

Towards a Benchmark for Ontology Merging

Salvatore Raunich, Erhard Rahm

University of Leipzig, Germany

Abstract. Benchmarking approaches for ontology merging is challenging and has received little attention so far. A key problem is that there is in general no single best solution for a merge task and that merging may either be performed symmetrically or asymmetrically. As a first step to evaluate the quality of ontology merging solutions we propose the use of general metrics such as the relative coverage of the input ontologies, the compactness of the merge result as well as the degree of introduced redundancy. We use these metrics to evaluate three merge approaches for different merge scenarios.

1 Motivation

Ontologies and taxonomies are increasingly used to semantically categorize or annotate information, e.g., for e-commerce or e-science. For example, product catalogs of online shops or comparison portals categorize products to help users and applications finding relevant information. Since many ontologies refer to the same domain and to the same objects, there is a growing need to integrate or merge such related ontologies. The goal is to create a merged ontology providing a unified view on two or more input ontologies.

Ontology merging is a challenging problem especially for large and heterogeneous ontologies and require semi-automatic approaches to reduce the manual effort. Several such merge approaches have already been proposed, however their relative quality is largely unknown. One increasingly adopted and promising idea is to decompose the complex integration problem into match and merge subtasks and leverage the advances made for automatic ontology and schema matching [13] to solve the first subproblem. The merge subtask can then utilize a match mapping identifying corresponding concepts in the input ontologies that should be merged. Such a *match-based merging* is followed in [11] [17] [16] [12] [15]. In general, merging can be symmetric or asymmetric with respect to the input ontologies. *Symmetric solutions* (e.g., [10], [9], [8] [7]) are most common and aim at completely integrating all input ontologies with the same priority. *Asymmetric approaches*, by contrast, take one of the input ontologies as the target and merge the other input ontologies into this target [11] [15] [6] thereby giving preference to the target ontology.

Given the different merge approaches we see an increasing need to quantitatively evaluate their quality and performance. For the subproblem of ontology matching such evaluations are now quite common [1] [3] and there is also a benchmark for determining schema mappings [2]. Typically the quality of a match algorithm is determined by evaluating it on some match problems for which a manually defined perfect match result is provided for comparison. While a similar approach for evaluating merge approaches has been advocated for in [4] we argue that there is in general no single perfect merge

result but that there can be several similarly valid solutions. Also the quality of a merged ontology likely depends on the domain and its intended use making it difficult to define a general benchmark for ontology merging.

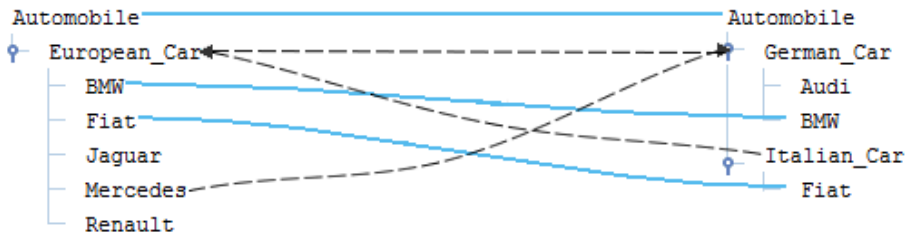


Fig. 1. Running Example

For illustration let us consider two sample taxonomies (also referred to as source and target taxonomies) in Figure 1 that classify European cars in different ways. The first (source) taxonomy uses a single concept *European_Car* while the target distinguishes between *German* and *Italian* cars. Let us suppose that a match mapping is already given as input (solid lines), automatically generated by a matching tool or manually designed by an expert user. Figure 2 shows four possible solutions that a merging tool or human expert can produce. Merged concepts present in both inputs are marked with a star in the solutions. For example, the concept *BMW** is a combined concept covered by an equivalence correspondence in the match mapping. The problem is to evaluate the quality of the different solutions, e.g., to find the best merge result.

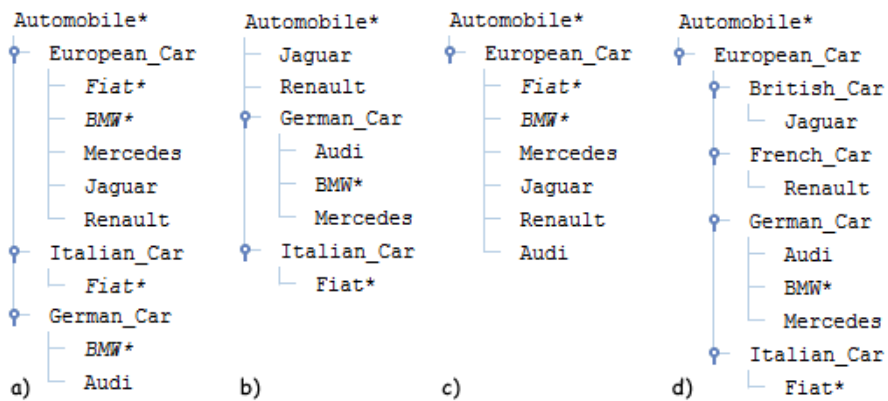


Fig. 2. a) Full Merge - b) Target-driven - c) Source-driven - d) Expert Result

Two of the solutions (a,d) are the result of a symmetric and two the result of an asymmetric merging (b,c). Solution (a) is the output of a straight-forward *Full Merge* approach that takes the union of the input ontologies and combines equivalent (matching) concepts. It is easy to see that such a Full Merge preserves all input concepts and relationships. However, the example shows that such a solution is not always desirable since it can introduce a semantic overlap (e.g., between *Italian_Car*, *German_Car* and *European_Car*) by redundantly representing the same information and introducing multiple inheritance for some concepts (*Fiat* and *BMW*). Furthermore, the concept *Mercedes* should be better placed under the more specific concept *German_Car*.

Such problems may be avoided by asymmetric merge approaches such as ATOM [15] that preserve only the concepts and relationships of one input and integrate only non-redundant concepts of the other ontology. Solutions (b) and (c) show the ATOM results when the target input and source input ontology is chosen as the preferred ontology, respectively. In both cases, the result is more compact than for the Full Merge and has no multiple inheritance. In solution (b) the concepts Jaguar and Renault could not be well placed since they are directly under root concept *Automobile* as opposed to the other leaf concepts. Solution (c) is more homogeneous but has dropped inner target concepts (such as *German_Car*, *Italian_Car*) to conform to the structure of the source input ontology. Note that both asymmetric solutions preserve all leaf concepts and can thus maintain all information of the input ontologies.

Finally, (d) shows the most complex merge solution that may be manually generated by an expert user. Like (a) it retains all concepts and relationships of the input ontologies but it introduces new inner concepts for British and French cars. Furthermore, all leaf concepts (e.g. Mercedes) are perfectly placed without multiple inheritance. Such an improved placement and minimization of redundancy is difficult to achieve automatically. However, it can be supported by providing enhanced match mappings containing not only equivalence correspondences but also semantic relationships *less general* or *more general* correspondences [5] [15]. In Figure 1, we represent such correspondences with dashed lines, e.g., that source concept *Mercedes* is less general than the target concept *German_Car* while source concept *European_Car* is more general than target concepts *German_Car* and *Italian_Car*.

The four solutions show that there are different reasonable merge results and that an expert solution may require the introduction of additional concepts and the use of further knowledge not present in the input ontologies. Furthermore, a comprehensive expert merge solution is difficult to achieve especially for larger ontologies and may actually be suboptimal for some use cases. For example, if the merged ontology should be used as a mediated ontology (e.g. for a price comparison portal) it is important to be able to integrate additional source ontologies in an incremental way with minimal change to the previous mediated ontology. Such a use case could be better served with an asymmetric merge than with a symmetric solution.

As a first step to evaluate the quality of ontology merge solutions we advocate for the use of simple approaches that do not depend on the provision of a perfect merge result. We rather focus on general criteria such as the input coverage and compactness of the merge result as well as the degree of redundancy. Before we introduce our benchmark metrics in Section 3, we first discuss the general desiderata of a merge solution in more

detail (Section 2). In Section 4, we present first evaluation results using the introduced metrics for different merge scenarios and merge approaches.

2 Desiderata of a merge solution

Before one can determine the quality of an ontology merging algorithm it is necessary to define the requirements or desiderata for (semi-)automatic merging. Several research papers have defined such requirements [11] [17] [14] albeit for different kinds of ontologies and schemas and mostly without considering requirements imposed by specific use cases. Since we do not want to limit ourselves to the evaluation of a specific algorithm we merely discuss general requirements that we consider as especially significant for the quality of a merge solution, namely the preservation of information of the input ontologies and the understandability (and thus usability) of the merged ontology.

In this paper we focus on the wide-spread class of is-a-based ontologies (taxonomies) that may or may not include multiple inheritance. Ontologies consist of a set of concepts and a set of relationships of type "is-a" ("subclass") between concepts forming a rooted, acyclic graph structure. Additional kinds of relationships such as "part-of" are possible. A concept represents a collection of objects with similar properties; each concept has a name (or label) and optionally further attributes. Concepts may also have associated instances, e.g. product offers of a certain product category. We further focus on merging two ontologies at a time and assume the existence of an equality-based match mapping indicating pairs of corresponding concepts.

A key requirement for ontology merging is a high degree of *information preservation* so that the information represented in the input ontologies is preserved in the merge result. In this respect, we differentiate between symmetric and asymmetric merge approaches. For symmetric approaches, we require that all concepts, attributes, and relationships of both input ontologies are preserved in the merge result. In particular, for every concept in any input ontology there must be a corresponding concept in the merged ontology. Relationships may also implicitly be preserved via newly introduced intermediate concepts. For example, the input relationship "BMW is-a *European.Car*" of Figure 1 is implicitly preserved by "BMW is-a *German.Car* is-a *European.Car*". It is easy to see that solutions (a) and (d) in Figure 2 satisfy the information preservation properties.

For asymmetric merge approaches, we demand the preservation of all concepts, attributes and relationships only for the preferred input ontology into which the other ontology is merged in. Information from the non-preferred input ontology should also be retained but without introducing redundant or conflicting information presentation (see below). For both, symmetric and asymmetric merge we demand that all instance objects of both input ontologies - if present - should be preserved, i.e. every object needs to be assigned (migrated) to a proper concept in the merge result.

Another important requirement is that the merge algorithm should support a good *understandability* and usability of the merge result. This is by necessity a subjective and thus vague requirement. The understandability of the merge result is also strongly dependent on the understandability of the input ontologies, e.g. their size and whether good modeling practices such as the avoidance of unnecessary redundancy have been

followed. In general we believe that it is beneficial for understandability to have compact ontologies and to avoid conflicting or redundant representations of the same information. In particular, we must ensure the property of *equality preservation*, i.e. matching concepts from both input ontologies (as represented in the input mapping) need to be merged to the same concept in the result ontology so that they are represented only once. The merge result thus becomes more compact and less redundant than a simple union of the input ontologies (if there is at least one matching concept). All solutions in Figure 2 satisfy the equality preservation property.

The discussion and example in the introductory section has shown that providing equality preservation and information preservation is not sufficient to achieve a well understandable merge result. In particular, the Full Merge solution (a) in Figure 2 suffers from a redundant and conflicting representation of several concepts that is avoided by symmetric solution (d) as well as the asymmetric solutions (b) and (c). We thus view it as desirable to reduce semantic overlap and redundant / conflicting concept placements. A frequent cause of redundancy is the assignment of a merged concept under several parent concepts leading to multiple root paths to the merged concept (e.g. for concepts *Fiat* and *BMW* in Figure 2(a)). [14] proposes a property to control the semantic overlap in the merge result by requiring that the merge must not lead to additional (is-a) root paths for leaf concepts. This criterion implies that no multiple inheritance is introduced for merging tree-structured input ontologies. The semantic overlap property is not satisfied by the Full Merge result (a) but for solutions (b), (c) and (d) in Figure 2.

3 Quality measures for ontology merging

A benchmark for ontology merging should mainly be able to fairly evaluate the quality of different merge tools by defining appropriate quality metrics and providing suitable test scenarios. It would also be of interest to measure the runtime performance (efficiency) of different algorithms and to evaluate the manual user effort to determine the final merge result. In this section we mainly discuss the first point, metrics for assessing the merge quality. In the next section, we present a series of test scenarios that have been used to evaluate different merge algorithms. We will briefly discuss the user effort involved in the considered tools but leave its quantitative evaluation for future work.

Evaluating the quality of an ontology merging approach is a difficult task for reasons that we have already discussed to some extent. First, it implies measuring the quality of the output ontology which is at least partially subjective, e.g. regarding its understandability. Furthermore, there is a strong dependency on the quality of the input ontologies, e.g. whether they contain errors, whether they are well modelled with little redundancy, how complete they cover a given domain, etc. To keep the evaluation tractable, we only aim at evaluating the quality of the merge result relative to the input ontologies. Ideally, the input ontologies are correct and exhibit no or only little overlap between concepts. We assume that the given input match mapping/alignment is correct although obtaining such a mapping is non-trivial for larger ontologies.

Another main problem to evaluate the quality of a merge approach is that there is generally no unique perfect merge result for comparison (as assumed in [4]). As already discussed there are often several equally suited merge results that can still differ

substantially. Especially for larger ontologies it is also very laborious (and subjective) to specify expert solutions.

To overcome these problems we propose the use of the following simple criteria to evaluate the quality of merge results and thus merge approaches: *coverage*, *compactness* and *redundancy*.

Coverage. The coverage of a merge solution is related to the degree of information preservation and measures the share of input concepts preserved in the result. We can differentiate various sub-cases such as *source coverage* and *target coverage* for the degree to which concepts of the source and target input ontologies are preserved. We may also consider the *leaf coverage* as the degree to which input leaf concepts are preserved. This can be of interest when only leaf concepts carry instance objects so that it becomes important to preserve all leaf concepts to avoid information loss. The *overall coverage* is defined as the arithmetic average of source and target coverage. Coverage values range between 0 and 1 (or 0-100%). Symmetric merge approaches should achieve a coverage of 1, i.e. cover all input concepts. Smaller values than 1 do not necessarily imply an information loss but may be a consequence of avoiding redundancy in the merge result.

In our running example, the symmetric solutions (a) and (d) achieve a source and target coverage of 1. By contrast, the asymmetric, target-driven solution (b) achieves a source coverage of 0.86 (concept *European_Car* is not preserved) and the source-driven solution (c) achieves a target coverage of only 0.67 (2 of 6 concepts are missing: *German_Car* and *Italian_Car*). All solutions have a leaf coverage of 1.

Compactness. This measure checks the size or compactness of the generated merge result which is related to its understandability since a merge solution should not be unnecessarily large. In addition to the *absolute result size* (number of concepts in the result ontology) we determine the *relative result size* compared to the Full Merge result. The size of the Full Merge result can be approximated as the sum of the number of concepts of both input ontologies minus the number of merged concepts (i.e. the number of equality match correspondences in the input mapping). By definition, the relative result size of the Full Merge is 1. Values smaller than 1 indicate an improved compactness, e.g. if some input concepts are dropped to avoid redundancy in the merge result. Values larger than 1 are also possible when the result ontology contains newly introduced concepts, e.g. added by an expert to consolidate differences in the input ontology.

The absolute result size of our running example is 10, 9, 8 and 12 respectively for (a), (b), (c) and (d); the relative result size w.r.t. the full merge solution is 1.0, 0.9, 0.8 and 1.2.

Redundancy. As discussed in Section 2, it is desirable to reduce the degree of redundancy or semantic overlap in an integrated ontology for improved understandability. This can be supported by avoiding the introduction of additional paths from the ontology root to leaf concepts. In addition to the absolute number of leaf paths we use a relative measure to evaluate the degree of semantic redundancy. Let LP_S and LP_T be the number of leaf paths in the source and target input ontology, respectively, and M_L the number of matching leaf concepts in the input match mapping. Then $LP_B = LP_S + LP_T - M_L$ is the least possible number of leaf paths that does not introduce any redundancy. We define the *relative redundancy* in a merge result as the

ratio between the number of leaf paths in the result and LP_B . Values larger than 1 indicate the introduction of redundant paths for merged concepts and thus leaves. This is considered harmful to the understandability of the merge result. A value of 1 is typically optimal as it indicates the successful avoidance of redundant paths. Values smaller than 1 imply that some leaf concepts are not covered in the merge result referring to an information loss and reduced (leaf) coverage.

For our running example, we count 8 different leaf paths in the full merge result (a) and 6 for the other solutions (b), (c) and (d), leading to a relative redundancy of 1.33 for solution (a) and of 1.0 for (b), (c) and (d).

The introduced metrics reflect some of the tradeoffs in achieving the contradicting goals of information preservation and good understandability. While we consider a relative redundancy of 1 as optimal, the best values for coverage and compactness are less clear, i.e. they may be smaller than 1 or (for compactness) even larger than 1. We discuss these issues further in the next section when we present the evaluation results.

4 Experimental Results

We first sketch the algorithms we were able to evaluate and then introduce our test scenarios. We then use the proposed quality measures to evaluate the algorithms for the different scenarios.

4.1 Merge algorithms

While many systems for schema and ontology merging have been proposed and implemented during the last years, we could only find two tools (providing three algorithms) for conducting our evaluation: PROMPT [10] and ATOM [15]. We briefly discuss their characteristics.

PROMPT provides an algorithm for both aligning and merging ontologies. The merging algorithm is semi-automatic since it can perform some tasks automatically, while the other tasks are suggested to the user as a list of possible operations to execute. The main merge approach is symmetric. However, in case of conflicts the system also provides the possibility to manually give preference to one input resulting in a partially asymmetric solution. PROMPT is available as a plug-in for Protégè, an open-source ontology design and editor.

ATOM provides a fully automatic ontology merging based on a given match mapping between the input taxonomies. Two algorithms are provided. First, a symmetric Full Merge as introduced in Section 1 is offered. The main approach is an asymmetric algorithm that preserves one input ontology as the merge target and incrementally extends it with concepts and relationships from the other input ontology. It can either use an equivalence-based match mapping as input or an enhanced match mapping containing equivalence, is-a and inverse is-a relationships between concepts of the input taxonomies.

4.2 Test scenarios for ontology merging

For our evaluation, we use five test cases of different size and complexity. We consider three hand-crafted small-sized scenarios for which we can also manually provide a merge solution. We additionally evaluate two larger real-life merge problems.

- **Cars:** This test case is our running example presented in Section 1. This scenario merges two taxonomies classifying European cars in different ways.
- **Computers:** Two small product catalogs of different computer and hardware shops are to be merged. One input ontology classifies products first by manufacturer (e.g., *Dell* or *HP*) and then by product type (e.g., *Laptops*, *Accessories*, etc.) while the second input uses the opposite order.
- **Anatomy Subset:** This is a small subset of the fourth scenario (*Anatomy*) and includes multiple inheritance. It merges part of the subgraph describing "*Eye Muscle*" in the Mouse Anatomy to the subgraph describing an equivalent concept in the NCI Thesaurus.
- **Anatomy:** *Anatomy* is the full match scenario with multiple inheritance proposed by the OAEI [1] that merges the Adult-Mouse Anatomy (over 2,700 concepts) with the anatomical part of the NCI Thesaurus (NCIT) (about 3,300 concepts).
- **eBay:** eBay scenario merges different versions of the eBay product catalog containing on average more than 22,000 concepts organized in a tree structure.

4.3 Evaluation of test scenarios

We used PROMPT and ATOM to solve the introduced test scenarios. Since ATOM is fully automatic while PROMPT strongly requires user interaction to generate the integrated ontology, we have not compared the user effort needed to build the final result and we simulate an automatic algorithm with PROMPT manually discarding all wrong suggestions. Except for large size scenarios we were able to generate a full merge solution using PROMPT. On the other side, ATOM implements both a full merge and an asymmetric target-driven algorithm; a source-driven merge solution can be easily generated by changing the source and the target in the input of the algorithm. In addition to the full merge solution we will also evaluate the source-driven and target-driven solutions produced by ATOM. For the three small-sized scenarios we also determined manually an expert solution for comparison.

We summarize the coverage, compactness and redundancy results of our experiments in Table 1. We show results for three mentioned automatic merge approaches; an expert solution could only be evaluated for the first three scenarios. We observe that the symmetric full merge and the expert solutions always achieve a complete coverage of 1. The asymmetric solutions achieve a complete coverage only for their preferred input ontology but do not include some concepts from the other ontology to avoid introducing redundancy. As a result their overall coverage is somewhat reduced, albeit merely by 1-2% for the real-life scenarios *Anatomy* and *eBay*. All determined merge solutions preserve all leaf concepts that typically carry instance objects.

About the compactness, we note that the symmetric merge solutions determine the largest result ontology in all scenarios. In particular, for our *Cars* scenario we had additional concepts in the merge result leading to a compactness of 1.2. Similar than for

		Coverage				Compactness		Redundancy	
		Source	Target	Leaf	Overall	Abs. Size	Rel. Size	# Leaf Paths	Relative
Cars	Full Merge	1	1	1	1	10	(1)	8	1.33
	Source-driven	1	0.67	1	0.83	9	0.9	6	1
	Target-driven	0.86	1	1	0.93	8	0.8	6	1
	Expert Result	1	1	1	1	12	1.2	6	1
Computers	Full Merge	1	1	1	1	20	(1)	13	1.3
	Source-driven	1	0.86	1	0.93	18	0.9	10	1
	Target-driven	0.9	1	1	0.95	19	0.95	10	1
	Expert Result	1	1	1	1	20	1	10	1
AnatomySubset	Full Merge	1	1	1	1	9	(1)	7	2.33
	Source-driven	1	0.83	1	0.91	8	0.89	3	1
	Target-driven	0.33	1	1	0.66	8	0.89	3	1
	Expert Result	1	1	1	1	9	1	3	1
Anatomy	Full Merge	1	1	1	1	4.5K	(1)	14K	1.94
	Source-driven	1	0.97	1	0.98	4.4K	0.98	7.2K	1
	Target-driven	0.96	1	1	0.98	4.4K	0.98	7.2K	1
eBay	Full Merge	1	1	1	1	23.3K	(1)	21.5K	1.05
	Source-driven	1	0.99	1	0.99	23.2K	0.99	20.4K	1
	Target-driven	0.99	1	1	0.99	23.2K	0.99	20.4K	1

Table 1. Summary of Experiments

coverage, the asymmetric solutions achieve smaller values than 1 albeit merely by 1-2% for the real-life scenarios *Anatomy* and *eBay*.

More significant differences result for the redundancy evaluation which is primarily influenced by the number of paths to leafs and thus to how concepts are organized and connected within ontologies. In Table 1 we report both the absolute number of leaf paths for the merge solutions as well as their relative redundancy. We observe that the Full Merge approach produces a substantial amount of redundancy and semantic overlap even for the real-life test cases. The relative redundancy is especially severe (factor 1.94 and 2.33) for the two anatomy scenarios exhibiting multiple inheritance. By contrast, the manually determined expert solutions avoid redundancy altogether although they cover both input ontologies completely and may include additional concepts (as for the first scenario). The asymmetric solutions of ATOM are also able to achieve the optimal relative redundancy of 1, but in a fully automatic way and even for the large real-life scenarios.

5 Conclusions

We proposed the use of general metrics to evaluate the quality of ontology merging solutions such as the relative coverage of the input ontologies, the compactness of the

merge result and the degree of introduced redundancy. We found these metrics useful to evaluate different symmetric and asymmetric merge approaches for diverse test cases of different size and complexity. We observed that straight-forward symmetric merge approaches can achieve a complete coverage (information preservation) of the input ontologies but are generally too large due to the introduction of a significant semantic overlap and redundant paths, e.g. when compared to manually specified expert solutions. We thus see a need for improved symmetric merge approaches that achieve automatically a reduced degree of redundancy. Asymmetric merge approaches completely cover (preserve) only one of the input ontologies as useful for many applications. The evaluation showed that these approaches only miss 1-2% of the input concepts in real-life scenarios but achieve a perfect relative redundancy. This means that they successfully avoid introducing semantic overlap in the merge result and thus improve its understandability.

References

1. The Ontology Alignment Evaluation Initiative. <http://oei.ontologymatching.org>.
2. B. Alexe, W. Tan, and Y. Velegrakis. STBenchmark: Towards a Benchmark for Mapping Systems. *Proc. of the VLDB Endowment*, 1(1):230–244, 2008.
3. Z. Bellahsene, A. Bonifati, and E. Rahm, editors. *Schema Matching and Mapping*. Springer, 2011.
4. F. Duchateau and Z. Bellahsene. Measuring the Quality of an Integrated Schema. In *ER*, pages 261–273, 2010.
5. F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-Match: an Algorithm and an Implementation of Semantic Matching. In *ESWS*, pages 61–75, 2004.
6. A. Guzmán-Arenas and A.-D. Cuevas-Rasgado. Knowledge Accumulation through Automatic Merging of Ontologies. *Expert Syst. Appl.*, 37(3):1991–2005, 2010.
7. K. Kotis and G. A. Vouros. The HCONE Approach to Ontology Merging. In *ESWS*, pages 137–151, 2004.
8. P. Lambrix and H. Tan. SAMBO-A System for Aligning and Merging Biomedical Ontologies. In *Journal of Web Semantics*, volume 4, pages 196–206, Sept. 2006.
9. D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An Environment for Merging and Testing Large Ontologies. In *KR*, pages 483–493, 2000.
10. N. F. Noy and M. A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *AAAI/IAAI*, pages 450–455, 2000.
11. R. A. Pottinger and P. A. Bernstein. Merging Models Based on Given Correspondences. In *VLDB*, pages 862–873, 2003.
12. A. Radwan, L. Popa, I. R. Stanoi, and A. A. Younis. Top-K Generation of Integrated Schemas Based on Directed and Weighted Correspondences. In *SIGMOD*, pages 641–654, 2009.
13. E. Rahm. Towards Large Scale Schema and Ontology Matching. In *Schema Matching and Mapping*, chapter 1, pages 3–27. Springer, 2011.
14. S. Raunich and E. Rahm. Target-driven Merging of Taxonomies. Technical report, University of Leipzig, 2010.
15. S. Raunich and E. Rahm. ATOM: Automatic Target-driven Ontology Merging. In *Proc. of ICDE*, 2011.
16. K. Saleem, Z. Bellahsene, and E. Hunt. PORSCHE: Performance ORiented SCHEMA mediation. *Inf. Syst.*, 33(7-8):637–657, 2008.
17. D. Thau, S. Bowers, and B. Ludäscher. Merging Taxonomies under RCC-5 Algebraic Articulations. In *ONISW*, pages 47–54, 2008.