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”
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.
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
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**

<table>
<thead>
<tr>
<th></th>
<th>Cupid</th>
<th>COMA++</th>
<th>Falcon</th>
<th>Rimom</th>
<th>Asmov</th>
<th>Agr.Maker</th>
<th>OII Harmony</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>relational</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>schemas</td>
<td>XML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ontologies</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>OAEI participation</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>compreh. GUI</td>
<td>-</td>
<td>✓</td>
<td>(✓)</td>
<td>?</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Matchers</td>
<td>linguistic</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>structure</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>Instance</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>use of ext.dictionaries</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>schema partitioning</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>parallel matching</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>dyn. matcher selection</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>mapping reuse</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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
  - 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
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*

- Automatic Target-Driven Ontology Merging
- 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.

**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 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

![Integrated Concept Graph](image)

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

Not relevant source concepts are highlighted in the source taxonomy (e.g. Sedan and Wagon)

Inverse is-a correspondences: define how to “split” a source concept in two or more merge concepts

All target concepts have a corresponding concept in the merge result
Experimental Results

<table>
<thead>
<tr>
<th>Merge Example</th>
<th>Anatomy</th>
<th>eBay Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mouse</td>
<td>NCI v94</td>
</tr>
<tr>
<td>Input size</td>
<td>Concepts</td>
<td>2,700</td>
</tr>
<tr>
<td></td>
<td>Leaf paths</td>
<td>2,300</td>
</tr>
<tr>
<td># correspondences</td>
<td>~1,000</td>
<td>~20,200</td>
</tr>
<tr>
<td>FULL solution</td>
<td>Concepts</td>
<td>5,100</td>
</tr>
<tr>
<td></td>
<td>Leaf paths</td>
<td>12,900</td>
</tr>
<tr>
<td>ATOM solution</td>
<td>Concepts</td>
<td>5,000</td>
</tr>
<tr>
<td></td>
<td>Leaf paths</td>
<td>6,900</td>
</tr>
<tr>
<td>Execution time</td>
<td>1 second</td>
<td>7 seconds</td>
</tr>
</tbody>
</table>

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

* 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
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**


**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, ...
Complex Ontology Diff
- Match as basis for determining the DIFF
- Rule-based DIFF approach

Input: Match Mapping $\text{match}(O_{\text{old}}, O_{\text{new}})$
- Result of a match between $O_{\text{old}}$ und $O_{\text{new}}$ consisting of correspondences: $\text{matchC}(c1, c2)$

Output: Evolution Mapping $\text{diff}(O_{\text{old}}, O_{\text{new}})$
- Changes between $O_{\text{old}}$ and $O_{\text{new}}$ as a set of simple and complex change operations
- Distinction between $\text{diff}_{\text{basic}}$ and $\text{diff}_{\text{compact}}$

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

<table>
<thead>
<tr>
<th></th>
<th>concept</th>
<th>relationship</th>
<th>attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>map</td>
<td>mapC</td>
<td>mapR</td>
<td>mapA</td>
</tr>
<tr>
<td>add</td>
<td>addC</td>
<td>addR</td>
<td>addA</td>
</tr>
<tr>
<td>del</td>
<td>delC</td>
<td>delR</td>
<td>delA</td>
</tr>
</tbody>
</table>

- 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 COnontoDiff

- COG (Change Operation Generating) Rule library
  - Rules (b-COG, c-COG, a-COG)
- Ruled-based change detection
- Aggregation
- Working repository

- Match

- Ontology version \(O_{old}\)
- Ontology version \(O_{new}\)
- Background Knowledge \(BK\)
- \(diff_{compact}(O_{old}, O_{new})\)
- \(diff_{basic}(O_{old}, O_{new})\)
**Match Input**

- $matchC(\text{Alps}, \text{Alps})$
- $matchC(\text{South Tyrol}, \text{Trentino/South Tyrol})$
- $matchC(\text{Trentino}, \text{Trentino/South Tyrol})$

**Basic Change Detection**

- **b-COG** rules (COG: Change Operation Generating)
  - Determination of all basic changes
  - $add$: 
    
    - $b \in O_{new} \land \exists a (a \in O_{old} \land matchC(a, b))$
    - $\rightarrow create[addC(b)]$

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

- $addC(\text{Monte Rosa Group})$
- $addC(\text{Apennines})$
- $mapC(\text{South Tyrol, Trentino/South Tyrol})$
- $mapC(\text{Trentino, Trentino/South Tyrol})$
Complex Change Detection

- c-COG rules
  - Generation of complex changes

- merge:
  
  \[
  a, b \in O_{\text{old}} \land c \in O_{\text{new}} \land a \neq b \land \text{mapC}(a, c) \land \text{mapC}(b, c) \\
  \land \nexists d \in O_{\text{new}} \land \text{mapC}(a, d) \land c \neq d \\
  \land \nexists e \in O_{\text{new}} \land \text{mapC}(b, e) \land c \neq e \\
  \rightarrow \text{create}[	ext{merge}\{\{a\}, c\}, \text{merge}\{\{b\}, c\}], \\
  \text{eliminate}[	ext{mapC}(a, c), \text{mapC}(b, c)]
  \]

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

Aggregation

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

- merge:
  
  \[
  c \in O_{\text{new}} \land A, B \subseteq O_{\text{old}} \land \text{merge}(A, c) \land \text{merge}(B, c) \land A \neq B \\
  \rightarrow \text{create}[	ext{merge}(A \cup B, c)], \text{eliminate}[	ext{merge}(A, c), \text{merge}(B, c)]
  \]

merge({Trentino, South Tyrol}, Trentino/South Tyrol)
### Final Result

**diff\_basic**

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

**diff\_compact**

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

**14 basic changes**

**4 complex changes**

### COntoDIFF Evaluation

- **Gene Ontology version diffs**

#### diff\_compact

<table>
<thead>
<tr>
<th></th>
<th>2008</th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>add</strong></td>
<td>4,187</td>
<td>1,355</td>
</tr>
<tr>
<td><strong>del</strong></td>
<td>1,407</td>
<td>316</td>
</tr>
<tr>
<td><strong>map</strong></td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>addLeaf</strong></td>
<td>768</td>
<td>796</td>
</tr>
<tr>
<td><strong>merge</strong></td>
<td>70</td>
<td>83</td>
</tr>
<tr>
<td><strong>move</strong></td>
<td>1,499</td>
<td>1,200</td>
</tr>
<tr>
<td><strong>substitute</strong></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>toObsolete</strong></td>
<td>225</td>
<td>66</td>
</tr>
<tr>
<td><strong>addSubGraph</strong></td>
<td>294</td>
<td>467</td>
</tr>
<tr>
<td><strong>Σ</strong></td>
<td>8,450</td>
<td>4,284</td>
</tr>
</tbody>
</table>
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.

Algorithm Sketch

- Input: two ontology versions \( O_{old} \) and \( O_{new} \)
- 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 \( \frac{1}{2} \) year between 2004 and 2009
- Three evolution patterns

![Graph showing change intensities over time]

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

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