Results of GeRoMeSuite for OAEI 2009

Christoph Quix, Sandra Geisler, David Kensche, Xiang Li

Informatik 5 (Information Systems)
RWTH Aachen University, Germany
http://www.dbis.rwth-aachen.de

Abstract. *GeRoMeSuite* is a generic model management system which provides several functions for managing complex data models, such as schema integration, definition and execution of schema mappings, model transformation, and matching. The system uses the generic metamodel *GeRoMe* for representing models, and because of this, it is able to deal with models in various modeling languages such as XML Schema, OWL, ER, and relational schemas.
A component for schema matching and ontology alignment is also part of the system. After the first participation in OAEI last year, and having established the basic infrastructure for the evaluation, we could focus this year on the improvement of the matching system. Among others, we implemented several new match strategies, such as an instance matcher and a validation method for alignments.

1 Presentation of the system

Manipulation of models and mappings is a common task in the design and development of information systems. Research in Model Management aims at supporting these tasks by providing a set of operators to manipulate models and mappings. As a framework, *GeRoMeSuite* [6] provides an environment to simplify the implementation of model management operators. *GeRoMeSuite* is based on the generic role based metamodel *GeRoMe* [5], which represents models from different modeling languages (such as XML Schema, OWL, SQL) in a generic way. Thereby, the management of models in a polymorphic fashion is enabled, i.e. the same operator implementations are used regardless of the original modeling language of the schemas. In addition to providing a framework for model management, *GeRoMeSuite* implements several fundamental operators such as Match [11], Merge [10], and Compose [8].

The matching component of *GeRoMeSuite* has been described in more detail in [11], where we present and discuss in particular the results for heterogeneous matching tasks (e.g. matching XML Schema and OWL ontologies). An overview of the complete *GeRoMeSuite* system is given in [6].

1.1 State, purpose, general statement

As a generic model management tool, *GeRoMeSuite* provides several matchers which can be used for matching models in general, i.e. our tool is not restricted to a particular domain or modeling language. Therefore, the tool provides several well known matching strategies, such as string matchers, Similarity Flooding [9], children and parent matchers, matchers using WordNet, etc. In order to enable the flexible combination of these basic matching technologies, matching strategies combining several matchers can be configured in a graphical user interface.

Because of its generic approach, *GeRoMeSuite* is well suited for matching tasks across heterogeneous modeling languages, such as matching XML Schema with OWL. We discussed in [11] that the use of a generic metamodel, which represents the semantics of the models to be matched in detail, is more advantageous for such heterogeneous matching tasks than a simple graph representation.

Furthermore, *GeRoMeSuite* is a holistic model management and not limited to schema matching or ontology alignment. It supports also other model management tasks such as schema integration [10], model transformation [4], mapping execution and composition [7, 8].

1.2 Specific techniques used

The basis of *GeRoMeSuite* is the representation of models (including ontologies) in the generic metamodel *GeRoMe*. Any kind of model is transformed first into the generic representation, then the model management operators can be applied to the generic representation. The main advantage of this approach is that operators have to be implemented only once for the generic representation. In contrast to other (matching) approaches which use a graph representation without detailed semantics, our approach is based on the semantically rich metamodel *GeRoMe* which is able to represent modeling features in detail.

For the OAEI campaign, we focused on improving our matchers for the special case of ontology alignment, e.g. we added some features which are useful for matching ontologies. For example, the generic representation of models allows the traversal of models in several different ways. During the tests with the OAEI tasks, we realized that, in contrast to other modeling languages, traversing the ontologies using another structure than class hierarchy is not beneficial. Therefore, we configured most of our matchers that take the model structure into account just to work with the class hierarchy. Furthermore, we implemented so called 'children' and 'parent' matchers, which propagate the similarity of elements up and down in the class hierarchy.

For OAEI 2009, we added also an *Instance Matcher*, which uses instances to determine the similarity of classes and properties. Due to the flexibility and extensibility of our matching framework, the implementation of an additional matcher can be done with only a few lines of code. Basically, we just need to choose a traversal strategy which includes instances, apply one of the existing string matchers, and then choose an appropriate structural matcher to propagate the similarity of the instances to classes and properties.

In addition to last year, we also experimented with another string matcher which based on the SecondString library (`http://secondstring.sourceforge.net/`, [1]). The library provides several different string distance metrics which can be combined in various ways. The combination of 'soft' tokenization, TF-IDF based weighting of tokens, and the classical Jaro/Winkler string metric (called Soft-TFIDF) has shown good results in string matching tasks [1]. However, for the benchmarks track, we did not find any significant difference to the string metric of [12]. For other ('real') matching tasks, the use of Soft-TFIDF might be beneficial, but we have to evaluate this with further tests.

Furthermore, we implemented a validation method using similar methods as ASMOV [3]. For difficult matching tasks with initially low values for precision and recall, the validation could increase the quality of the results by 10-20%. It is obvious, that for easy matching tasks, such as the 10x tasks in the benchmark track, the improvement cannot be so large. However, also in these tests the validation helped to achieve a perfect result.

1.3 Adaptations made for the evaluation

As only one configuration can be used for all matching tasks, we worked on strategies for measuring the quality of an alignment without having a reference alignment. We compared several statistical measures (such as expected value, variance, etc.) of alignments with different qualities in order to identify a 'good' alignment. Furthermore, these values can be used to set thresholds automatically.

During the tests, we made the experience that the expected value of all similarities, the standard deviation, and the number of mappings per model element can be used to evaluate the quality of an alignment.

Fig. 1 indicates the strategy which we used for the matching tasks in the benchmark track. All aggregation and filter steps use variable weights and thresholds, which are based on the statistical values of the input similarities.

The role matcher is a special matcher which compares the roles of model elements in our generic role-based metamodel. In principle, this results in that only elements of the same type are matched, e.g. classes with classes only and properties with properties only.

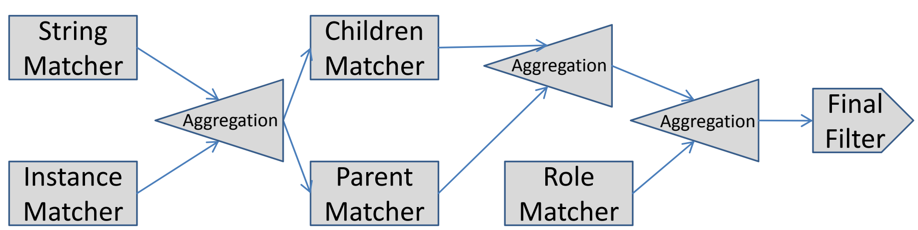

Fig. 1. Matching Strategy for OAEI

In contrast to last year, we removed Similarity Flooding [9], as it had no positive effect on the match quality. Structural similarity is already taken into account by the children and parent matchers; an additional structural matcher seems to blur the results.

On a technical level, we implemented a command line interface for the matching component, as the matching component is normally used from within the GUI framework of *GeRoMeSuite*. The command line interface can work in a batch modus in which several matching tasks and configurations can be processed and compared.

1.4 Link to the system and parameters file

More information about the system can be found on the homepage of *GeRoMeSuite*: `http://www.dbis.rwth-aachen.de/gerome/oaei2009/`

The page provides also links to the configuration files used for the evaluation.

1.5 Link to the set of provided alignments (in align format)

The results for the OAEI campaign 2008 are available at `http://www.dbis.rwth-aachen.de/gerome/oaei2009/`

2 Results

2.1 Benchmark

At the cost of some performance (matching now takes about 15-25 seconds for each task instead of 5-15 as last year), our results have been significantly improved in 2009 for the benchmark track.

Overall, our matching component achieved very similar values for precision and recall, which seems to be rather unusual, if we compare our results with the results of other systems for previous years, where the precision was usually higher than recall.

Tasks 101-104 In all these very basic tasks, we achieved the perfect result.

Task	Precision	Recall 08
101	1,00	1,00
103	1,00	1,00
104	1,00	1,00

Tasks 201-210 In these tasks, the linguistic information could not always be used as labels or comments were missing. If no labels and comments are available, instance information might still help to find the right matches. We included an instance matcher in our configuration this year, which resulted in significant improvement for the 202 test cases.

Task	Precision	Recall
201	0,92	0,98
201-2	1,00	1,00
201-4	0,98	0,99
201-6	0,98	0,98
201-8	0,96	0,98
202	0,64	0,38
202-2	0,99	0,90
202-4	0,94	0,78
202-6	0,94	0,62
202-8	0,79	0,49
203	1,00	1,00
204	1,00	1,00
205	1,00	0,97
206	0,94	0,97
207	0,94	0,97
208	1,00	1,00
209	0,81	0,61
210	0,66	0,81

2.2 Tasks 221-231

The ontologies in these tasks lacked some structural information. As our matcher still uses string similarity in a first step, the results were perfect except for the case 223 for which we missed one match.

Task	Precision	Recall
221	1,00	1,00
222	1,00	1,00
223	0,99	0,99
224	1,00	1,00
225	1,00	1,00
228	1,00	1,00
230	1,00	1,00
231	1,00	1,00

Tasks 232-266 These tasks are some combinations of the tasks before. For most of the tasks, the performance of our matcher was much better than last year. However, for some matching tasks (e.g. 257, 262, 265, and 266), our system produced no result. Unfortunately, we could not resolve this problem before the deadline.

3 Comments

We participate this time the second time in OAEI and see a significant improvement of our matcher compared to last year. Thus, a structured evaluation and comparison of ontology alignment and schema matching components as OAEI is very useful for the development of such technologies.

However, mappings between models are constructed for various reasons which can result in very different mapping results. For example, mappings for schema integration may differ from mappings for data translation. Therefore, different semantics for ontology alignments should be taken into account in the future, as it has been pointed out for schema matching in [2].

4 Conclusion

As our tool is neither specialized on ontologies nor limited to the matching task, we did not expect to deliver very good results. However, we are very satisfied with the overall results, especially compared to last year.

We will continue to work on the improvement of our matching system, especially taking into account additional validation methods, a clustering approach to handle scalability issues, and automatic methods for tuning and configuration of schema matchers. We hope to participate again with an improved system in the OAEI campaign next year.
Acknowledgements: This work is supported by the DFG Research Cluster on Ultra High-Speed Mobile Information and Communication (UMIC, `http://www.umic.rwth-aachen.de`) and by the Umbrella Cooperation Programme (`http://www.umbrella-coop.org/`).

References

1. W. W. Cohen, P. D. Ravikumar, S. E. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. S. Kambhampati, C. A. Knoblock (eds.), *Proc. IJCAI Workshop on Information Integration on the Web (IIWeb)*, pp. 73–78. 2003.
2. J. Evermann. Theories of Meaning in Schema Matching: A Review. *Journal of Database Management*, **19**(3):55–82, 2008.
3. Y. R. Jean-Mary, E. P. Shironoshita, M. R. Kabuka. Ontology matching with semantic verification. *Journal of Web Semantics*, **7**(3):235–251, 2009.
4. D. Kensche, C. Quix. Transformation of Models in(to) a Generic Metamodel. *Proc. BTW Workshop on Model and Metadata Management*, pp. 4–15. 2007.
5. D. Kensche, C. Quix, M. A. Chatti, M. Jarke. *GeRoMe*: A Generic Role Based Metamodel for Model Management. *Journal on Data Semantics*, **VIII**:82–117, 2007.
6. D. Kensche, C. Quix, X. Li, Y. Li. *GeRoMeSuite*: A System for Holistic Generic Model Management. C. Koch, J. Gehrke, M. N. Garofalakis, D. Srivastava, K. Aberer, A. Deshpande, D. Florescu, C. Y. Chan, V. Ganti, C.-C. Kanne, W. Klas, E. J. Neuhold (eds.), *Proceedings 33rd Intl. Conf. on Very Large Data Bases (VLDB)*, pp. 1322–1325. Vienna, Austria, 2007.
7. D. Kensche, C. Quix, X. Li, Y. Li, M. Jarke. Generic Schema Mappings for Composition and Query Answering. *Data and Knowledge Engineering*, 2009. To appear.
8. D. Kensche, C. Quix, Y. Li, M. Jarke. Generic Schema Mappings. *Proc. 26th Intl. Conf. on Conceptual Modeling (ER'07)*, pp. 132–148. 2007.
9. S. Melnik, H. Garcia-Molina, E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. *Proceedings of the 18th International Conference on Data Engineering (ICDE)*, pp. 117–128. IEEE Computer Society, San Jose, CA, 2002.

10. C. Quix, D. Kensche, X. Li. Generic Schema Merging. J. Krogstie, A. Opdahl, G. Sindre (eds.), *Proc. 19th Intl. Conf. on Advanced Information Systems Engineering (CAiSE'07)*, LNCS, pp. 127–141. Springer-Verlag, 2007.
11. C. Quix, D. Kensche, X. Li. Matching of Ontologies with XML Schemas using a Generic Metamodel. *Proc. Intl. Conf. Ontologies, DataBases, and Applications of Semantics (ODBASE)*, pp. 1081–1098. 2007.
12. G. Stoilos, G. B. Stamou, S. D. Kollias. A String Metric for Ontology Alignment. Y. Gil, E. Motta, V. R. Benjamins, M. A. Musen (eds.), *Proc. 4th International Semantic Web Conference (ISWC)*, *Lecture Notes in Computer Science*, vol. 3729, pp. 624–637. Springer, 2005.