COMA – A system for flexible combination of schema matching approaches

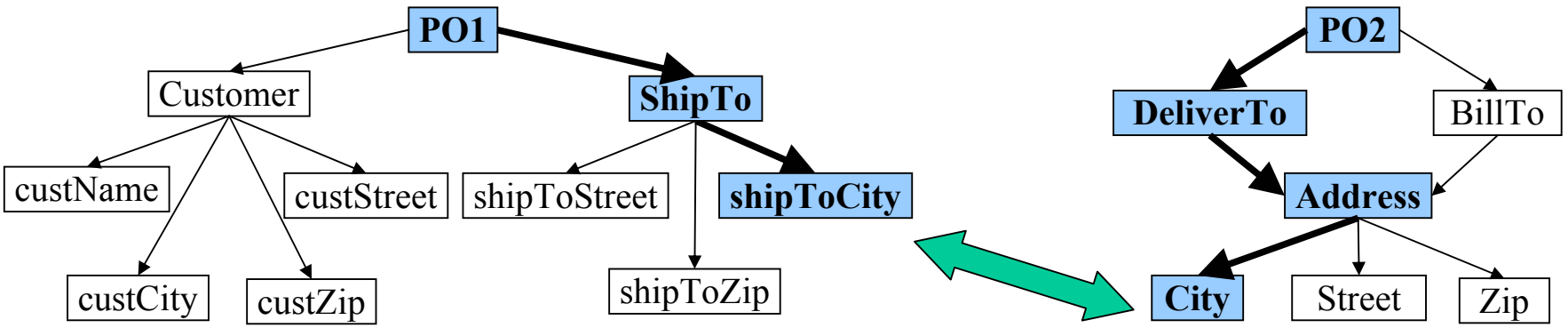
Hong-Hai Do, Erhard Rahm
University of Leipzig, Germany
dbs.uni-leipzig.de

Content

- Motivation
- The COMA approach
  - Comprehensive matcher library
  - Flexible combination scheme
  - Novel reuse-oriented match approach
- Evaluation setup and results
- Conclusions and future work

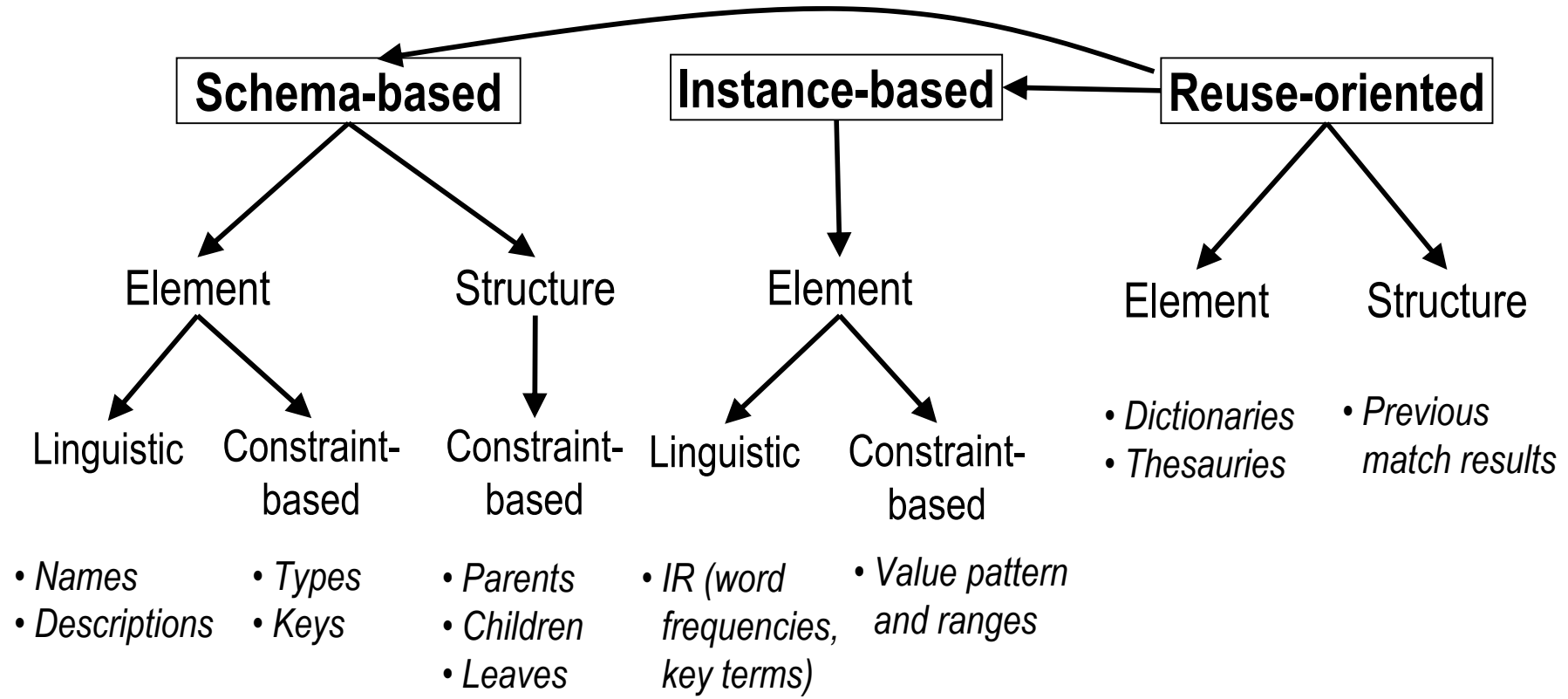
Motivation

- Schema matching: Finding semantic correspondences between two schemas
- Crucial step in many applications
  - Data integration: mediators, data warehouses
  - E-Business: XML message mapping
  - ...
- Currently manual, time-consuming, tedious
  - Need for approaches to automate the task as much as possible



PO1.ShipTo.shipToCity ↔ PO2.DeliverTo.Address.City

Individual Match Approaches



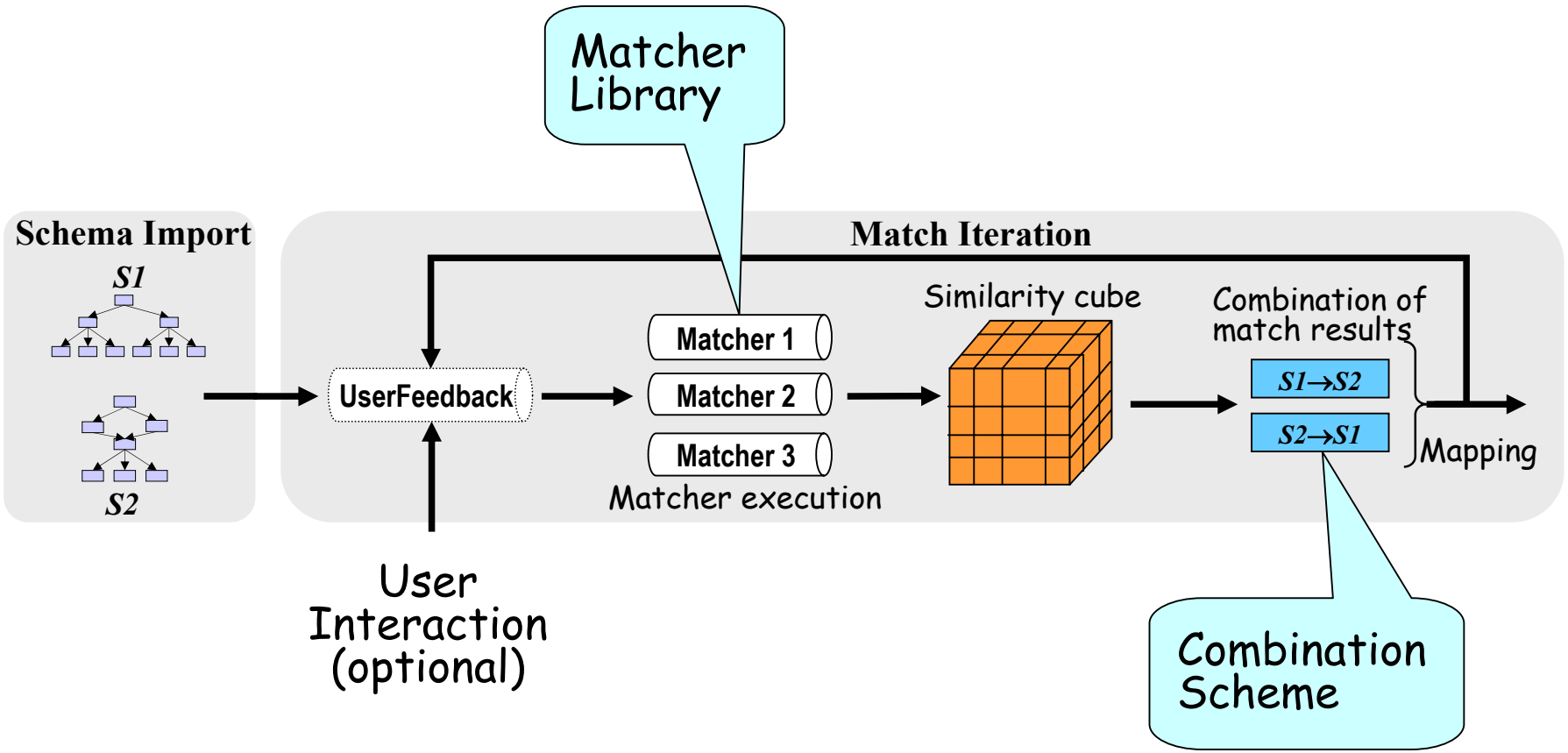
Survey paper [Rahm, Bernstein - VLDB Journal'01]

Combining Match Approaches

- Combination of match algorithms
  - *Hybrid*: fixed combination, difficult to extend and improve
    - currently most common: Cupid, SemInt, SimilarityFlooding, DIKE, MOMIS, TranScm
  - *Composite*: combination of the results of independently executed matchers
    - currently only for machine learning-based techniques: LSD, GLUE

- COMA: Framework for flexible ***CO***mbination of ***MA***tch algorithms
  - Extensible matcher library
  - Combination scheme with various combination strategies

System Architecture



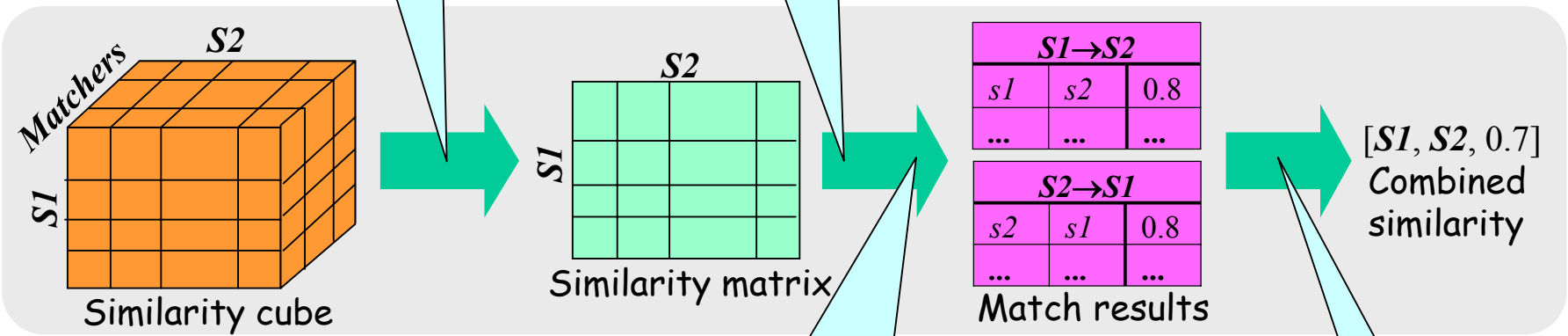
Combination Scheme

1. Aggregation of matcher-specific results

Average, Max, Min, Weighted

SmallLarge, LargeSmall, Both

2. Match direction



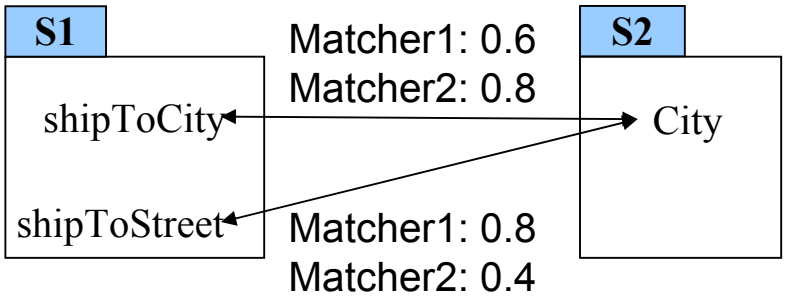
3. Selection of match candidates

MaxN (Max1), Threshold, MaxDelta, Threshold+MaxN, Threshold+MaxDelta

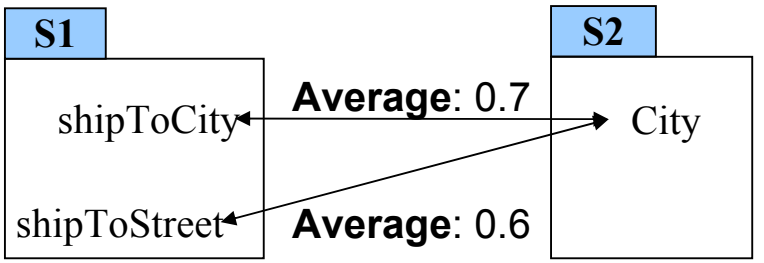
4. Computation of combined similarity

Dice, Average

Match Processing: Example



1. Aggregation



2. Direction

|S1|>|S2|

LargeSmall
(Match candidates
for smaller schema S2)

S2 elements	*S1 elements*	*Sim*
City	shipToCity	0.7
	shipToStreet	0.6

3. Selection

Max1

S2 elements	*S1 elements*	*Sim*
City	shipToCity	0.7

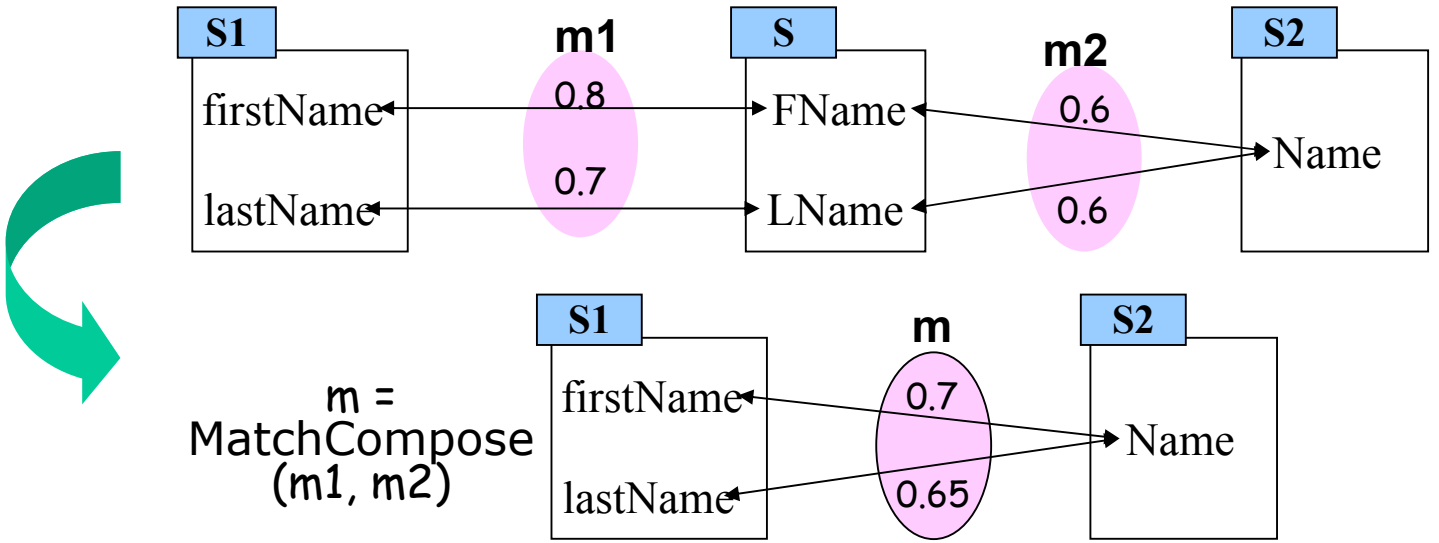
Threshold(0.5)

S2 elements	*S1 elements*	*Sim*
City	shipToCity	0.7
City	shipToStreet	0.6

Matcher Library

Type	*Matcher*	*Schema Info*	*Auxiliary Info*	*Constituent Matchers*
Simple	Affix	Element names	–	–
	n-gram	Element names	–	–
	Soundex	Element names	–	–
	EditDistance	Element names	–	–
	Synonym	Element names	External dictionaries	–
	DataType	Data types	Data type compatibility table	–
	UserFeedback	–	User-specified (mis-) matches	–
Hybrid	Name	Element names	–	Affix, 3-Gram, Synonym
	TypeName	Data Types+Names	–	DataType, Name
	NamePath	Names+Paths	–	Name
	Children	Child elements	–	TypeName
	Leaves	Leaf elements	–	TypeName
Reuse-oriented	Schema	–	Existing schema-level match results	–

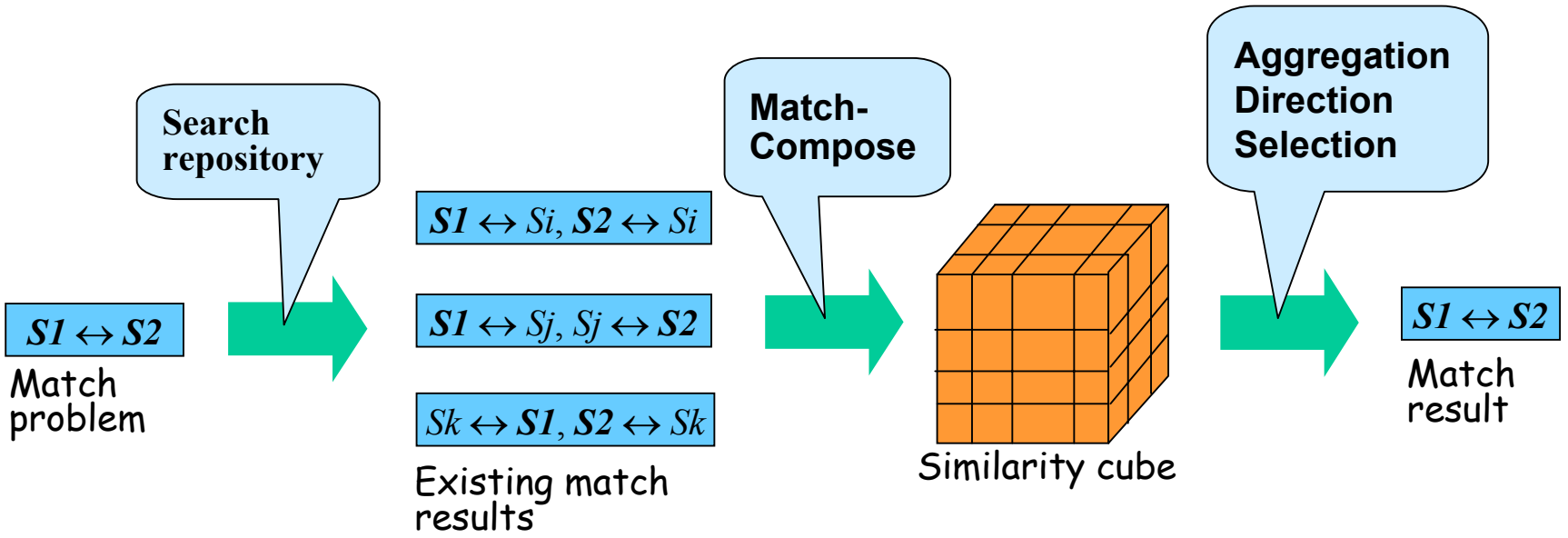
Reuse-oriented Matching



- The *MatchCompose* operation: Transitivity of element similarity
  - Composition of similarity relationships
- Reuse of multiple match correspondences
  - vs. reuse of single element-level correspondences from synonym tables, thesauries

Schema-level Reuse

- The Schema matcher:



  - Reuse complete match results at the schema level
  - Exploit all possible reuse opportunities
  - Limit negative effects of transitivity

Real-world Evaluation

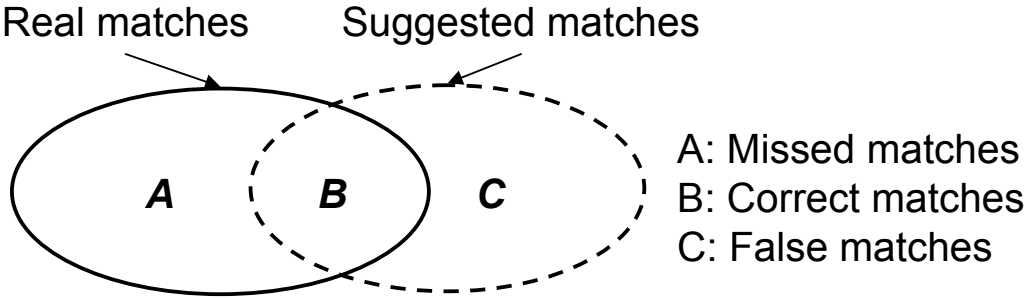
- 5 real-world schemas (XML – Purchase order), 10 match tasks
  - CIDX, Excel, Noris, Paragon, Apertum from biztalk.org
  - 40-145 elements

- Systematic evaluation (automatic mode)
  - **1** *Series* = **10** *Experiments*: Test of **1** configuration of (Matcher, Aggregation, Direction, Selection, Combined similarity) with **10** match tasks
  - 12,312 series = 123,120 experiments

Matchers		***Aggregation***	***Direction***	***Selection***	***Combined Sim***
No reuse	5 single		-LargeSmall -SmallLarge -Both	-MaxN(1-4) -Delta(0.01-0.1) -Threshold(0.3-1.0) -Threshold(0.5)+ MaxN(1-4) -Threshold(0.5)+ Delta(0.01-0.1)	-Average -Dice
	11 combinations	-Max -Average -Min			
Reuse	2 single				
	12 combinations	-Max -Average -Min			-Average
Σ =	16 + 14	3	3	36	2

Match Quality Measures

- Comparison of automatically with manually (i.e. real) derived match correspondences



- Quality measures:

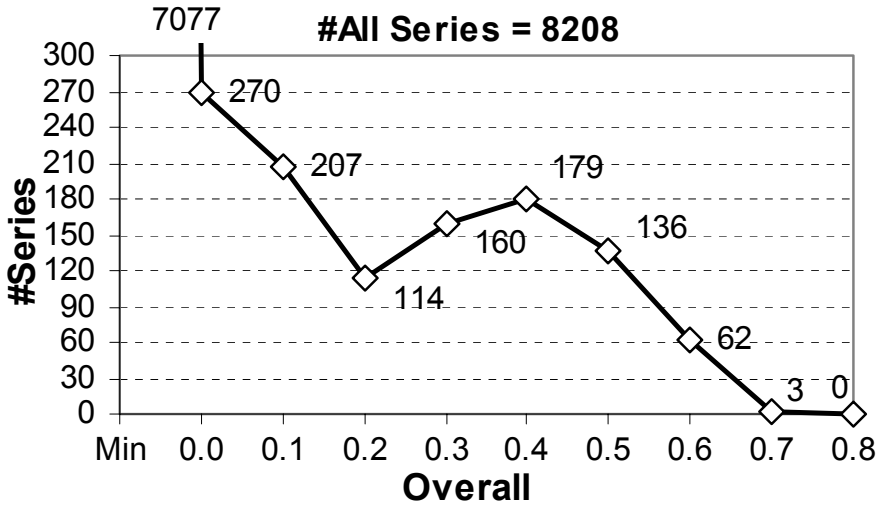
$$Precision = \frac{|B|}{|B|+|C|} \qquad Recall = \frac{|B|}{|A|+|B|}$$

SimilarityFlooding [ICDE02]:

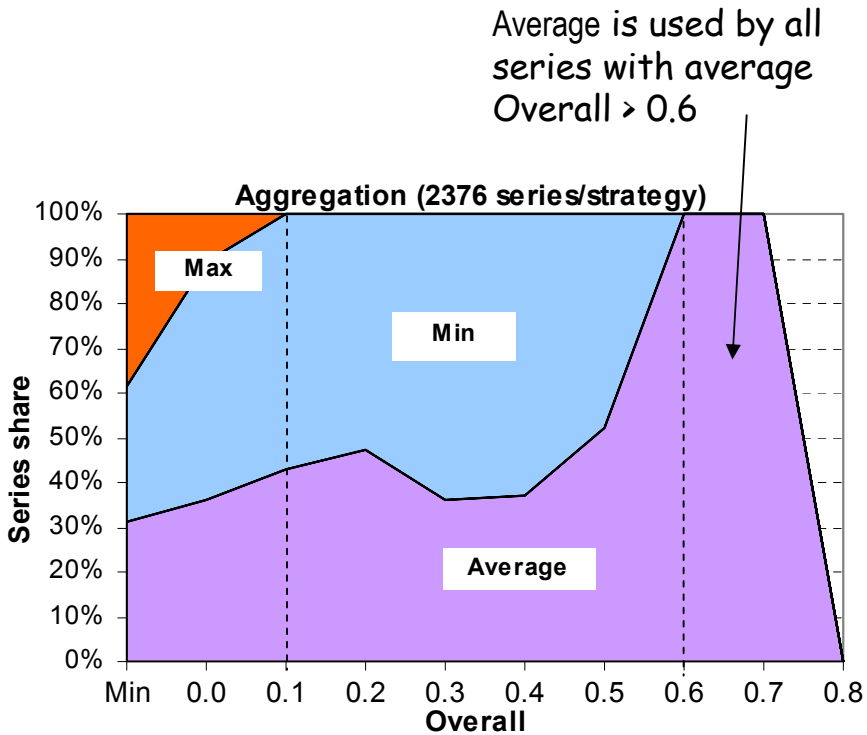
$$Overall = 1 - \frac{|A|+|C|}{|A|+|B|} = \frac{|B|-|C|}{|A|+|B|} = Recall * \left(2 - \frac{1}{Precision}\right)$$

  - *Overall*: post-match effort to add missed and to remove false matches; negative *Overall* → no gain
  - Computed for single experiments and averaged over 10 experiments for each series (average *Overall*, etc.)

Results: Combination Strategies (1)



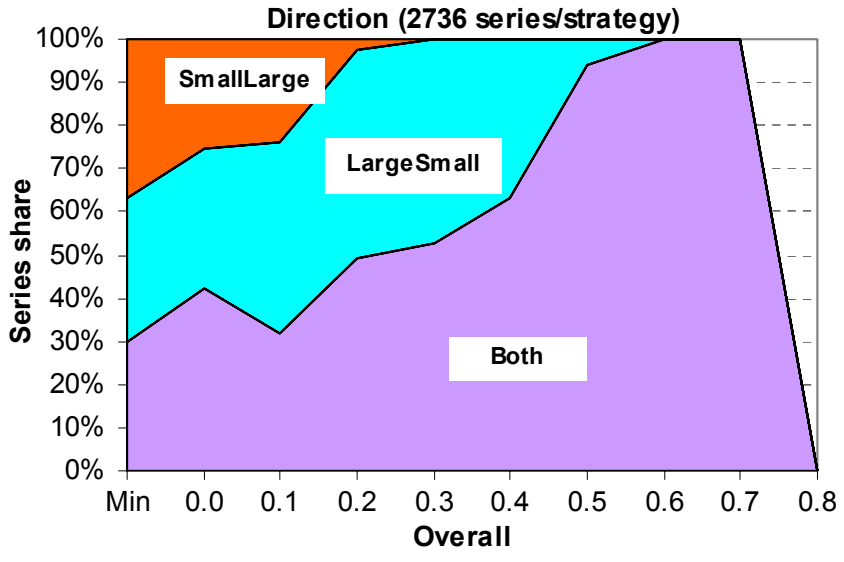
- Most no-reuse series have negative average *Overall*
- "Good" matcher/strategy:
  - Positive average *Overall*
  - High presence in higher *Overall* ranges

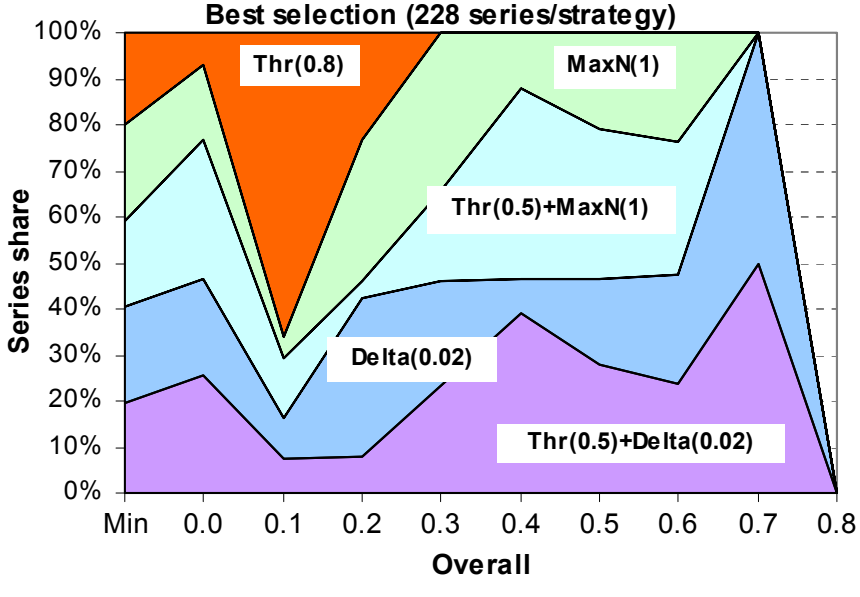


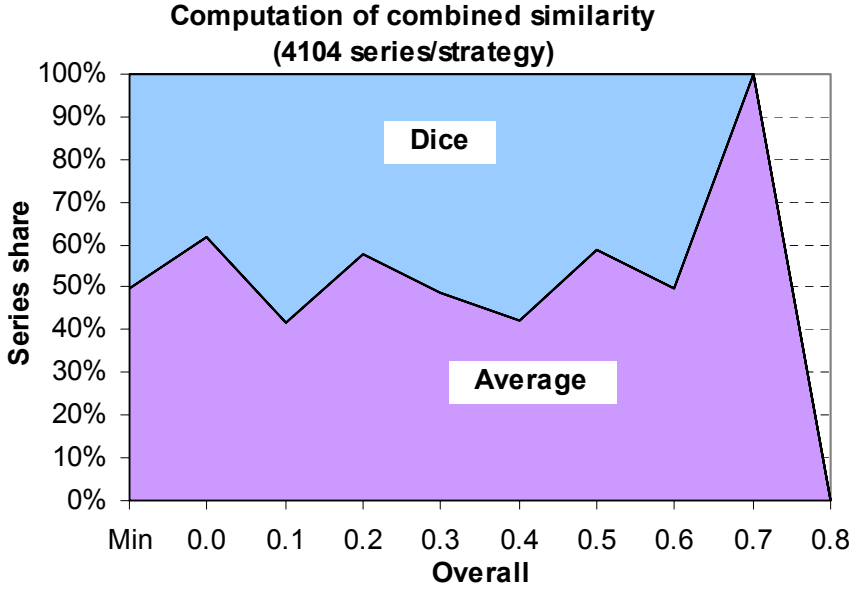
- *Aggregation*: Average (compensating)

Results: Combination Strategies (2)

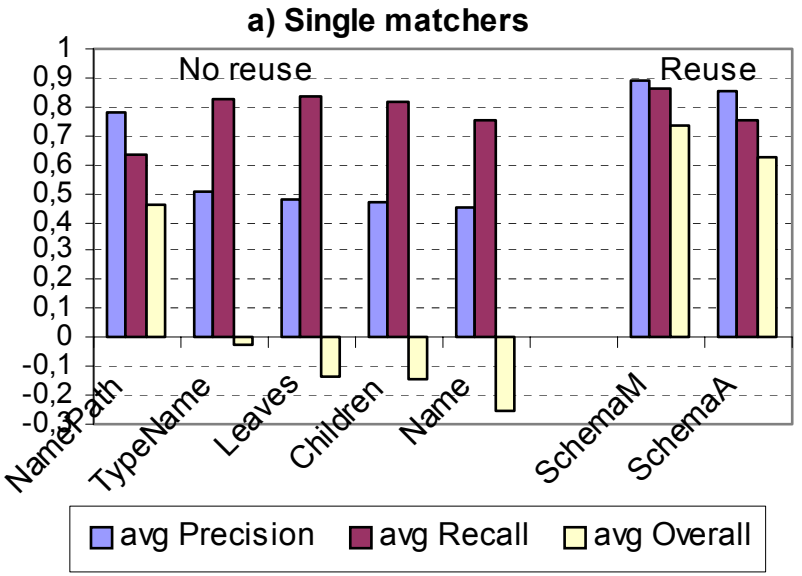
- *Direction*: Both (considering both directions)
- *Selection*: Threshold+Delta (above threshold + within tolerance)
- *Combined similarity*: Average (pessimistic)
- *Matcher*: All (combination of all hybrid matchers)







Results: Single Matchers

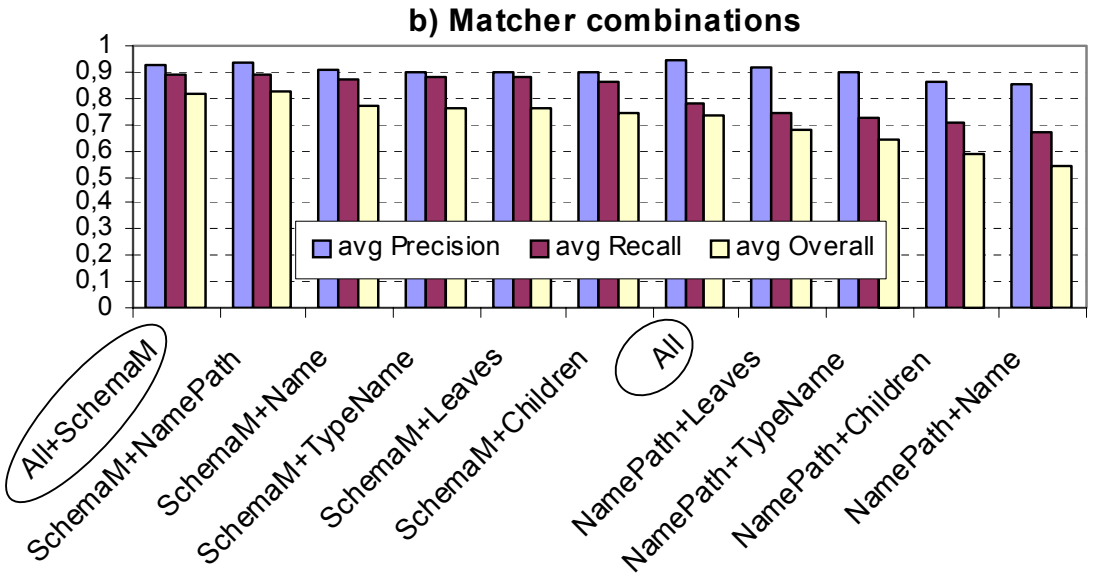


SchemaM: Schema with *manually* derived (real) match results

SchemaA: Schema with match results *automatically* derived using the default match operation

- Instability of some single (hybrid) matchers (negative *Overall*) because of shared elements
  - E.g. *DeliverTo.Address* and *BillTo.Address*
- Considering hierarchical names (NamePath) more accurate
- Schema-level reuse very effective:
  - Essential improvement over no-reuse hybrid matchers
  - Reusing approved match results better than automatically derived match results

Results: Combined Match Approaches

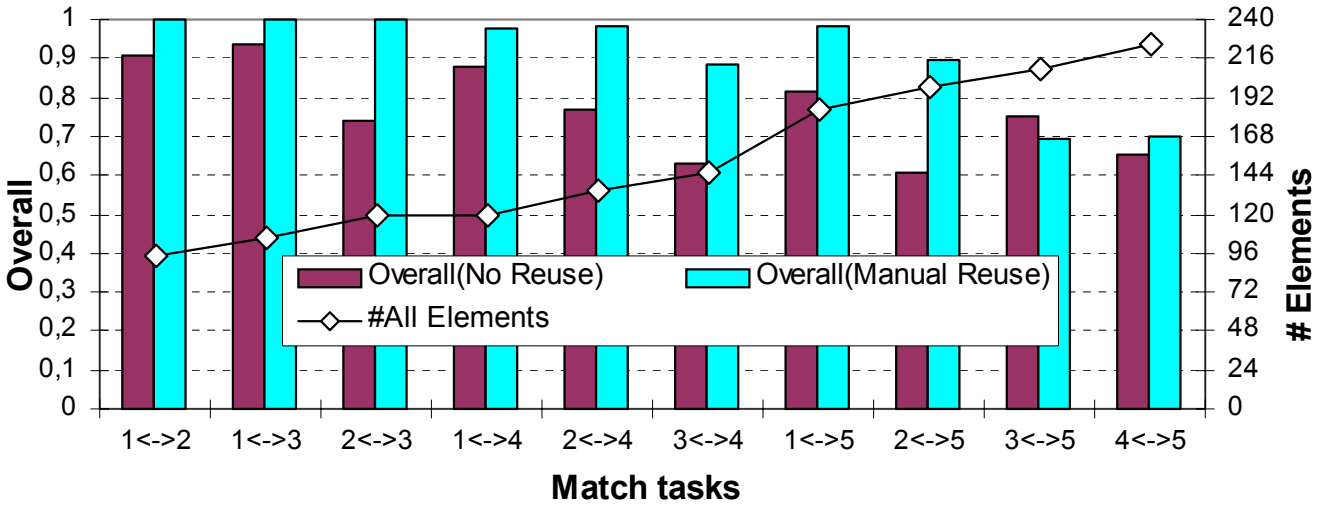


All: Combination of all no-reuse hybrid matchers

All+SchemaM: Combination of all no-reuse hybrid matchers and SchemaM

- Reuse matchers outperform no-reuse matchers
  - Best no-reuse All : 0.73 average *Overall* (*Precision* 0.95, *Recall* 0.78)
  - Best reuse All+SchemaM: 0.82 average *Overall* (*Precision* 0.93, *Recall* 0.89)
- Combinations outperform single hybrid matchers
  - Combined matchers, e.g. All, consider many aspects at the same time
  - NamePath+Leaves: effective scheme, considering paths to identify context of shared elements, and leaves to cope with structural conflicts

Results: Match Sensitivity



- Impact of schema characteristics :
  - Degrading match quality with increase of schema size
- Best combinations: no-reuse All and reuse-oriented All+Schema
  - High stability across different match tasks
  - Little tuning effort for the default match operation

Conclusions and Future Work

- The COMA framework
  - Extensible matcher library, including novel reuse approach
  - Powerful combination scheme for both specifying match operations and constructing new matchers from existing ones
- Comprehensive evaluation on real-world schemas
  - High effectiveness on large schemas
  - Reuse: essential improvement over no-reuse
  - Composite approach as THE solution for matcher combination
- Future work
  - Matchers: more powerful reuse strategies, instance-based matchers
  - More intelligent combination strategies
  - Application to more real-world scenarios, esp. in bioinformatics