A High-level Architecture of a Metadata-based Ontology Matching Framework

Malgorzata Mochol
Freie Universität Berlin, Institut für Informatik
Takustr. 9, D-14195 Berlin, Germany
mochol@inf.fu-berlin.de

Elena Paslaru Bontas Simperl
Freie Universität Berlin, Institut für Informatik
Takustr. 9, D-14195 Berlin, Germany
paslaru@inf.fu-berlin.de

Abstract

One of the pre-requisites for the realization of the Semantic Web vision are matching techniques which are capable of handling the open, dynamic and heterogeneous nature of the semantic data in a feasible way. Currently this issue is not being optimally resolved; the majority of existing approaches to ontology matching are (implicitly) restricted to processing particular classes of ontologies and thus unable to guarantee a predictable result quality on arbitrary inputs. Accounting for the empirical findings of two case studies in ontology engineering, we argue that a possible solution to cope with this situation is to design a matching strategy which strives for an optimization of the matching process whilst being aware of the inherent dependencies between algorithms and the types of ontologies these are able to process successfully. *We introduce a matching framework that, given a set of ontologies to be matched described by ontology metadata, takes into account the capabilities of existing matching algorithms (matcher metadata) and suggests, by using a set of rules, appropriate ones.*

1 Introduction

Semantic technologies provide a standardized infrastructure for pervasively creating, using and exchanging machine-understandable information. One of the pre-requisites for the take-up of these technologies at Web scale are *matching methods* capable of handling the open, dynamic and heterogeneous nature of Semantic Web information in a feasible way. Semantic Web-compatible matching methods are expected to satisfy two core requirements. First, they should be applicable by ontology engineers which create an application ontology as a combination of existing knowledge resources. As ontology development is definitely not targeted at people with a high level of expertise in Computer Science there is a need for means aiding them in selecting and applying ontology management tools, including *matching algorithms*. Second, matching methods are acknowledged as a core enabling technology for mediating Web Service interactions. Under these circumstances the selection of the most appropriate matching service is envisioned to be performed automatically.

Matching conceptual structures, be that databases, XML schemes, conceptual graphs or, more recently, Semantic Web ontologies, is fundamental in areas such as data integration, data warehouses or information retrieval. The significance of this research topic is best reflected by the high number of matching algorithms (*matchers*) proposed in the literature (e.g. [5, 8, 9, 15, 17]). However, despite of the impressive number of research initiatives in the field, and despite of the diversity of ideas and techniques employed, current matching approaches still show important limitations when applied to the emerging Semantic Web. Their usage in arbitrary application settings causes considerable manual customization efforts (e.g. GLUE, COMA[2]). Some approaches require particular representation and natural languages[3] or do not perform well on inputs with heterogeneous (graph) structures (e.g. Cupid[6]) while others are restricted to tree-based conceptual models (e.g. S-Match[6]) or require human intervention to compute accurate matches (e.g. Similarity Flooding[10]). Combining multiple matchers still does not overcome all these deficiencies: the quality of the so-called meta-learner approaches (e.g. LSD[4]) is directly proportional to the efforts invested in training the algorithms[18]. Furthermore, black box solutions such as COMA[3] prove to be unable to adapt well to complex application scenarios due to the inflexibility of the built-in matching combinations. Since each base matcher performs differently under different circumstances simple, pre-defined composition methods are incapable of capturing such performance variation[18].

This survey provides strong indication that the question of building a single overarching matching algorithm capable of efficiently handling arbitrary ontologies can not be solved adequately in the near future. This finding is confirmed by many case studies in the literature aiming at comparing, merging or integrating ontologies with the help of current technologies. Experiences gained in real-world case studies

in ontology engineering[13] indicate that a possible solution to this dilemma is to design a new matching strategy (as opposed to a combined matching heuristic) which strives for an optimization of the matching process whilst being aware of the inherent dependencies between matching algorithms and the types of ontologies they are able to process *successfully*.

Section 2 introduces the **M**etada-based **O**ntology **MA**tching (MOMA) framework which resorts to semantically represented metadata on ontologies and matchers in order to express a set of rules that can be applied to detect algorithms suitable for processing a pre-defined ontological input. The approach is evaluated in the context of two case studies in the human resource and medical domain (Sections 3 and 4, respectively). In Section 5 we conclude the present results and sketch planed research.

2 MOMA Framework

The MOMA framework uses additional information about the ontologies (*ontology metadata*, Sec. 2.1.2) and available matching services (*matcher metadata*, Sec. 2.1.1) in order to determine which of the latter are appropriate in a given application context. The ontology metadata captures information about matching-relevant ontology features such as the size of the model or the language used for labelling ontological primitives. In turn the matcher metadata describes the most important characteristics of the matching services: input and output parameters, applied heuristics etc. The core of the MOMA framework consists of a *selection engine*, responsible for the decision making process by means of rules grouped in a *rule repository*, and an *execution engine* responsible for completing the matching task (Fig. 1).

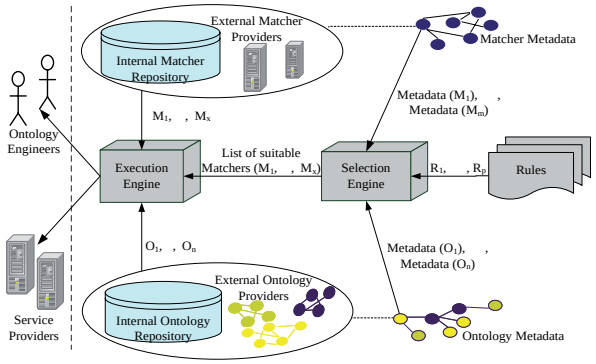

Figure 1. MOMA Framework

We foresee two usage patterns the proposed framework: data and service providers are able to systematically publish their resources (ontologies or new matching algorithms) Web-wide with the help of the MOMA framework. In doing so they are expected to provide the required descriptive metadata for the subscribed resources, as this guarantees a higher visibility of their products as regarding incoming inquiries. On the other hand, matching consumers consult the MOMA framework in order to get (information about) matching algorithms adequate for a particular task. This applies for both humans, ontology engineers looking for means to compare similar ontological sources, and Web Services seeking for automatized methods to generate mediation ontologies.

2.1 The Metadata

In order to ensure a rich and formal representation of the ambiguous semantics of the metadata, and to enable its integration and exchange in and between Semantic Web applications we modelled this information in ontological form and implemented it using Semantic Web representation languages. The ontologies were developed in accordance with established ontology engineering methodologies, while the empirical findings acquired during the case studies[13] were the main input for the elaboration of the requirements underlying the metadata schemes (cf. Sec. 3 and 4).

2.1.1 The Matcher Metadata

The matcher metadata[1] captures information about ontology matchers [12]. In order to specify the contents of the target metadata model we compiled a list of matcher features which are empirically proved to have an impact on the quality of matching tasks. For the classification of the algorithms we currently rely on [16] which makes the distinction between *individual* and *combining* matchers. The former might work on instance data (*instance/contents-based*) or restrict to schema information (*schema-only based*). Both might be applied to individual schema elements, such as properties or concept labels (*element-level*). Schema-only based approaches can deal with combinations of schema elements (complex schema structures), thus allowing for the computation of mappings by analyzing sub-graphs (*structure-level*). An element-level matcher uses *linguistic* as well as *constraint-based* techniques, while a schema-only based matcher applies only the latter. *Combining matchers* are divided into two categories: *composite matchers* (e.g. GLUE[5],COMA[3], CMC[18]) which combine the different results of independently executed matchers and *hybrid matchers* (e.g. Cupid[8]) where different matching dimensions are used within a single algorithm. Beside the mentioned classification, the metadata model includes matcher characteristics such as input type, matching

[1] Available at `http://nbi.inf.fu-berlin.de/research/wissensnetze/matching/matchingmetadata.owl`.

level (atomic and non-atomic level) or cardinality[11].The evaluation of metadata was performed by means of an in-situ experiment in which the resulting model was compared against the requirements derived from the case studies.

2.1.2 The Ontology Metadata

The matcher inputs (i.e. ontologies) are described using the metadata model introduced in[7] which can be applied to describe ontologies in various phases of their life-cycle.[2] Accounting for the fact that matching algorithms cannot be applied with the same success expectations regardless of any dimension of the ontology metadata model we have identified the following ontology features as relevant:
•*syntactic features* such as the *number of specific ontological primitives* that affect the matching execution performance and quality of the structured-based matchers that typically perform better on simple graph structures.
•*semantic features* such as modelled domain, representation and natural language, level of formality, domain generality that restrict the number of applicable matching algorithms, which might be adequate for a sub-set of these features.
These characteristics are referenced in the matcher metadata model to refine the description of the matching inputs.

2.2 The Matching Rules

For a given set of ontologies to be matched the selection engine must decide which matching algorithms are applicable in order to obtain the desired outputs. The engine is aware of the background information detailing the available matching services and the properties of the input ontologies. However, in order to *automatically infer* which algorithms suit to certain inputs it needs explicit knowledge regarding the dependencies between these algorithms and the structures on which they operate. We formalize this knowledge into (domain independent) *dependency rules*. Some of them are:
1. Match only ontologies in similar domains
2. Match upper-level to domain ontologies using linguistic matchers
3. Use only linguistic matchers for informal ontologies
4. Use structure-based matchers for ontologies with different natural languages
5. Do not apply linguistic matchers to ontologies with incompatible concept names
6. Use constraints-based matchers only for formal ontologies and only if ontologies contain axioms
7. Apply only scheme matchers if no instance data available
8. Apply only instance matchers to a single ontology
9. Apply only matchers which are capable of dealing with the representation language of the inputs
The rules came as a result of analyzing recent publications in this research discipline and were confirmed empirically within the domains of human resource and medicine (cf. Sec. 3 and 4, respectively)[3]. We are currently experimenting with dependency rules addressing performance and scalability issues and with user-driven methods which allow an extension or refinement of the rule set employed.

3 Case Study: Human Resources

The "Knowledge Nets" project[4] explores the potential of the Semantic Web from a business and technical viewpoint by examining the effects of the deployment of semantic technologies on particular application scenarios and market sectors. For this purpose, we built a use case scenario for the recruitment domain (HR scenario) in which we analyzed the online process of seeking and procuring jobs[1]. In the first step we built an OWL ontology to define occupations, skills and industrial background in the process of data exchange between employers, applicants and job portals. To pinpoint the appropriate job for an applicant or a suitable candidate for a job opening we needed semantic matching approaches which can deal with the highly formal HR-ontology and with the specific application requirements. To support the decision regarding the selection of a suitable matching approach, we applied the MOMA framework. As we are dealing with a single ontology the 1^{st} rule is satisfied but since we are not working with multiple inputs the 2^{nd}, 4^{th} and 5^{th} rules cause no reduction in the number of potentially suitable matchers. The 1^{st} rule is satisfied, as we are dealing with a single ontology domain. Since we are not working with multiple inputs, the 2^{nd}, 4^{th} and 5^{th} rules cause no reduction in the number of potentially suitable matchers. Furthermore, the 3^{rd} rule which refers to the informal and semi-formal ontologies and 7^{th} rule which addresses ontologies without instances are also irrelevant. The HR-ontology is implemented in a formal representation language and includes instances. Since the ontology does not contain any axioms the constraint-based matchers are eliminated from the list of possible matcher candidates (6^{th} rule). After applying the 8^{th} and 9^{th} rule we isolate from the list of possible matching algorithms those *instance matchers* (except constraint-based matchers) which deal with the representation language of our ontology.

[2] A complete description of the ontology metadata model is out of the scope of this paper. Model available at `http://swpatho.ag-nbi.de/context/meta.owl` and at `http://omv.ontoware.org`.

[3] Refer to[12] for further details on the rules and their execution.

[4] `http://nbi.inf.fu-berlin.de/research/wissensnetze`

3.1 Lessons Learned

Applying the dependency rules to the HR scenario resulted in restriction of the potentially useful matchers to those being able to handle populated ontologies represented in OWL DL. In the implementation of the semantic job portal we used a hybrid approach which actually deploys instance matchers but in combination with e.g. substring similarity matcher. This measures the similarity of concept labels based on common substrings and edit distances, and belongs to the category of schema-based linguistic matchers. To provide more precise answers to the question of which matching approach is most suitable to a given problem we need to differentiate between combined and individual matchers not only on the metadata level but also by rule definition. Since job portals must automatically compare a particular applicant profile against a multitude of job openings only automatic matching approaches which support a 1:n match cardinality can be applied in this case. Though such requirements could be expressed using the matcher metadata they are currently not covered by our rules set yet.

4. Case Study: Medicine

The project "A Semantic Web for Pathology"[5] analyzes the usage of semantic technologies in a retrieval system for image and text data in the medical domain. A Semantic Web ontology is used to enable content-based retrieval and to guide the automatic semantic annotation of textual pathology reports [14]. Due to the complexity of the application domain and interoperability considerations the ontology engineering process focused on reusing the multitude of medical ontologies instead of modelling the application knowledge from scratch. Ontology matching techniques played a crucial role for the completion of this task as we had to merge various ontologies modelling interrelated domains to the final application ontology. The ontology engineering process was operated as follows. We first identified and analyzed over 100 medical ontologies that covered aspects related to our application domain *lung pathology*. The result of this phase was a list of potentially relevant knowledge resources, which, however, differed to a large extent w.r.t. the granularity of the conceptualization, the level of formality, the implementation language, etc.: i) SNOMED and DigitalAnatomist[6] describe the anatomy of the lung as well as typical diseases and are aligned to UMLS;[7] ii) UMLS Semantic Network contains generic and core medical concepts as part of UMLS; iii) XML-HL7 is a standard, XML-based format for the representation of patient records; and iv) Immunohistology Guidelines are used by domain experts in diagnosis procedures in the medical organization involved in the project. In a second step medical ontologies like SNOMED and Digital Anatomist had to be tailored to the particular needs of our restricted application domain: lung pathology. For this purpose domain experts identified 4 central concepts ("lung", "pleura", "trachea" and "bronchia") and extracted similar or related concepts from the two ontologies (i.e. concepts which are connected by any type of relationship with the core concepts). The result was a total set of approx. 1000 concepts describing the anatomy of the lung and lung diseases. As these concepts are classified according to the upper-level UMLS Semantic Network the latter was also integrated to the final ontology. Candidate relationships were extracted from Digital Anatomist, SNOMED and UMLS Semantic Network. From approx. 50 relations evaluated by domain experts about 20 generic or medicine-specific core relations are to be included to the target ontology. The conceptualization phase was followed by the translation of the UMLS data to OWL [14]. In parallel to the reuse activities, large parts of pathology-specific knowledge were conceptualized using a text-based ontology learning approach as this knowledge was not covered by any of source ontologies. In order to integrate these alternative development outcomes we were confronted with the problem of choosing a matching approach able to deal with the size and the complexity of the resulting ontologies: O1) an ontology of lung anatomy and diseases obtained from pruning SNOMED and DigitalAnatomist; O2) the UMLS Semantic Network; O3) a customized version of the XML-HL7 scheme used to describe the structure and content of the application data (textual pathology reports); O4) a manually developed ontology of immunohistology; and O5) a lung pathology ontology extracted semi-automatically from medical texts. After generating the metadata capturing the matching-relevant features of the mentioned ontologies we examined the constraints imposed by these features on the selection of a suitable matcher. As stated by the 1^{st} rule the involved ontologies shared many commonalities w.r.t. their domain. The 9^{th} rule restricts the set of potentially applicable algorithms to these handling XML schemes and OWL/RDFS. Further on, the algorithms were not required to match instance data. The upper-level ontologies UMLS Semantic Network and XML-HL7 could be matched on a linguistic basis with the remaining ones, which were more domain-specific (2^{nd}). As stated by 4^{th} rule, the integrated SNOMED and DigitalAnatomist ontologies could not be linguistically matched to the pathology-specific sources (O4 and O5) due to different natural languages. The remaining rules were out of the scope of the application setting. As a result of these examinations, we applied a structure-based matcher to merge the domain ontologies O1, O4 and O5.

[5] http://swpatho.ag-nbi.de

[6] http://www.snomed.org, http://www.digitalanatomist.com

[7] http://www.nlm.nih.gov/research/umls

The English-labelled domain ontology O1 was matched against the UMLS Semantic Network and the XML scheme.

4.1 Lessons Learned

The proposed matching strategy enabled domain experts *with poor expertise in Ontological Engineering* a rapid understanding of the ontology matching process. Applying the rules resulted in a restriction of the search space to structure-based matchers and English-oriented linguistic similarity measures to be applied on two groups of ontologies (domain ontologies, and English-labelled ontologies, respectively). However, while the rules definitely speeded-up the selection of the suitable matching candidates, the remaining candidates could not be applied to the application setting without human-driven interventions. As no matcher was found to deal with properties (contained in all of the six sources) these had to be added manually to the final result. Furthermore, the majority of the algorithms returned similar ontological concepts, but no means to merge them. A third problem was related to performance and scalability, as the size of the medical ontologies (>1000 concepts) imposed serious problems. This is an indication for the need of rules relating aspects like the size or the complexity of the ontology to matching services.

5 Conclusion and Future Work

With this paper we present the MOMA framework which is characterized by the usage of the metadata for ontologies and matchers and by the application of prescribed rules to determine which matchers are suitable for individual cases. Even the first evaluation of our framework within the context of HR and medicine use cases (more formal evaluation follows) reveals that matcher candidates can be selected through the framework application, but further work on rules, especially w.r.t. the syntactic features of ontologies with specific performance and accuracy parameters, and refinement of the metadata are required. Further on, as the MOMA framework is also targeted at automatized matching tasks, it should be extended with feasible means for an automatic rule execution and for the computation of similar ontology domains.

References

[1] C. Bizer, R. Heese, M. Mochol, R. Oldakowski, R. Tolksdorf, and R. Eckstein. The Impact of Semantic Web Technologies on Job Recruitment Processes. In *Proc. of the WI*, pages 1367–1383, 2005.

[2] H. H. Do, S. Melnik, and E. Rahm. Comparison of Schema Matching Evaluations. In *Proc. of GI-Workshop Web and Databases*, 2002.

[3] H. H. Do and E. Rahm. COMA—a system for flexible combination of schema matching approaches. In *Proc. of the VLDB*, 2002.

[4] A. Doan, P. Domingos, and A. Halevy. Reconciling Schemas of disparate Data sources: A Machine Learning Approach. In *Proc. of the ACM SIGMOD01 Conference*, 2001.

[5] A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology matching: A machine learning approach. *Handbook on Ontologies*, pages 385–516, 2003.

[6] F. Giuchiglia and P. Shvaiko. Semantic Matching. *Knowledge Web Review Journal*, pages 265–280, 2004.

[7] J. Hartmann, E. Paslaru-Bontas, R. Palma, and A. Gomez-Perez. DEMO - A Design Environment for Metadata About Ontologies. In *Proc. of the ESWC2006*, 2006.

[8] J. Madhavan, P. A. Bernstein, and E. Rahm. Generic Schema Matching with Cupid. In *Proc. of the VLDB2001*, 2001.

[9] D. McGuinness, R. Fikes, Rice, J., and S. Wilder. The Chimaera ontology environment. In *Proc. of the AAAI2000*, pages 1123–1124, 2000.

[10] S. Melnik, editor. *Generic Model Management. Concepts and Algorithms*. Springer Verlag, 2004.

[11] M. Mochol. Metadata-Based Matching Framework for Ontologies. In *Proc. of CAiSE'06 Doctoral Consortium*, 2006.

[12] M. Mochol and E. Paslaru Bontas. A Metadata-Based Generic Matching Framework for Web Ontologies. Technical Report TR-B-05-03, FU Berlin, June 2005.

[13] E. Paslaru Bontas, M. Mochol, and R. Tolksdorf. Case Studies on Ontology Reuse. In *Proc. of the I-KNOW*, 2005.

[14] E. Paslaru Bontas, S. Tietz, R. Tolksdorf, and T. Schrader. Generation and Management of a Medical Ontology in a Semantic Web Retrieval System. In *Proc. of the CoopIS/DOA/ODBASE (1)*, pages 637–653, 2004.

[15] J. Poole and J. Campbell. A Novel Algorithm for Matching Conceptual and Related Graphs. *Conceptual Structures: Applications, Implementation and Theory*, pages 293–307, 1995.

[16] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *Journal of Very Large Data Bases*, 2001.

[17] G. Stumme and M. Alexander. FCA-MERGE: Bottom-up merging of ontologies. In *Proc. of the IJCAI2001*, pages 225–230, 2001.

[18] K. Tu and Y. Yu. CMC: Combining Multiple Schema-Matching Strategies based on Credibility Prediction. In *Proc. of DASFAA*, 2005.