9.	Denmark	655	0.9	8
10.	Greece	623	0.9	14

Table 5: Citations by country

	Institution	# Cit.	# Pub.
1.	IBM	9593	73
2.	Stanford University	7064	63
3.	University of Wisconsin-Madison	5150	61
4.	Bell Labs & AT&T Labs	4573	59
5.	University of Maryland	3299	34
6.	Microsoft	2411	27
7.	University of California, Berkely	1925	25
8.	INRIA (France)	1887	22
9.	University of Washington	1506	16
10.	University of Munich (Germany)	1367	15

Table 6: Citations by institution

- based on institution of first author
- only papers with at least 20 citations (w/o self-citations) are considered

MOMA Overview

- MOMA = Mapping based Object Matching
- Object consolidation framework
  - Peer-to-Peer environment with heterogeneous sources
  - Generation of instance mappings (correspondences)
  - Special case: duplicate detection within 1 source (generation of self-mapping)
- Key features
  - Utilization of existing mappings (reuse)
  - Extensible matcher library
  - Mapping combination
  - Construction of match workflows
- Note: similar objectives than in schema matching, e.g. in COMA / COMA++

Object Matching

- Goal: create „good“ same-mapping between two sets of input objects $A \subseteq LDS_A$ and $B \subseteq LDS_B$ of the same object type
    - e.g. between subsets of Publications@DBLP and Publications@GoogleScholar
- Mappings are represented by sets of instance correspondences
  - applicable for same-mappings (match results) and association mappings
  - inverting is trivial

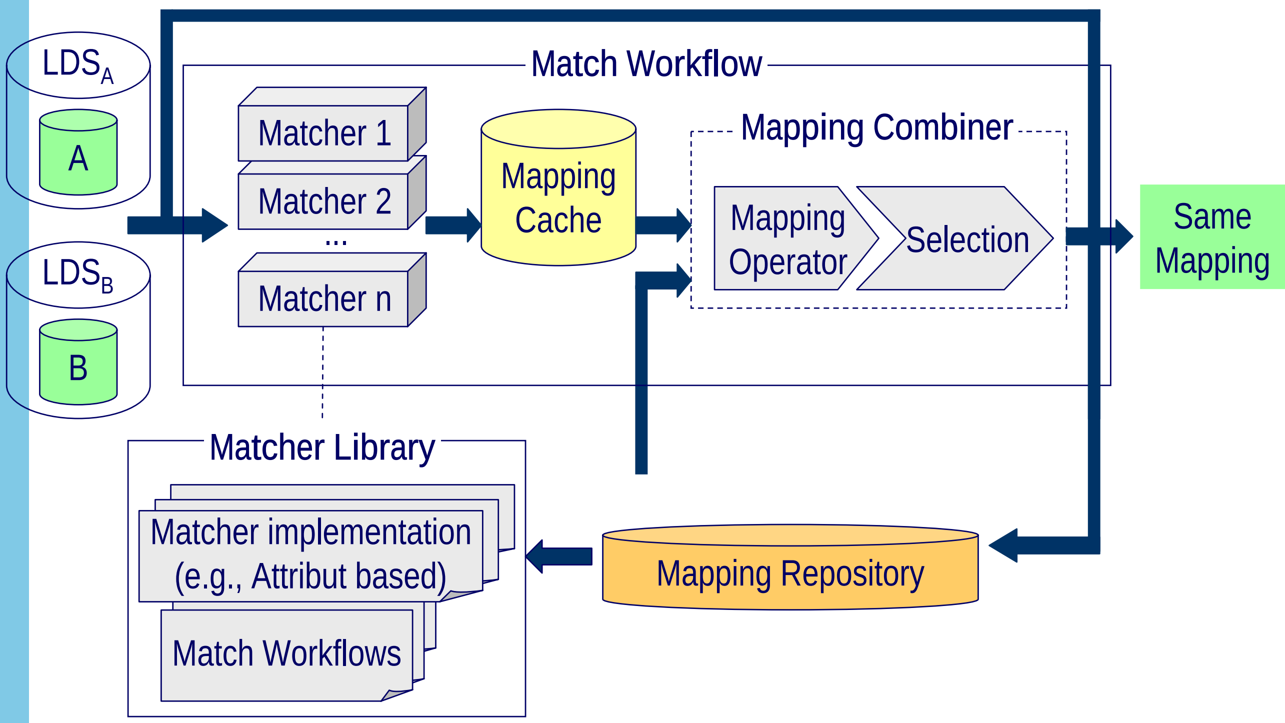
LDS_A	$LDS_{A'}$	Sim
a_1	a'_1	1
a_2	a'_1	0.9
a_3	a'_3	0.8

same-mapping for authors

LDS_P	LDS_A	Sim
p_1	a_1	1
p_1	a_2	1
p_2	a_1	1

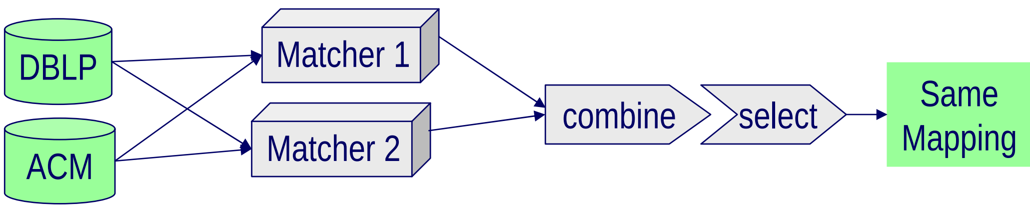
association mapping for publications and authors

Architecture



Match Workflows

- Coordinated execution of matchers and combination of mappings
  - single-attribute matcher (e.g. based on specific string similarity function)
  - multi-attribute matcher (hybrid matcher)
  - context matcher ...
- Example: Independent matcher execution



- Implemented as *iFuice* scripts

$M1 := **attrMatch** (*$DBLP*, *$ACM*, „[title]“, TFIDF, 0.9);

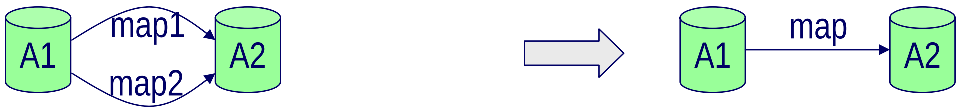
$M2 := **attrMatch** (*$DBLP*, *$ACM*, „[year]“, EditDistance, 0.7);

$Union := **union** (*$M1*, *$M2*, *avg*);

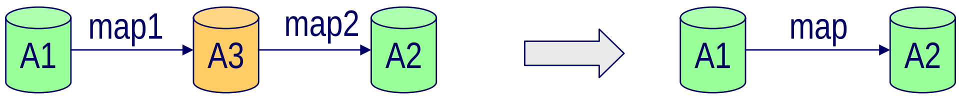
$Result := **select** (*$Union*, 0.8);

Match Strategies

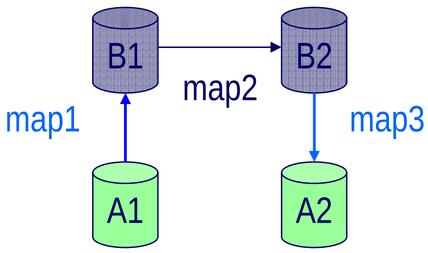
- Merge same-mappings

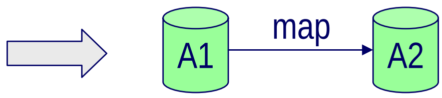


- Compose same-mappings



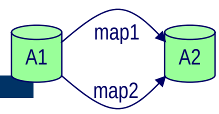
- Compose same- and association-mappings (*Neighborhood matcher*)





map1 and map3 are association-mappings
of the same semantic mapping type

Merging Same-Mappings



- Simple combination schemes: union, intersect, consolidate
  - *combination of similarity values: avg, min, max, weighted, left, right*
- Trade-off
  - Intersect: precision↗, recall ↘
  - Union: recall ↗, precision ↘
- Consolidate

  = Intersect + „non-conflicting“ correspondences
  - Correspondence (a, b) of map1 is non-conflicting, if map2 doesn‘t contain any correspondence for a or b
- Example

map1	
a1	b1
a2	b2
a4	b4

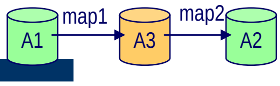
map2	
a1	b1
a3	b3
a4	b5

intersect	
a1	b1

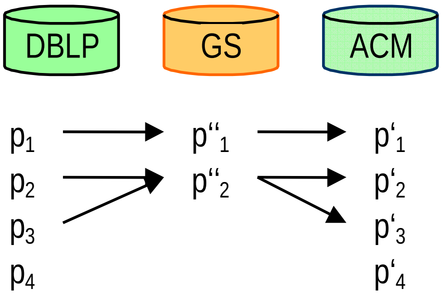
union	
a1	b1
a2	b2
a3	b3
a4	b4
a4	b5

consolidate	
a1	b1
a2	b2
a3	b3

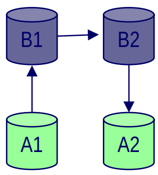
Composition of Same-Mappings



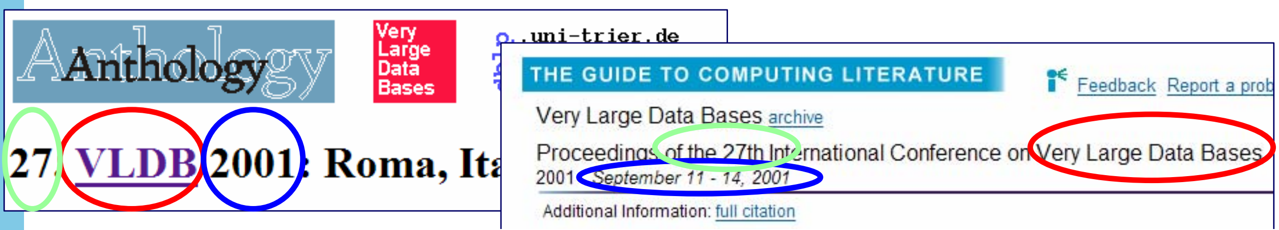
- Reuse same-mappings
- Example: Pubs@DBLP - Pubs@ACM
  - Map1: Pubs@DBLP - Pubs@GS (e.g., result of previous merge)
  - Map2: Pubs@GS - Pubs@ACM (e.g., existing mapping provided by GS)
- Compose (map1, map2, f)
  - (a1,a2) $\in$ composed mapping IF $\exists$ a3: (a1,a3) $\in$ map1 AND (a3,a2) $\in$ map2
  - similarity function f combines similarity values of compose paths leading to (a1,a2)
- Intermediate source should have good data quality / coverage



Neighborhood Matcher



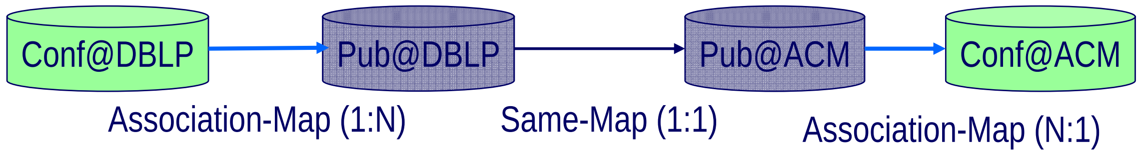
- Combine same- & association mappings
- Attribute matching may suffer from highly different value representations
- Use information stored in associated objects
- Bibliographic example: Conference@DBLP - Conference@ACM



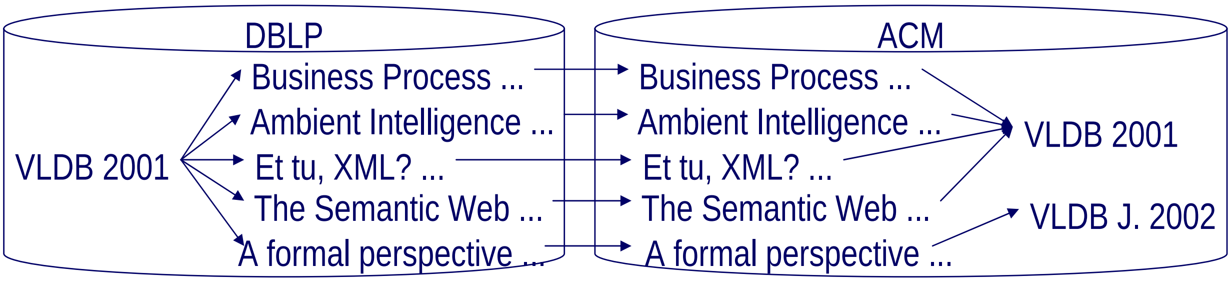
- Utilize Publications of conference
- „Two conferences are the same if they share a significant number of publications.“

Neighborhood Matcher (2)

- Compose 3 mappings



  - reuse *pub. same-mapping* to determine *conf. same-mapping*

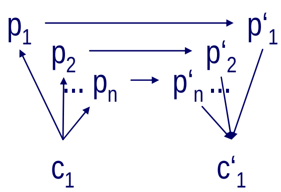


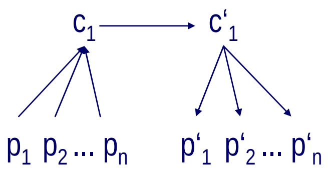
	DBLP		ACM	Similarity Value
compose =	VLDB 2001	⟶	VLDB 2001	0.8 (4 out of 5 paths)
	VLDB 2001	⟶	VLDB J. 2002	0.2 (1 out of 5 paths)

Mapping Cardinality

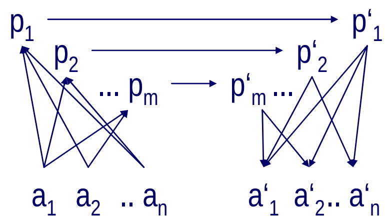
- Semantic cardinality of mappings determines quality of resulting mapping

a) 1:N (1:1) Conf-Publication





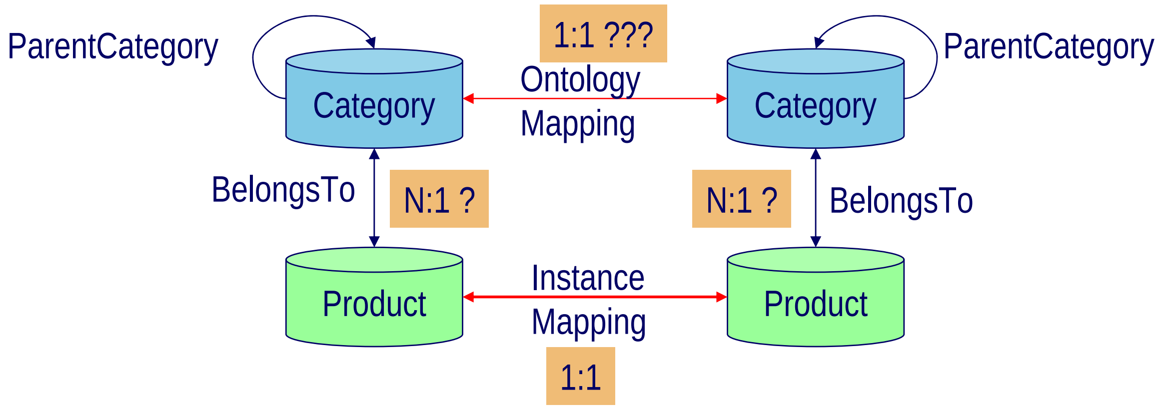
b) N:1 Publication-Conf



c) N:M Author-Publication

- N:1, N:M cases still useful for combination with other mappings, e.g. attribute match results

Use of Ontology Matching



- Mapping between product catalogs (ontology matching) can be used to match products (object matchings)
  - matching products should have matching product category
- Matching products can be used for ontology matching, i.e. to determine matching product categories

Experiment (EC-Fuice)

- 3 E-shops (Amazon.de, EBay.de, Softunity)
- software / video / games products
  - about 2000 categories, 40.000+ products on Amazon.de
- perfect instance-level (product) matching possible using unambiguous EAN (barcode)
  - products are often listed under several categories (e.g. action games and Xbox games)
  - i.e. category-product association mapping is N:M
- category (ontology) “same-mapping” is N:M
  - products of one category in shop 1 mostly map to several categories in shop 2 and vice versa
  - schema matching tools (and humans) have problems to find “correct” mapping

Evaluation

- Real data sources: DBLP, ACM, GoogleScholar (GS)
- DBLP:
  - 2616 publications, 3319 authors
  - 130 venues from 10 years: 20 conferences (Sigmod, VLDB), 110 journal issues (TODS, VLDBJ, Sigmod Record)
- Match problems
  - publications: DBLP-ACM, DBLP-GS, GS-ACM
  - authors: DBLP-ACM
  - venues: DBLP-ACM
- Perfect mapping: manually determined

Tuning

- Flexibility has its price
- Finding a good configuration of matchers and combination strategies is difficult
- Many possibilities for tuning:
  - Single-attribute matcher
    - Choice of similarity function
    - Threshold
  - Multi-attribute (hybrid) matcher
    - Choice of attributes
    - Choice of similiarity functions
  - Combination of independent matchers
    - Choice of combination strategy
    - Threshold

Related Work

- Surveys
  - Rahm, Do: *Data Cleaning: Problems and Current Approaches*. IEEE Techn. Bull. Data Eng., 2000
  - Gu, Baxter, Vickers, Rainsford. *Record Linkage: Current Practice and Future Directions*. Technical Report, 2003
- Frameworks (new operators for data cleaning, user-controlled workflows)
  - AJAX (Galhardas et al., VLDB 2001)
  - Potter's Wheel (Raman et al., VLDB 2001)
  - TAILOR (Elfeky et al., Data Eng. 2002)
- Tools
  - DataCleanser (EDD), Merge/Purge Library (Sagent/QM Software), MasterMerge (Pitnew Bowes) ...
  - MS SQL Server 2005: Data Cleaning Operators (Fuzzy Join / Lookup)

Related Work (2)

- Attribute Similarity
  - String distance metrics; Edit Distance, Jaro-Winkler, TFIDF, SoftTFIDF, ...
  - Comparison → Cohen (KDD03-Workshop on Data Cleaning ...)
  - Learnable string distance metrics: Bilenko et al. (KDD, 2003)
- Manually specified combination of matchers
  - Rules : Hernandez et al. (SIGMOD 1995)
  - Constraints: Shen et al. (AAAI 2005) („age 2 cannot match with salary 200K")
- Adaptive combination of matchers
  - Active Atlas (Tejada et al., Information Systems 2001)
    - Combination of multiple similarity scores for object pairs
    - Interactive decision tree learning to identify most informative example for the user to classify next
  - Multiple profilers: Doan et.al. (IIWeb, 2003)
- Context-based object matchers
  - Co-authors: Bhattacharya, Getoor (DMKD 2004)
  - Warehouse hierarchies: Ananthakrishna et.al. (VDLB 2002)
  - XML hierarchies: Weis et al. (SIGMOD 2005)
  - XML graphs: Dong et al. (SIGMOD 2005)

Summary

- **iFuice**: generic way to mapping-based information fusion
  - utilizes existing instance correspondences
  - powerful operators and script facility to build data transformation and analysis workflows
  - succesful adoption in different domains
- Need for object matching in P2P data integration
  - generation of instance correspondences
- MOMA framework
  - combined use of several matchers
  - reuse of existing and previously determined instance mappings
  - need to combine similarity values in operators like compose, consolidate, ...
  - flexible match strategies, e.g. neighborhood matching
- Evaluation on challenging match tasks (web sources of different completeness, dirtyness, accessibility) showed effectiveness of MOMA
- Flexibility of MOMA framework provides opportunities for self tuning