wdilab

600 JAHRE
UNIVERSITÄT LEIPZIG

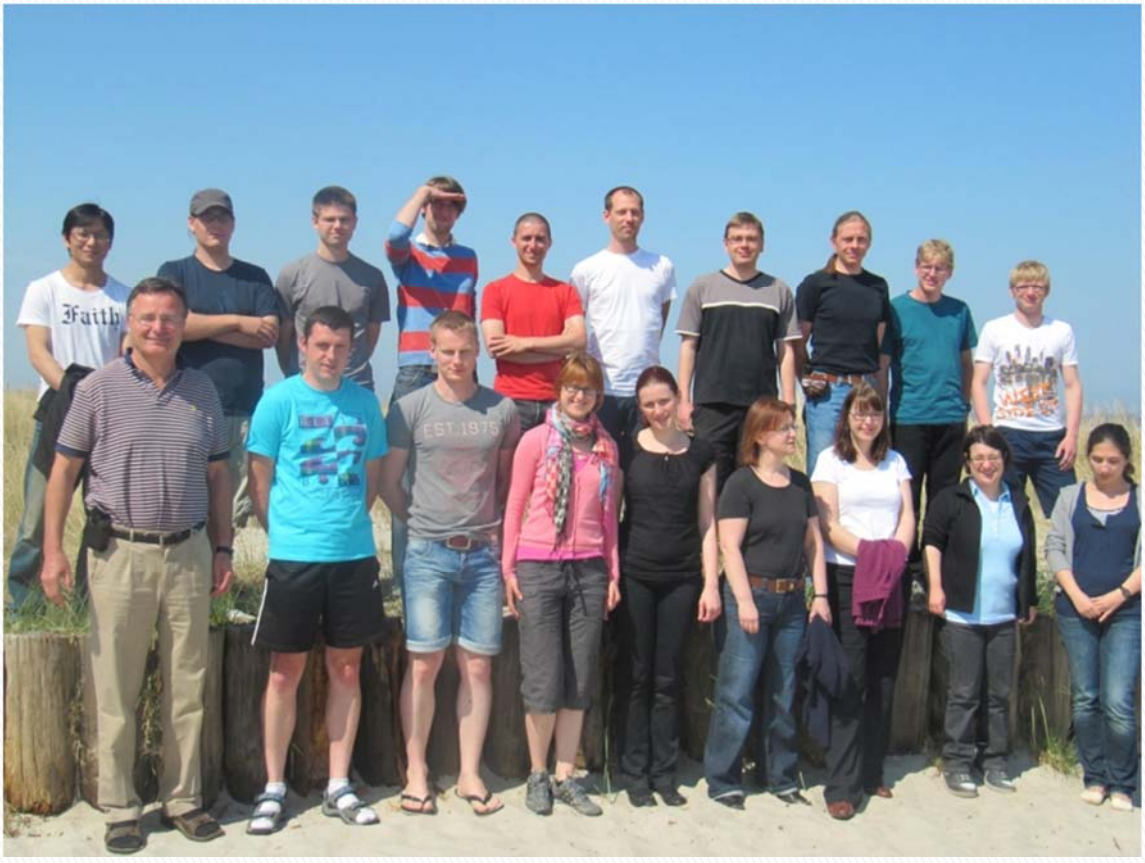
Evolution and merging of real-life ontologies

Erhard Rahm

http://dbs.uni-leipzig.de
http://wdilab.uni-leipzig.de

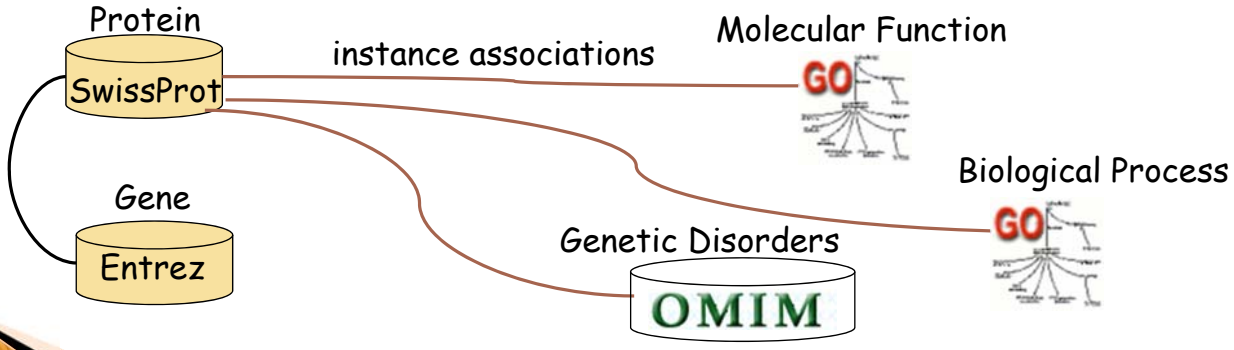
SEBD 2011, Maratea, Italy
June 28, 2011

Database Group 2011



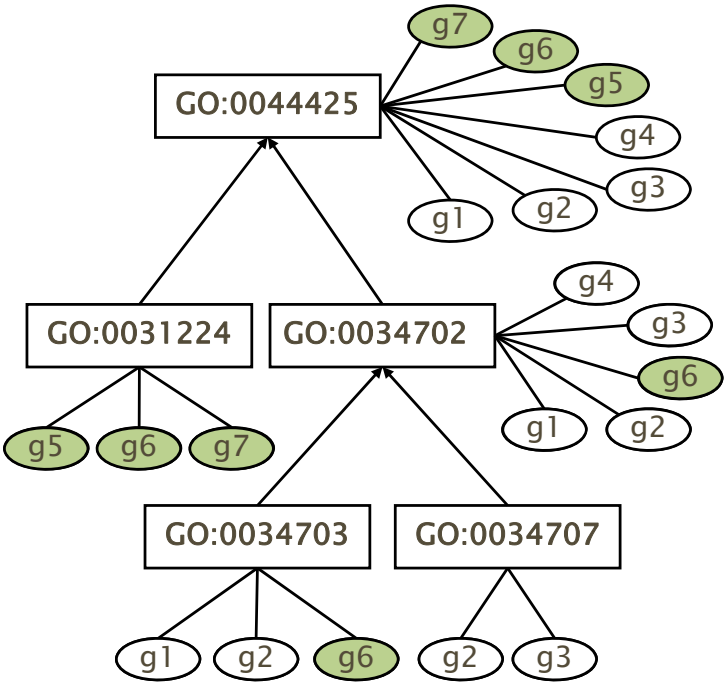
Life Science Ontologies

- Many ontologies for different areas, e.g.
  - Molecular Biology, Anatomy, Diseases etc.
- Large ontologies, e.g., Gene Ontology (GO), NCI Thesaurus with ten thousands of concepts
- Ontologies used to annotate genes and proteins
  - ✓Support for "functional" data analysis / term enrichment
- Instances: annotated objects; separate from ontology



Example Analysis: Term Enrichment

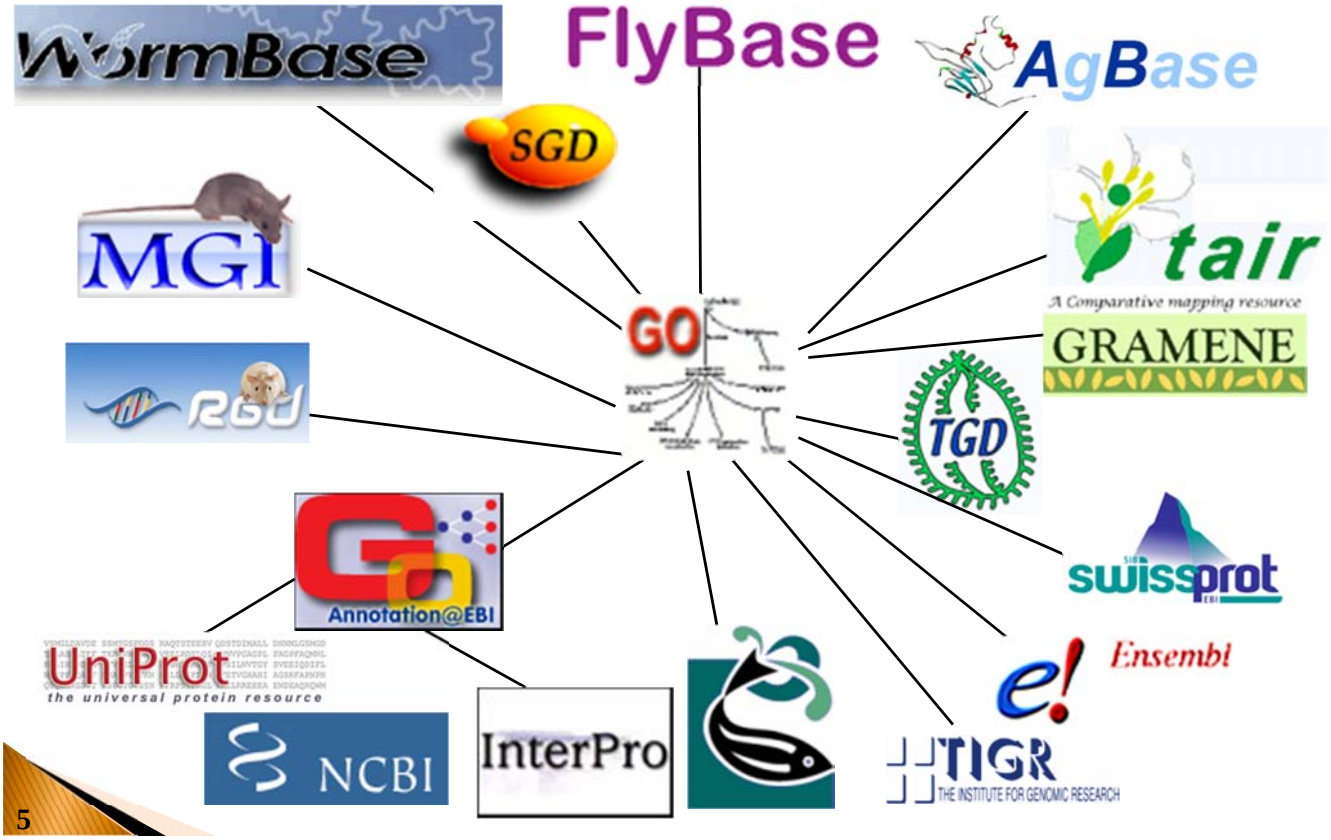
- Find significantly enriched (overrepresented) categories in a specific gene set compared to a background set (all considered genes)
- Usage of annotation propagation



g non interesting gene

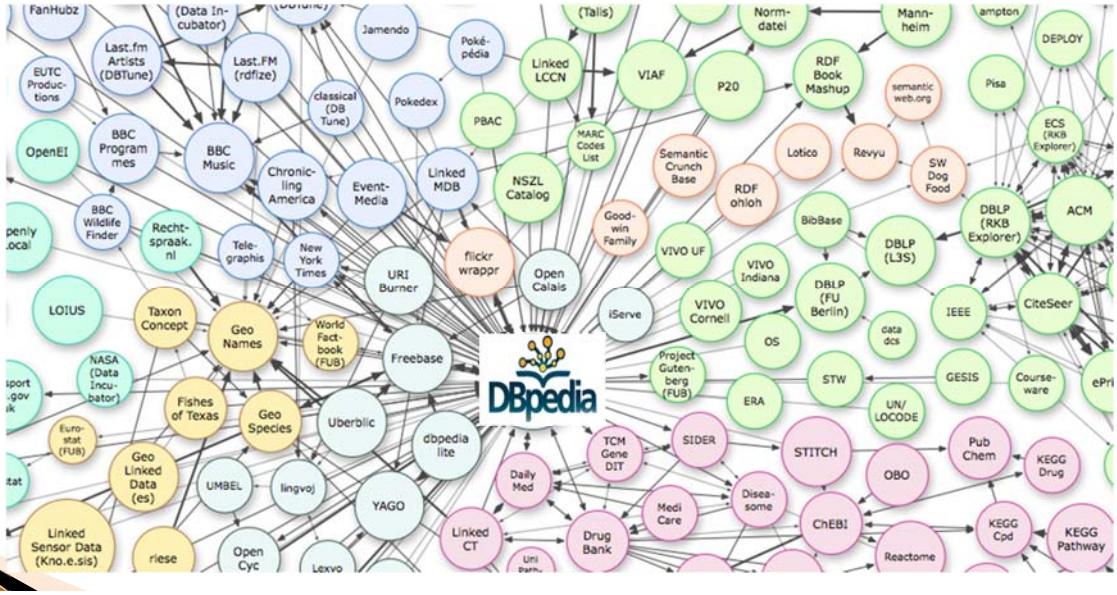
g interesting gene, e.g. "differentially expressed"

Example: Widespread Usage of GO



Linked Open Data

- Many linked data sources containing ontologies and associated instances
- Links (mappings) between ontology categories and instances helpful for data integration



Product Catalogs

- Categorization of products
- Instances: product descriptions
- Often very large: ten thousands categories, millions of products

Web Directories

- Categorization of websites
- Instances: website descriptions (URL, name, content description)
- Manual vs. automated category assignment of instances

- General lists or specialized (per region, topic, etc.), e.g.

YAHOO! DIRECTORY

dmoz open directory project

Google Directory

BUSINESS.COM

VFUNK GLOBAL DANCE MUSIC DIRECTORY

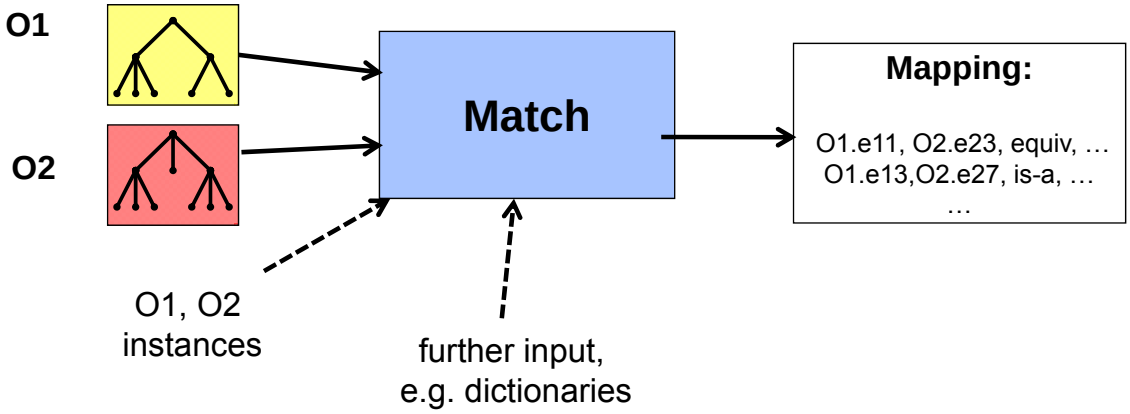
Ontologies: Usage Forms

- Support a shared understanding of terms/concepts in a domain
  - Annotation of data instances by terms/concepts of an ontology
- Semantically organize information of a domain
  - Find data instances based on concepts (queries, navigation)
- Support data integration
  - Mapping data sources to shared ontology
  - Mappings between related ontologies
- Support for link discovery and entity resolution
  - Search space reduction, context information

Technical Issues

- Ontology creation / learning
- Ontology usage
  - annotation creation, query processing, etc.
- Ontology matching / alignment
- Ontology integration / merging
- Ontology evolution

Ontology Matching / Alignment



- Process of identifying semantic correspondences between input ontologies
  - Result: ontology mapping
  - Mostly equivalence mappings: correspondences specify equivalent ontology concepts
- Variation of schema matching problem

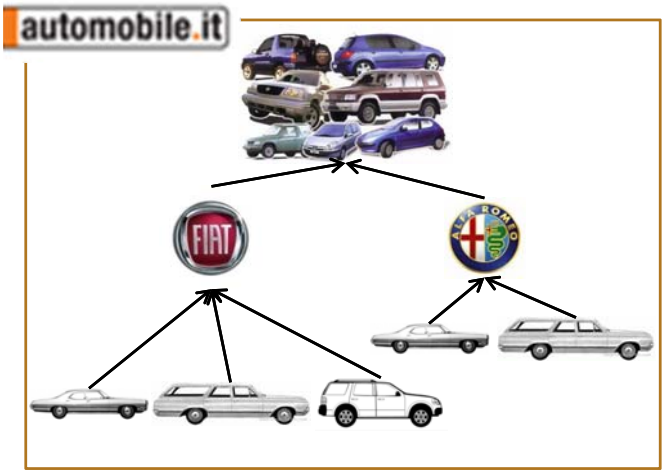
Match prototypes*

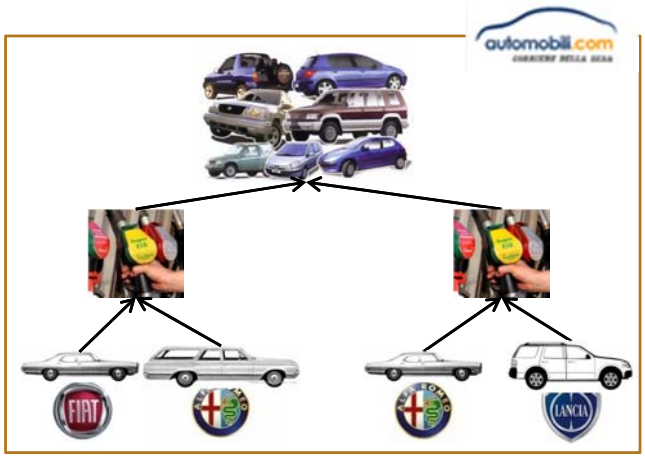
		Cupid	COMA++	Falcon	Rimom	Asmov	Agr.Maker	OII Harmony
year of introduction		2001	2002/2005	2006	2006	2007	2007	2008
Input	*relational*	√	√	-	-	-	-	√
schemas	*XML*	√	√	-	-	-	(√)	√
	ontologies	-	√	√	√	√	√	√
OAEI participation		-	√	√	√	√	√	-
compreh. GUI		-	√	(√)	?	?	√	√
Matchers	*linguistic*	√	√	√	√	√	√	√
	structure	√	√	√	√	√	√	√
	Instance	-	√	-	√	√	√	-
use of ext.dictionaries		√	√	?	√	√	√	√
schema partitioning		-	√	√	-	-	-	-
parallel matching		-	-	-	-	-	-	-
dyn. matcher selection		-	-	-	√	-	-	-
mapping reuse		-	√	-	-	-	-	-

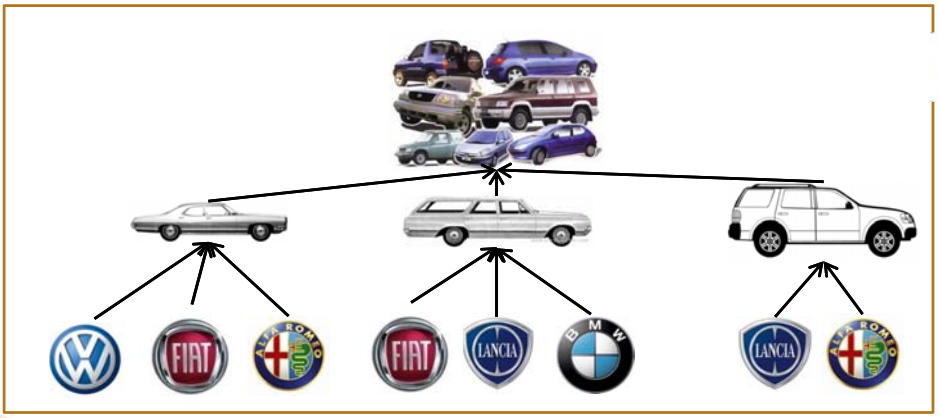
Rahm, E.: Towards large-scale schema and ontology matching. In: Schema Matching and Mapping, Springer-Verlag, 2011

Agenda

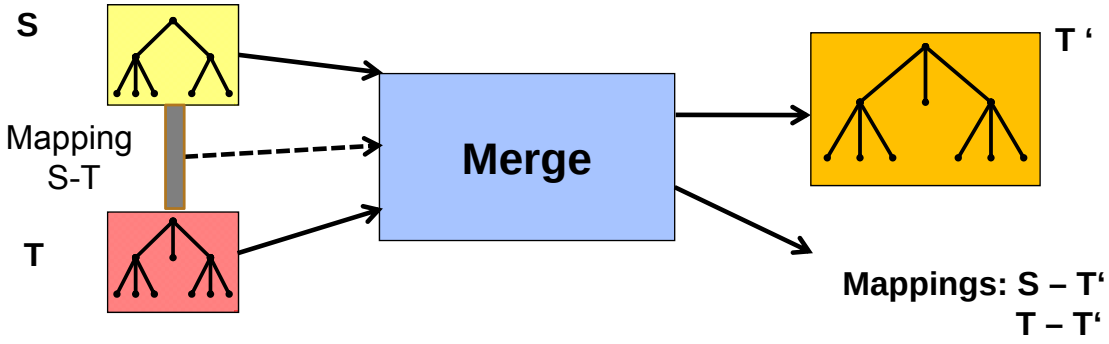
- Ontologies and ontology matching
- Ontology merging
  - Symmetric vs. asymmetric merge
  - ATOM approach
  - Evaluation
- Ontology Evolution
  - Desiderata of evolution support
  - COntoDIFF
  - Region Analyzer
- Conclusions and outlook







Ontology Merging



- Process of merging input ontologies into integrated ontology
  - symmetric merge or
  - target-driven merge
- Optional use of match mapping between input ontologies

Previous work

- Huge amount of work on schema integration
  - Mostly addressed both matching and merging
  - Complex solutions with high degree of manual interaction
- Recent schema merging approaches based on predetermined match mapping
  - [Pottinger and Bernstein 2003], [Pottinger and Bernstein 2008]
  - [Chiticariu, Kolaitis, Popa 2008], [Radvan, Popa , Stanoi, Younis 2009]
  - ...

Previous work (2)

- Relatively few approaches for ontology merging
  - ✓ PROMPT (1999-2000), Chimaera (2000)
  - ✓ FCA-Merge (2001), ...
- Combined approaches for match and merge
- High degree of user intervention needed
- Symmetric merge
  - full preservation of both input ontologies

- **Need for**
  - Match-based ontology merging
  - Target-driven merge
    -> ATOM approach

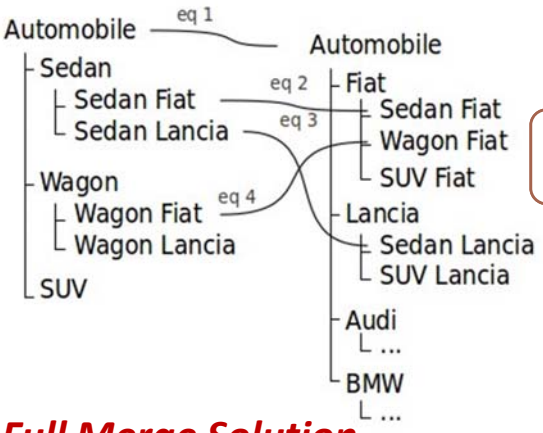
Symmetric Merge

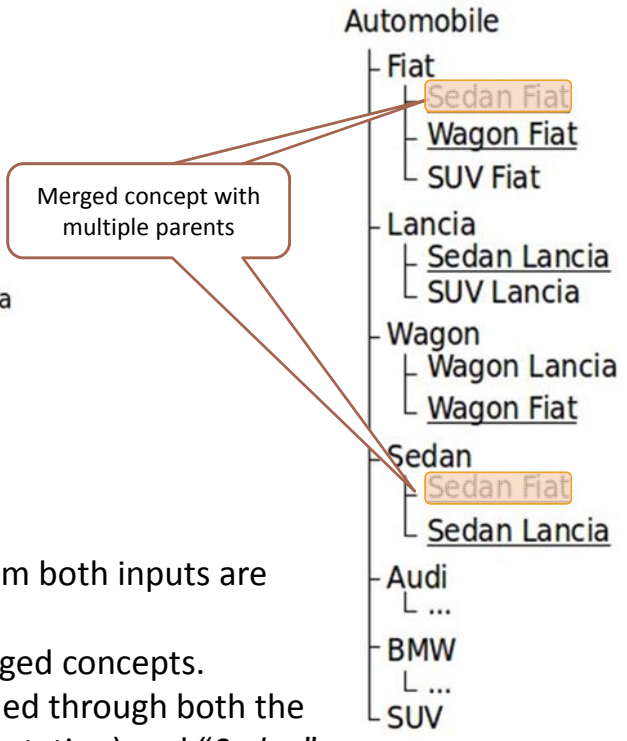
- Combines equivalent concepts and maintains all remaining input concepts and relationships of both input ontologies (**Full Merge**)

- Maintaining different organizations of the same information can reduce understandability and introduce **semantic overlap**
  - e.g. multiple paths to the same information / multiple inheritance

- **Reduced stability** for a preferred input ontology such as mediator ontology
  - e.g. product catalog of a price comparison portal

Example





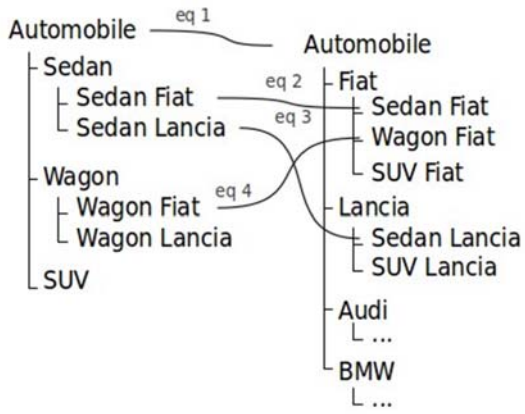
Full Merge Solution

- ✓all concepts and relationships from both inputs are preserved
- ✓Introduces multiple paths to merged concepts.
  - ✓E.g., "*Sedan Fiat*" can be reached through both the concepts "*Fiat*" (target representation) and "*Sedan*" (source representation)
- ✓Reduced understandability by mixing different categorizations

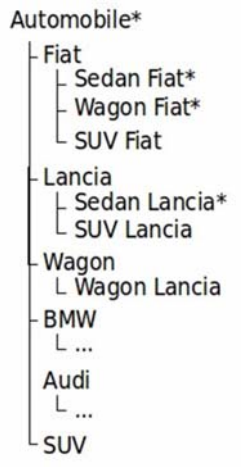
ATOM approach*

- **A**utomatic **T**arget-Driven **O**ntology **M**erging
- Asymmetric, target-driven merge approach
- Aims at reduced semantic overlap in merge result
  - Preserves target ontology but drops source concepts and relationships that would introduce redundancy in the merge result
- Utilization of input match mapping
  - Base version: equivalence correspondences
  - Optionally: is-a / inverse-is-a correspondences
- Automatic generation of default solution(s)
  - Result may interactively be adapted by users if needed
- Mapping generation, e.g. for instance migration

* Raunich, S., Rahm, E.: *ATOM: Automatic Target-driven Ontology Merging*, Proc. ICDE 2011



ATOM Solution

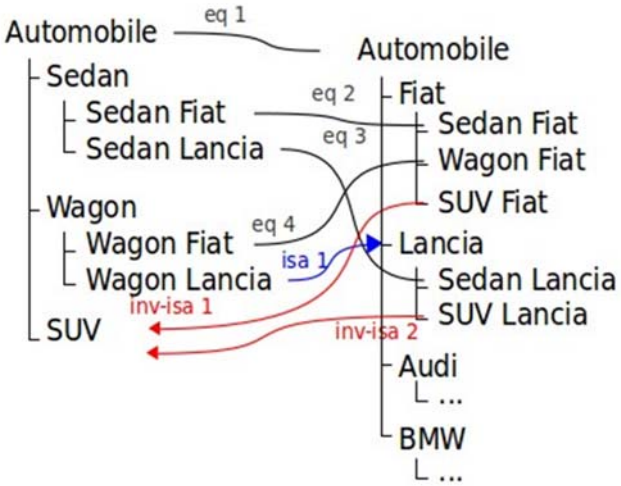


Full Merge Solution

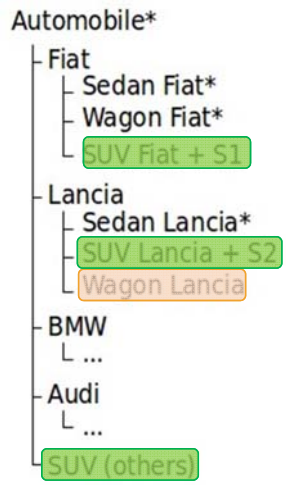
ATOM Solution

- Preserves the target ontology
- More compact than the full merge solution
- No multiple inheritance

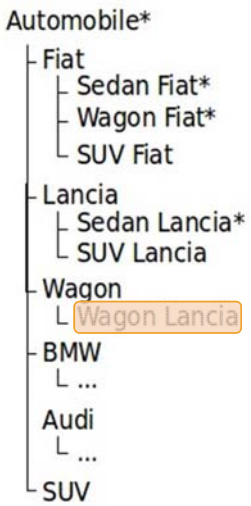
- semantic overlap is only partially reduced
  - some concepts could be better placed (e.g. Wagon Lancia)
  - overlap between general SUV concept and SUV Fiat and SUV Lancia
- More semantic input mapping allows further improvement



ATOM with Extended Mapping



ATOM with only Equivalence Mapping



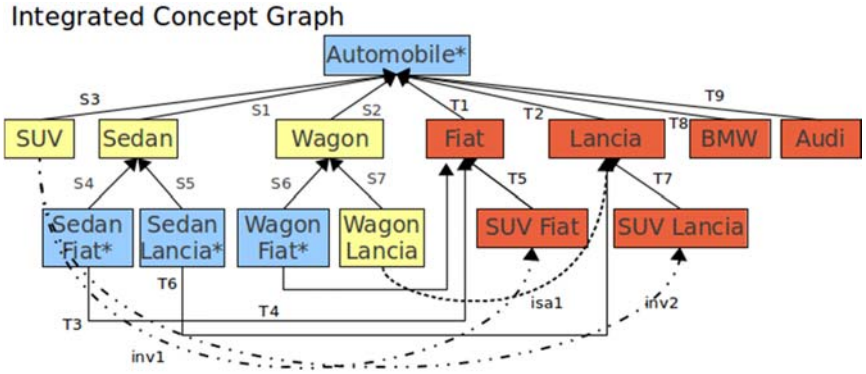
ATOM Solution with Extended Input Mapping

- ***is-a*** and ***inverse-is-a*** relationships in addition to equivalence correspondences

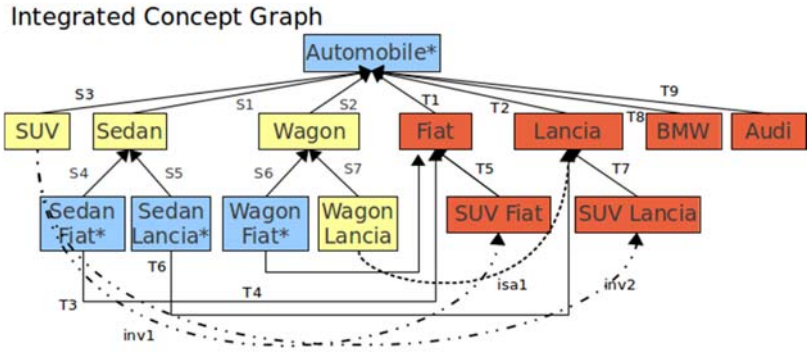
- The concept *Wagon Lancia* is now well placed
- no more overlap between general SUV concept and the more specific concepts SUV Fiat and SUV Lancia

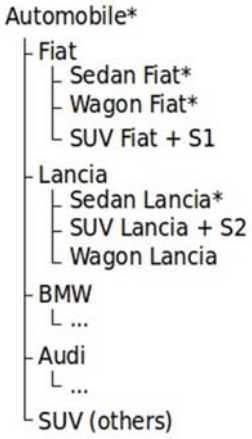
Merge Algorithm (1)

- Preliminary phase
  Uses input ontologies and mapping to create Integrated Concept Graph containing all S- and T-concepts and S- and T edges
  - Translate all input concepts merging equivalent ones
  - A „labeled“ edge for each input relationship from S or T
  - A „labeled“ edge for each „is-a“ and „inv-isa“ correspondence



Merge Algorithm (2)



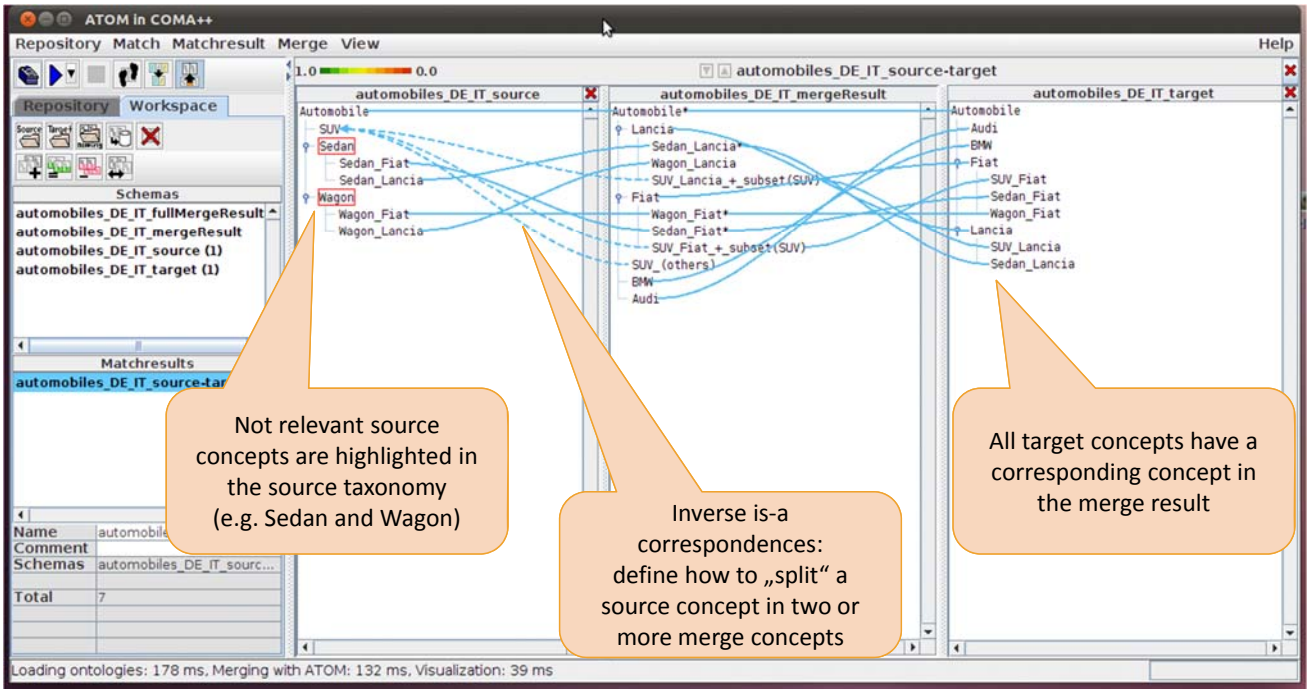


- Main Phase:
  - ✓ take over the target concepts and relationships in the merge result (*target preservation*)
  - ✓ include all leaf nodes from both target and source taxonomy (*instance preservation*)
  - ✓ include only inner source concepts that do not introduce redundant paths to leaf nodes (*control of semantic overlap*)
  - ✓ translate is-a and inverse-is-a relationships for improving the merge result

Main Properties of ATOM approach

- P1 – Target Preservation
  - all target concepts and relationships remain in the result
- P2 – Correspondence Preservation
- P3 – Instance Preservation
  - for both input ontologies
  - mappings S-T‘ and T-T‘ specify where instances should migrate
- P4 – Controlled Semantic Overlap
  - for each target (T) concept t in the merge result the number of root paths to t in the merge result T’ is not higher than in T
  - limits multiple paths to leaf nodes / nodes with associated instances

COMA++ integration



Experimental Results

Merge Example		Anatomy		eBay Catalog	
		Mouse	NCI	v94	v93
Input size	Concepts	2,700	3,300	21,000	23,500
	Leaf paths	2,300	**2,600**	18,400	19,700
# correspondences		~1,000		~20,200	
FULL solution	Concepts	5,100		23,400	
	Leaf paths	**12,900**		21,600	
ATOM solution	Concepts	5,000		23,300	
	Leaf paths	**6,900**		20,400	
Execution time		1 second		7 seconds	

Future Work

- Overcome current restrictions
  - Is-a relationships (taxonomies)
  - Instances for leaf concepts only
- User interaction to adapt proposed merge result
- Evaluation of merge quality

Agenda

- Ontologies and ontology matching
- Ontology merging
  - Symmetric vs. asymmetric merge
  - ATOM approach
  - Evaluation
- Ontology Evolution
  - Desiderata of evolution support
  - COntoDIFF
  - Region Analyzer
- Conclusions and outlook

Evolution of Life Science Ontologies

- Heavy evolution of life science ontologies
  - continuous release of new versions
- Evolution analysis of 16 ontologies*
  - Average growth of 60% in last four years
    - ✓ Gene Ontology: from 17,400 to 26,000 concepts
    - ✓ NCI Thesaurus: from 36,000 to 64,000 concepts
  - Deletes and changes also common
- Ontologies become more structured
  - longer paths
  - increasing use of part-of relationships in addition to is-a

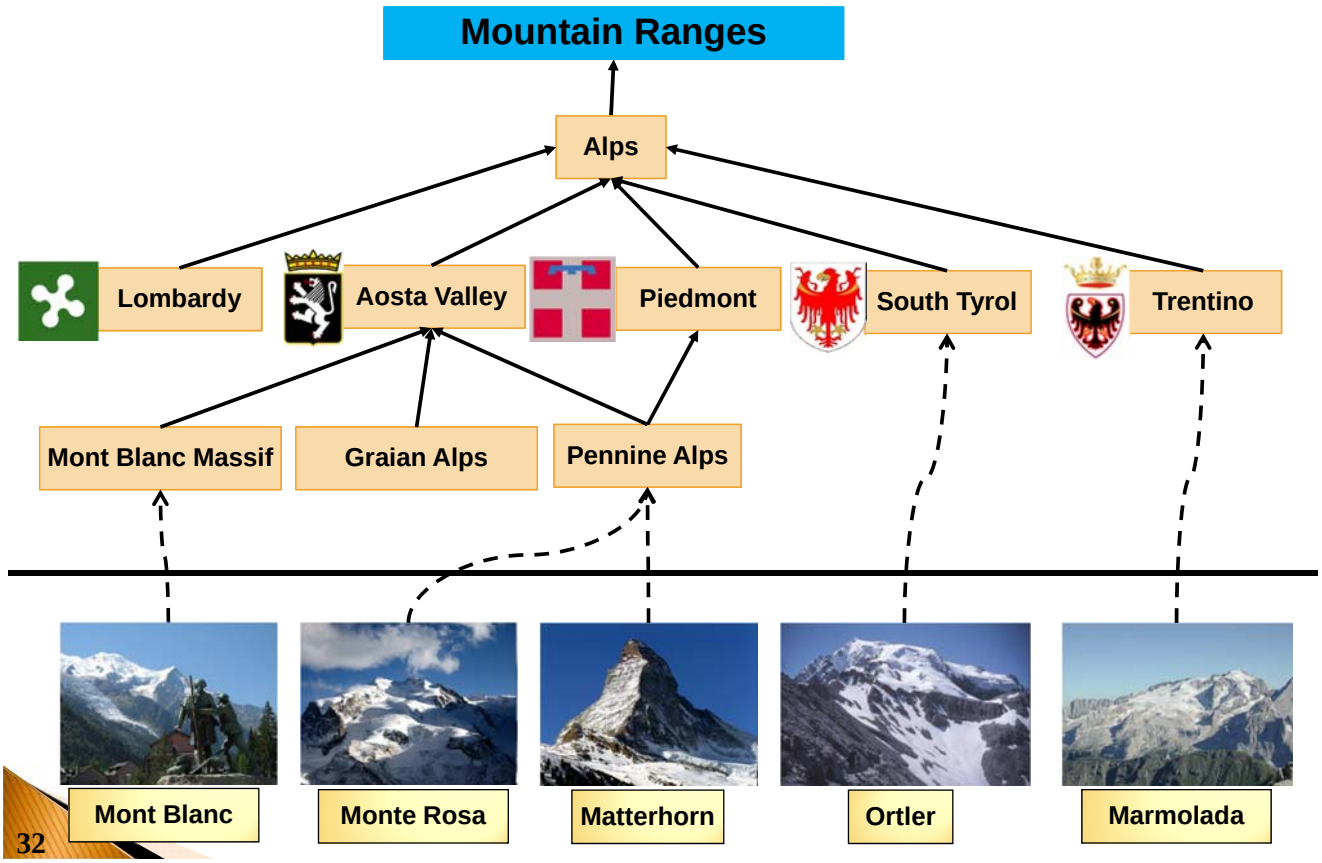
www.izbi.de/onex

* Hartung, M; Kirsten, T; Rahm, E.: *Analyzing the Evolution of Life Science Ontologies and Mappings.* Proc. 5th Data Integration in the Life Sciences (DILS), 2008

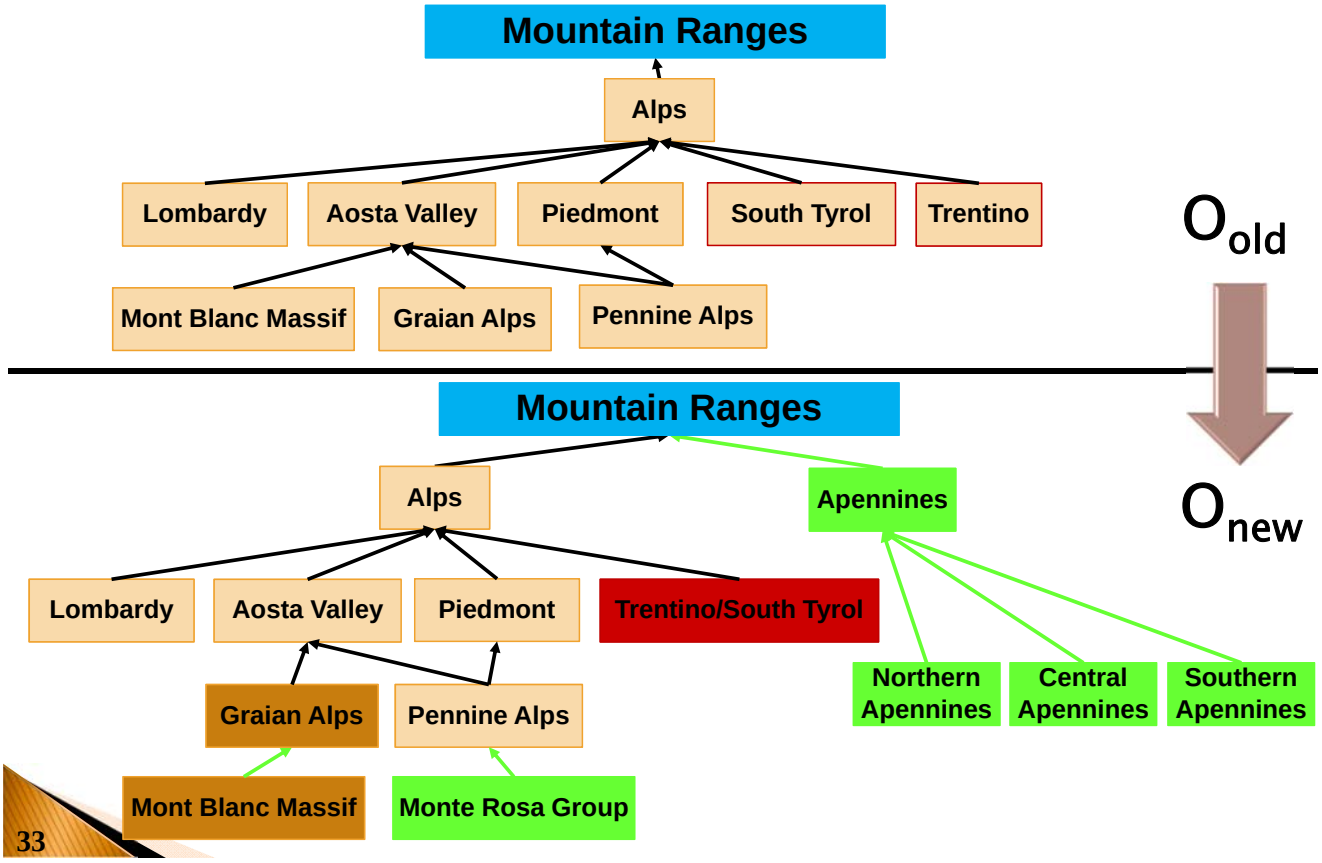
Effects of Ontology Evolution

- Annotations / Instances are impacted
  - Concept deletes or changes may require the deletion or migration of associated instances
- Analysis/query results may change
  - e.g. earlier findings for term enrichment analysis may become invalid
- Ontology mappings may have to be adapted

Exemplary Ontology



Ontology Evolution Example



Possible evolution effects for example

- Instance migration
  - E.g. for moved or for merged concepts
  - Move instances of South Tyrol and Trentino to merged category
    {Ortler,…} ∪ {Marmolada,…} → Trentino/South Tyrol

- Effect on query results
  - *e.g. "What are 4000m summits in the Graian Alps?"*

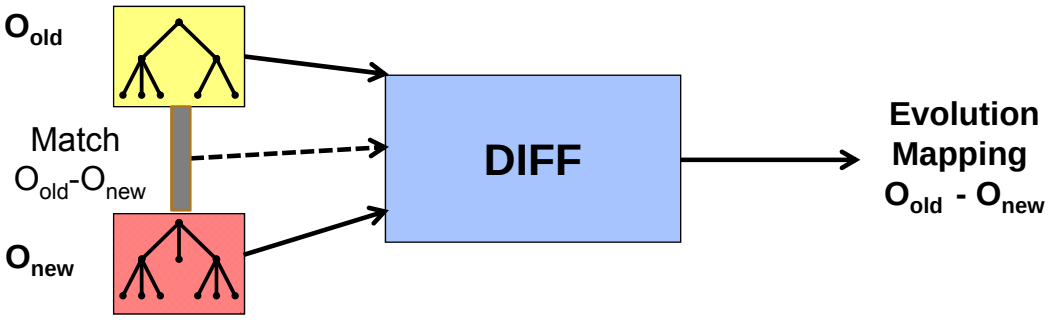
  - Old result: Gran Paradiso (1 result)
  - New result: Gran Paradiso, Mont Blanc, … (12 results)

Desiderata for evolution support*

- Versioning support
  - Support of different explicit ontology versions
  - Backward/Forward compatibility
- Change specification
  - Incrementally: set of simple and complex changes
  - Directly: provision of evolved ontology
- Declarative evolution mappings
  - Set of changes or use of logical mapping expressions
  - Automatic mapping generation by Diff between versions
- Automatic instance migration
- Propagation of changes to related mappings / ontologies
- Powerful tool infrastructure

*Hartung, M., Terwilliger, J., *Rahm, E.: Recent Advances in Schema and Ontology Evolutoon.* In: *Schema Matching and Mapping*, Springer-Verlag, 2011

Ontology DIFF



- Optional use of match mapping between input ontologies
- Evolution mapping can be expressed by set of change operation that evolve O_{old} into O_{new}
  - Simple (basic) and complex operations
  - Usable for evolution analysis, instance migration, …

COntoDiff

- Complex Ontology Diff
  - Match as basis for determining the DIFF
  - Rule-based DIFF approach
- Input: *Match Mapping match(O_{old},O_{new})*
  - ✓ Result of a match between O_{old} und O_{new} consisting of correspondences: *matchC (c1,c2)*
- Output: *Evolution Mapping diff(O_{old},O_{new})*
  - ✓ Changes between O_{old} and O_{new} as a set of simple and complex change operations
  - ✓ Distinction between $diff_{basic}$ and $diff_{compact}$
- Goal: compact, expressive evolution mapping

Assumed Ontology Model

- Focus on practically used ontologies
- Ontology O consists of a set of *concepts/categories* interconnected by *relationships* (e.g. of type „is-a“ or „part-of“).
  - O is represented by a **DAG** and has a designated root concept.
  - Concepts have *attributes*, e.g. Id, Name, Description
  - Concepts may have associated *instances*
- Ontologies may be versioned

Supported Change Operations

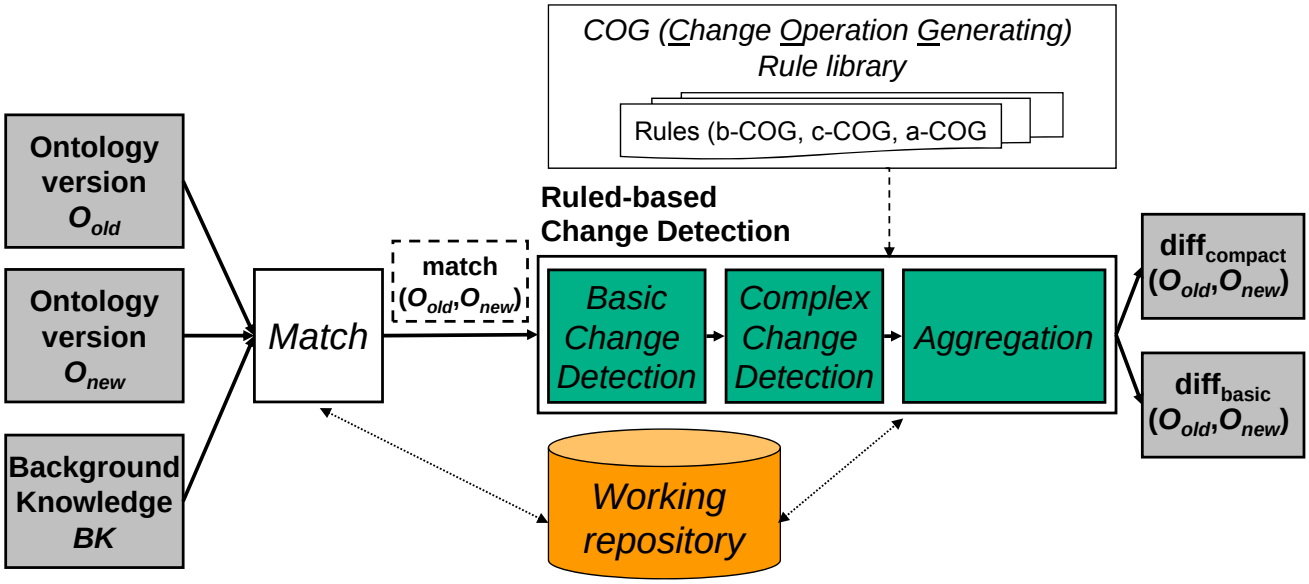
- **Basic Changes**
  - *add*, *del*, *map* for *concepts*, *attributes* and *relationships*

	concept	relationship	attribute
map	*mapC*	*mapR*	*mapA*
add	*addC*	*addR*	*addA*
del	*delC*	*delR*	*delA*

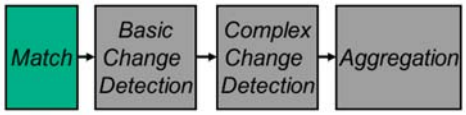
  - ✓ Example: *mapC(c1,c2)*: *c1* is mapped to different *c2*

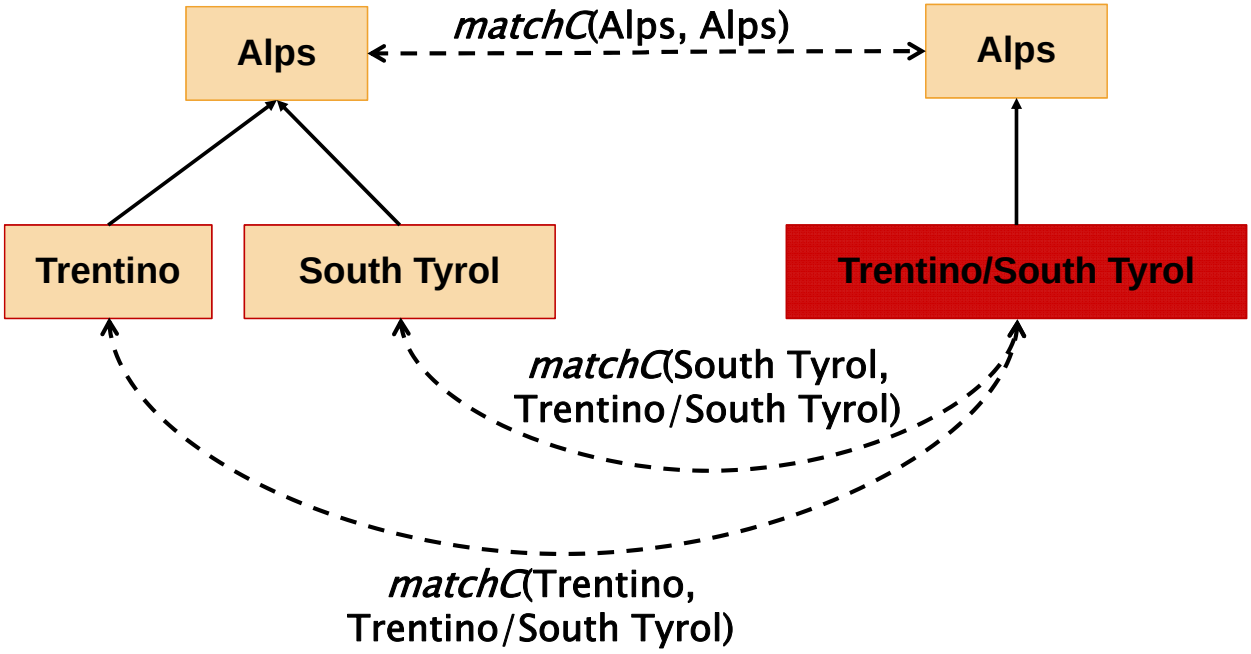
- **Complex Changes**
  - *merge:* merge of multiple concepts into one
  - *split:* split of a concept into multiple concepts
  - *move:* rearrange of a concept within the ontology
  - *addSubGraph:* addition of a complete subgraph
  - …

Schematic Overview of COntoDiff



Match Input

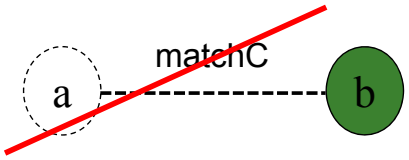




Basic Change Detection

- b-COG rules (COG: Change Operation Generating)
  - Determination of all basic changes
- *add:*

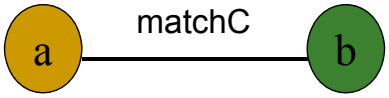
$$b \in O_{new} \wedge \nexists a(a \in O_{old} \wedge matchC(a,b) \rightarrow \textbf{create}[addC(b)]$$



addC(Monte Rosa Group)
addC(Apennines)
…

- *mapC*:

$$a \in O_{old} \wedge b \in O_{new} \wedge a \neq b \wedge matchC(a,b) \rightarrow \textbf{create}[mapC(a,b)]$$



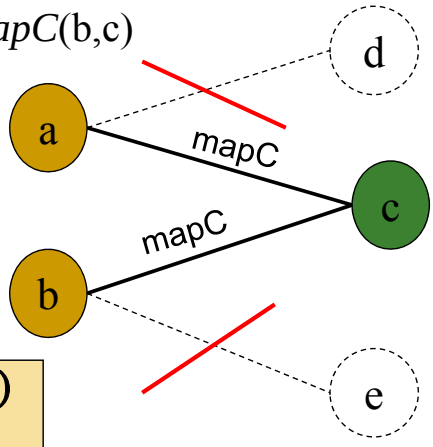
mapC(South Tyrol, Trentino/South Tyrol)
mapC(Trentino, Trentino/South Tyrol)

Complex Change Detection

- c-COG rules
  - Generation of complex changes

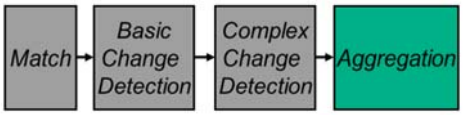
- *merge*:

$a,b \in O_{old} \wedge c \in O_{new} \wedge a \neq b \wedge mapC(a,c) \wedge mapC(b,c)$
$\wedge \nexists\, d(d \in O_{new} \wedge mapC(a,d) \wedge c \neq d)$
$\wedge \nexists\, e(e \in O_{new} \wedge mapC(b,e) \wedge c \neq e)$
$\rightarrow$ **create**[*merge*({a},c), *merge*({b},c)],
eliminate[*mapC*(a,c), *mapC*(b,c)]



merge({South Tyrol}, Trentino/South Tyrol)
merge({Trentino}, Trentino/South Tyrol)

Aggregation



- a-COG rules (recursively applicable)
  - Compaction of complex changes

- *merge*:

$c \in O_{new} \wedge A,B \subseteq O_{old} \wedge merge(A,c) \wedge merge(B,c) \wedge A \neq B$
$\rightarrow$ **create**[*merge*(A$\cup$B,c)], **eliminate**[*merge*(A,c), *merge*(B,c)]

merge({Trentino, South Tyrol}, Trentino/South Tyrol)

Final Result

$diff_{basic}$

mapC(Trentino,Trentino/South Tyrol)
mapC(South Tyrol,Trentino/South Tyrol)
addC(Monte Rosa Group)
addR(Monte Rosa Group,Pennine Alps)
addC(Apennines)
addR(Apennines,Italian Mountain Ranges)
addC(Central Apennines)
addC(Southern Apennines)
addC(Northern Apennines)
addR(Northen Apennines,Apennines)
addR(Central Apennines,Apennines)
addR(Southern Apennines,Apennines)
addR(Mont Blanc Massif,Graian Alps)
delR(Mont Blanc Massif,Aosta Valley)

14 basic changes

$diff_{compact}$

merge({Trentino,South Tyrol}, Trentino/South Tyrol)
addLeaf (Monte Rosa Group, {Pennine Alps})
addSubGraph(Apennines,{Northern Apennines, Central Apennines, Southern Apennines})
move(Mont Blanc Massif, Aosta Valley, Graian Alps)

4 complex changes

COntoDIFF Evaluation

- Gene Ontology version diffs
  - 2008: 2008-01 → 2009-01
  - 2009: 2009-01 → 2010-01

$diff_{basic}$

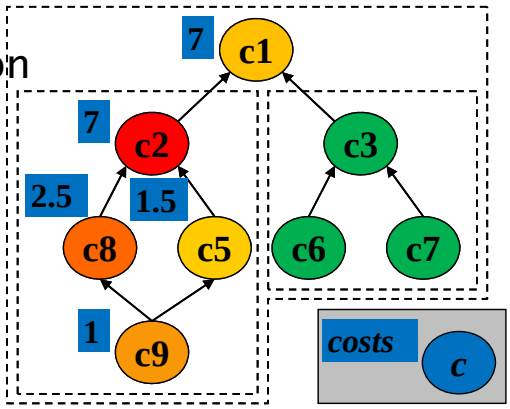
	2008	2009
add	12,385	11,477
del	3,210	1,841
map	186	186
Σ	15,781	13,504

$diff_{compact}$

	2008	2009
add	4,187	1,355
del	1,407	316
map	0	0
addLeaf	768	796
merge	70	83
move	1,499	1,200
substitute	0	1
toObsolete	225	66
addSubGraph	294	467
Σ	8,450	4,284

Where are changes located?

- Estimation of change-intensive and/or stable ontology parts
- Ontology region *OR*
  - Subgraph of an ontology with a single root concept *rc*
  - Contains all concepts in the *is_a* subgraph of *rc*
- Cost model
  - Relative cost per change operation
- Aggregate change costs per region
  - ✓ Absolute/relative region size
  - ✓ Absolute/average change costs
  - ✓ etc.



Hartung, M; Gross, A; Kirsten, T; Rahm, E: *Discovering Evolving Regions in Life Science Ontologies* Proc. 7th Intl. Conference on Data Integration in the Life Sciences (DILS), 2010

Algorithm Sketch

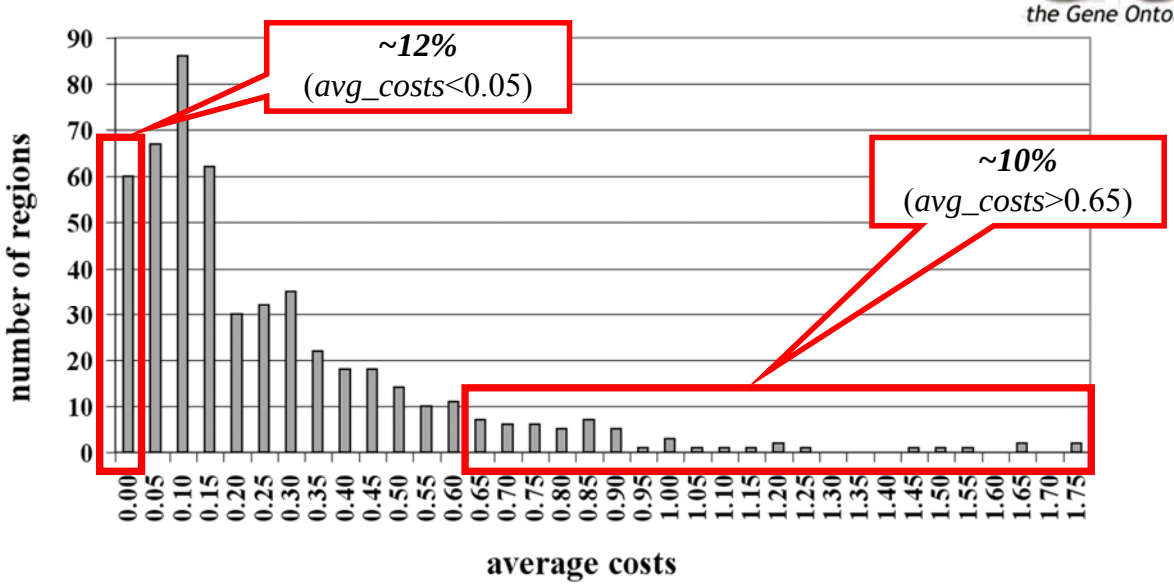
- Input: two ontology versions O_{old} and O_{new} change costs
- Output: O_{new} with computed aggregated costs

- Algorithm:
  1. Compute the DIFF (changes) between O_{old} and O_{new}
  2. Assign change costs to concepts affected by a change
  3. Propagate assigned costs upwards in O_{old} and O_{new}
  4. Unify propagated costs of O_{old} and O_{new}
  5. Apply region measures

- Extensible to multiple (>2) ontology versions

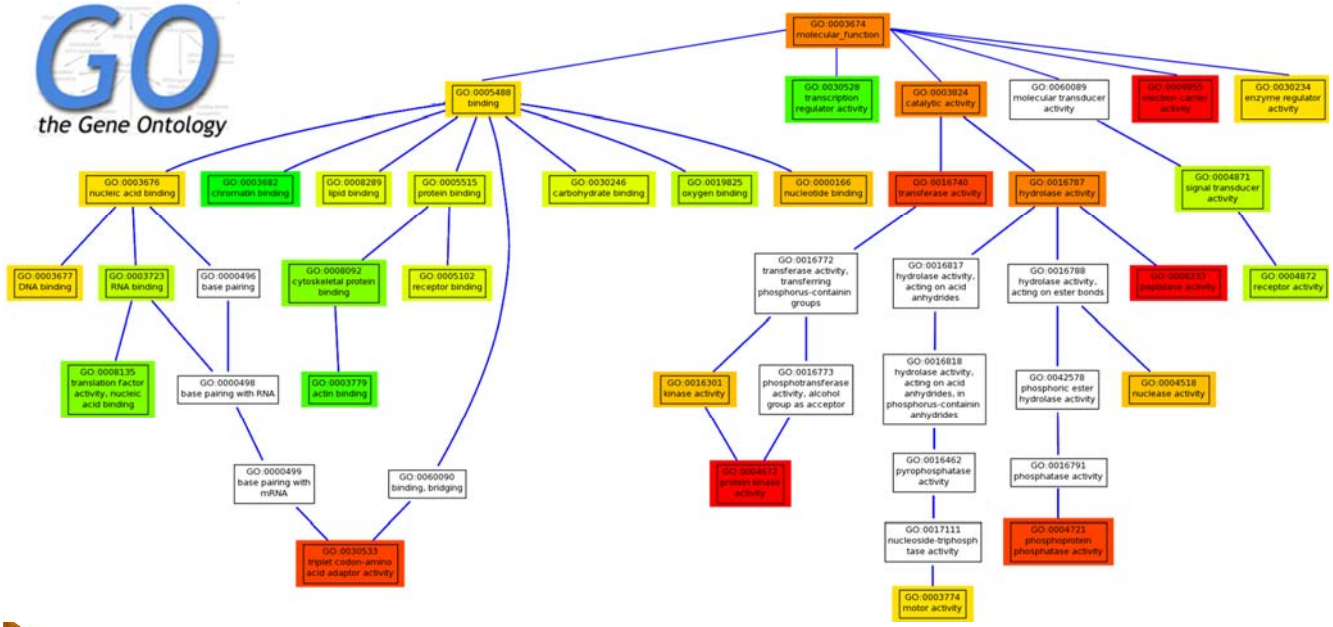
Distribution of Ontology Regions

- Distribution of ontology regions w.r.t. ***avg_costs***
  - Minimum ***rel_size*** of 0.3%
  - *Example:* GO Biological Processes in 2009



Stability of Top-Level Categories

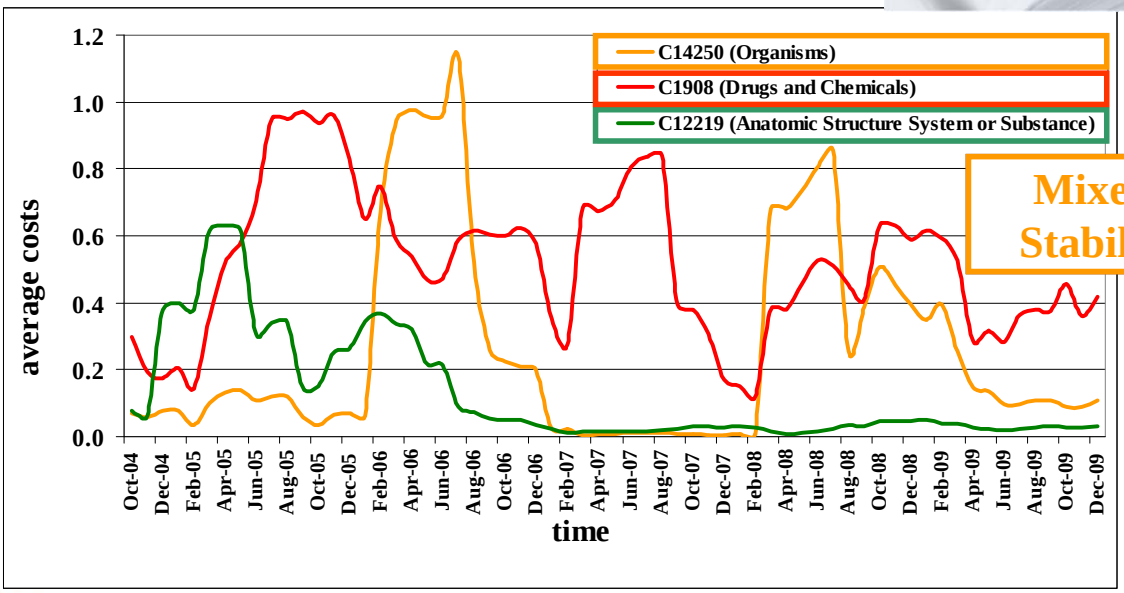
- Stability of top-level categories ("slim terms") in GO Molecular Functions (between 2007 and 2009)



Tracking of Change Intensities

- Thesaurus of NCI with 20 main categories
  - sliding window of ½ year between 2004 and 2009
- Three evolution patterns

NCIthesaurus



Agenda

- Ontologies and ontology matching
- Ontology merging
  - Symmetric vs. asymmetric merge
  - ATOM approach
  - Evaluation
- Ontology Evolution
  - Desiderata of evolution support
  - COntoDIFF
  - Region Analyzer
- Conclusions and outlook

Conclusions

- Match is key operator for ontology management
  - Many prototypes supporting a variety of matchers
  - More work still desirable
- Merge
  - Match mapping should be exploited
  - Asymmetric, target-driven approaches like ATOM are promising in different areas
  - Improved stability for mediator ontologies after integrating new source ontologies
- Ontology evolution
  - Support for Diff mappings helps to better deal with changes
  - COntoDiff: rule-based generation of compact evolution mappings

Areas for future work

- Ontology matching
  - Large-scale matching
  - Improved reuse of previous match results
  - Semantic mappings
  - …
- Merge
  - More general target-driven merge approaches
  - More work on reducing semantic overlap
  - Benchmark for merge approaches

Areas for future work (2)

- Ontology evolution
  - Comparison of different DIFF methods
  - Alternative mapping models
  - Further mapping operations, e.g. composition
  - Evolution of ontology mappings
- Ontology matching/merging/diff for **Linked Data**

References

- Bellahsene, Z.; Bonifati, A.; Rahm, E. (eds.): Schema Matching and Mapping. Springer-Verlag, 2011
- Hartung, M; Gross, A; Kirsten, T; Rahm, E: Discovering Evolving Regions in Life Science Ontologies. Proc. 7th Intl. Conf. on Data Integration in the Life Sciences (DILS), 2010
- Hartung, M.; Groß, A.; Rahm, E.: Rule-based Generation of Diff Evolution Mappings between Ontology Versions. CoRR abs/1010.0122, 2010
- Hartung, M; Kirsten, T; Rahm, E.: Analyzing the Evolution of Life Science Ontologies and Mappings. Proc. 5th Data Integration in the Life Sciences (DILS), 2008
- Hartung, M., Terwilliger, J., Rahm, E.: Recent Advances in Schema and Ontology Evolutoon. In: Schema Matching and Mapping, Springer-Verlag, 2011
- Rahm, E.: Towards large-scale schema and ontology matching. In: Schema Matching and Mapping, Springer-Verlag, 2011
- Raunich, S., Rahm, E.: ATOM: Automatic Target-driven Ontology Merging, Proc. ICDE 2011