Metadata-Based Matching Framework for Ontologies

Malgorzata Mochol

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

Abstract. Current algorithms cannot be used optimally in (semi)automatic ontology matching tasks as envisioned by the Semantic Web community, mainly because of the inherent dependency between algorithms and ontology properties. As one possible solution we suggest a metadata-based ontology matching framework that takes into account the difficulties of current matching algorithms. It uses additional data concerning different matchers to set the rules for the methods and their characteristics in relation with the ontologies to exploit the advantages of these approaches.

1 Introduction

Due to its open design, a fully developed Semantic Web will contain numerous, distributed and ubiquitously available ontologies. Users will be able to choose among the different ontologies to allow mediated access to Web information, or to integrate or transform them into application-specific, customized models. Hence, various approaches for ontology merging, mapping and matching will be needed. Among these methods matching algorithm plays the key operations for enabling the global success of Semantic Web because of its implications in almost every phase of an ontology engineering and management process[6]. The ontology matching approaches must be capable of handling heterogeneity of ontological sources available on the Web in terms of representation languages, varying degrees of maturity and granularity levels, natural languages and diverging views of the modelled domains (cf. Sec.4). Though containing valuable ideas and techniques current matching approaches are tailored to certain types of ontologies and lack exhaustive testing in real world scenarios. In this context we will present a *Metadata-based Ontology Matching (MOMA) Framework* that tackle some limitations by maintaining awareness of the link between matching algorithms and the various ontologies. Our approach allows for a more flexible, automated deployment of matching algorithms, depending on their suitability to certain phases of the ontology management process.

2 General approach of the thesis

In the face of weaknesses of different matching approaches(cf. Sec.3) we have defined our main questions of the thesis: how the suitable matching approach for a given problem can be chosen and which methods should be taken into account. We have identified the problem of criteria on which the approaches could be differentiated w.r.t the given ontologies and desired results, as well as examined the *possible factors* having an impact on the selection of the algorithms. Following this, we have *factor: approach*, i.e. matching algorithms with their attributes and *factor: input*, i.e. ontological sources to be matched and their characteristics. In the next step we have to decide, considering the requirements collected during the development of different Semantic Web applications[1, 7], which attributes of approaches and inputs are relevant and which could be neglected. The chosen attributes do not stand alone but are rather in a close relationship to each other. Therefore a set of rules have to be developed to link ontology and matching properties and help to determine which matching algorithms are to be used for which type of ontologies. After defining the factors and competency rules, engines for rules and matching execution must be developed. The first phase of the evaluation of the presented matching framework will occur with the adaption of this approach to the different application scenarios.

3 Matching Framework based on Rules and Metadata

Matching conceptual structures, be they database schemes, XML schemes, conceptual graphs or, more recently, Semantic Web ontologies, is a discipline with a long history and plays a significant role in various areas of Computer Science, such as data integration, data warehouses, agent communication and Web Service composition. The importance of the matching issue is reflected by the high number of matching algorithms which have been proposed and applied to particular application settings in the last decades[3, 5, 10, 17, 22]. Though containing valuable ideas they feature some limitations, e.g. they (i)cannot be applied across various domains with the same effect[8];(ii)require certain representation or natural languages[3];(iii)perform well on relatively small inputs with at most hundreds of concepts but have not been tested or are not scaled to process complex schemes;(iv)do not perform well on inputs with heterogeneous (graph) structures[8] or are restricted to tree-based conceptual models[8, 12];(v)deliver results that are based on a one-to-one mapping between taxonomies[4];(vi)need some manual pre-processing[2]. Hence, there is not and will never be an overarching matching algorithm for ontologies capable of serving all (heterogeneous) ontological sources. As one possible solution for this problem we propose a *Metadata-based Ontology Matching (MOMA) Framework*[1] which chooses the “suitable” approaches w.r.t given ontologies by using the additional information regarding the ontologies (ontology metadata) as well as information regarding different matchings (matching metadata) and putting these two (by means of rules) in relation to each other. The *MOMA Framework* consists of the following components (cf. Fig.1):(i)**matching repository** which contains reusable matching components and **metadata** describing their properties, e.g. information concerning the ontological formats served by particular matches and information

[1] `http://moma.ag-nbi.de`

regarding the natural language that the algorithm can handle; (ii)**ontology repository** which manages the matching inputs defined by the **ontology metadata**;(iii)**rule repository** linking ontology and matching properties and helping to determine which matching algorithms is to be used for which type of ontologies;(iv)**selection engine** responsible for the process to determine the algorithms applicable to a specific set of inputs and the engine responsible for the **execution** of the matching workflow.

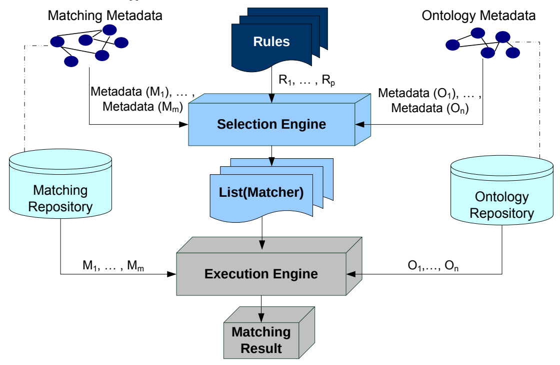

Fig. 1. Architecture of the Matching Framework

In the following we describe the *MOMA matching* and *ontology metadata*, which connected with each other by means of rules, are directly involved in the process of choosing the suitable matching algorithm.

3.1 MOMA Matching Metadata

Matching metadata describes single ontology matchers of the matching repository. The first version of the model used to classify the matching algorihms relies heavily on[18][2]. This classification makes a distinction between individual matchers which compute a mapping based on a single matching criterion and combining matchers which use multiple individual matchers:

• **Individual matchers** can work on instance data (*instance/contents-based matchers*) or consider only structure information, be they relationship types, data types or schema structures (*schema-only based matchers*). Both algorithms can be applied to individual schema elements, such as attributes or concept labels (*element-level matchers*). Schema-only based approaches can also deal with combinations of schema elements, thus allowing for the computation of mappings by analyzing sub-graphs (*structure-level matcher*). A single element-level matcher uses *linguistic* as well as *constraint-based* techniques, while a schema-only based matcher uses the latter only.

• **Combining matchers** are divided into: *composite matchers*[3, 5, 23] which combine the different results of independently executed matchers whereupon the order of the execution of the individual matchers can be assign manually or semi-automatically (manual or automatic composition), and *hybrid matchers*[10]

[2] Next version will take into account additional characteristics e.g.[14] and[19]

that follow a black box paradigm and do not allow manual intervention. Beside the aforementioned classification, the metadata model includes: (i)*input type*: instances, schemas or additional input (e.g. numerical values);(ii)*matching level*: atomic level, e.g. attributes in an XML schema and higher (non-atomic) level e.g. XML elements;(iii)*completeness*: a full match incorporating all the elements of the two schemes, as opposed to a partially match and (iv)*cardinality* - specifies whether a matcher compares one or more elements of one schema with one or more elements of another schema. Furthermore, properties of the ontologies to be matched, such as formality level, domain type, representation/natural language, supported used primitives etc. are defined in the ontology metadata (cf. Sec.3.2) and are referenced in the matching metadata to refine the description of the matching inputs. The matcher classification supplemented by the aforementioned features is conceptualized in a high formal representation language. The first version have been implemented in OWL[3] where the matcher types are defined as OWL classes within a hierarchical structure with "sub-class" relationships between them. By means of OWL constraints we have specified the characteristics of each type of matching algorithm (e.g. the input of an instance-based matcher can not be a scheme without instance data). This ontology in conjunction with the ontology metadata and the rules form the knowledge basis of the matching engine, which ultimately contribute to the automatic exclusion of unsuitable matchers.

3.2 MOMA Ontology Metadata

The information model used to describe matching inputs, i.e. ontologies, contains three features: syntactic, semantic and pragmatic. These features are parts of the so-called "ontology context"[15] which is an information model used to describe ontologies in various phases of their lifecycle. Matching algorithms cannot be applied with the same expectations of success regardless of all the dimensions of the ontology information model mentioned. In particular, we have identified the following ontology features as relevant for matching tasks: **quantitative syntactic features** (e.g. the number of specific ontological primitives) that influence the matching execution performance and **semantic features** (e.g. domain, representation/natural language and type of domain) that restrict the number of applicable matching algorithms, which can be domain-specific or accept specific types of schemes.
The model itself is defined in a formal representation language (current implemented in OWL[4]) and the dependencies between "ontology context" and different matching approaches are formalized in terms of rules (cf. Sec.3.3).

3.3 MOMA Rules

For a given pair of ontologies to be matched the selection engine must decide (using the information from *MOMA metadata*) which matching algorithms are to

[3] `http://wissensnetze.ag-nbi.de/matching/matchingmetadata.owl`
[4] `http://wissensnetze.ag-nbi.de/matching/meta.owl`

be applied to satisfy the given requirements. The engine is aware of background information that describes the available matchings and the properties of the input ontologies. However, in order to automatically infer which algorithms suit the concrete inputs, it needs explicit knowledge concerning the dependencies between these algorithms and the (type of) sources on which they operate. We have formalized this knowledge in terms of dependency rule-statements that determine which elements (here, which matchers) are to be used or excluded. Some general rules for ontologies and matching algorithms are defined[13] w.r.t.:
(1) *natural language*: if both ontologies are defined in the same natural language → eliminate all matchers which serve other languages
(2) *representation language*: if ontologies are defined in the given representation language → eliminate all matchers which are unable to serve this language
(3) *matching direction*: if the direction of matching algorithm is not defined → match the smaller ontology against the bigger one
(4) *instances*: if no instance data is available → apply only scheme matchers
(5) *axioms*: if it is a formal ontology and if ontologies contain axioms → use constraint-based matchers
(6) *ontology*: if it is a single ontology → apply only instance matchers.
For the purpose of the first implementation, the matching rules are defined in SWRL[9], a rule language for the Semantic Web, which allows us to formalize them in terms of the concepts defined in the two metadata models. However, the deployment of rules in decision-making processes requires a reasoning engine able to operate on (OWL) ontologies and (SWRL) rules — an issue which is still the subject of active research in the Semantic Web community.

4 Matching use cases

One of the projects we are working on explores the potential of Semantic Web from a business and a technical perspective by examining the effects of the deployment of Semantic Web technologies on specific application scenarios and market sectors[5]. For this purpose, we have developed a scenario for the recruitment domain in which we analyze the online job seeking and job procurement processes and the implications of using Semantic Web technologies in this area[1]. The first step towards the realization of the e-Recruitment scenario is the creation of a human resources ontology (HR ontology) to be used in a Semantic Web job portal that allows a uniform representation of job offers and applicant profiles, and semantic matching in job seeking and procurement tasks. In an effort to support common commercial practices and maximize the integration of job seeker profiles with job postings from various organizations, our target HR ontology reuses established domain-specific standards and classifications. At this point we realize there's a need for two different matching approaches to tackle the HR scenario: one which (at the initial stage of ontology development) allows us to compare relevant existing source ontologies with the aim to develop a target HR ontology (*reuse task*) while the second enables applicant profiles to

[5] Knowledge Nets project; `http://wissensnetze.ag-nbi.de`

be run up against job descriptions (*application task*). Ontology reuse may be defined as a process in which available knowledge is used as input to generate new ontologies. Depending on the extent and manner in which they are used, sources contributing to the target ontology one can distinguish between:(i)*reusing the vocabulary* - the terms of the source ontologies are incorporated into the target ontology (e.g. source ontologies are represented at a very low level of formality in which a clear definition of the terms is lacking or cannot be re-used efficiently, or when the new definitions of the concepts are incompatible with the original ones);(ii)*reusing the vocabulary and the semantics* - both terms and their semantic definitions are integrated into the source ontology (utilized when the meaning of the acquired concepts is extended in the source ontology, though nevertheless implying the compatibility of the semantic models involved);(iii)*reusing parts of an ontology* - parts (sub-ontologies) are extracted from the source ontologies for reuse. Provided a common representational formalism for our scenario, source ontologies (e.g. skills description, classification of occupations and business sectors) need to be compared and merged, by deploying all three types of reuse. For this purpose, we needed (scheme) matching algorithms capable of handling the heterogeneity of the incoming sources (defined by the ontology metadata) w.r.t. their natural or representation languages (cf. rules 1 and 2 from Sec.3.3), structure, domain (as they do not usually cover exactly the same domains[11]), and varying degrees of formality (cf. rule 5), granularity or maturity[16].
The second task of the matching approach concerns the process of finding an appropriate job for an applicant or a suitable candidate for a job opening. Since we have defined our HR ontology using *MOMA ontology metadata* which delivers information like: HR ontology is defined in a formal representation language (OWL), includes instances, does not contain any axioms and the concepts are defined in one natural language (here, German), we can apply our generic rules (e.g. rules 1, 2, 5) which eliminates unsuitable matching approaches. Since the job description as well as applicant profiles are based on the same HR ontology after applying rule 6, we can (using information from *MOMA matching metadata*) pinpoint from the list of possible matching algorithms the instance matchers which deal with the representation language of our ontology.
Our HR scenario has highlighted the importance of matching issues within Semantic Web field while at the same time showing the complexity and heterogeneity (two different matching approaches utilized for single use case) of the matching issue.

5 Related Work

Due to the fact that our approach does not intend to create a new matching algorithm we do not compare our work with individual algorithms like FCA-MERGE[22] or S-Match[8]. Since our *MOMA Framework* aims to apply existing matching solutions some approaches from the (semantic) web service composition[20, 21] as well as the hybrid and composite matching solutions are relevant. The best example of the latter approach is the COMA framework[3] which supports different applications and scheme types and provides an extensible library of matching algorithms, a component for combining the results obtained and extensive functionality for the evaluation of matching effectiveness. One of COMA's weak points comes, however, from the fact that the suggested combined methods may prove to be inadequate for complex situations. Since each base matcher performs differently in different settings, simple, pre-defined composition methods are incapable of capturing such performance variations[23]. In such cases, users are required to manually customize the matching workflow, however this means the framework cannot be directly employed in environments that functions without human intervention. A Web-enabled matching framework requires a highly flexible selection and composition of available matching services in order to take full advantage of the broad spectrum of ontologies across the network. We believe that joining different matching techniques in composite or hybrid algorithms offers only partial solutions to this issue of heterogeneity. Within these combined frameworks, the quality of the matching results is implicitly dependent on the features of the input ontologies, and as a result this framework cannot be put to optimal use in an open, distributed environment such as the Web.

6 Summary and Future work

This paper presents the preliminary ideas on the proposed PhD dissertation which intends to develop the ontology matching framework which is characterized by the usage of the metadata for ontologies and matching and by the application of rules to determine which matching algorithms are appropriate for individual cases. The preliminary version of the prototypical implementation of the *MOMA Framework* contains the ontologies with metadata, metadata for matching algorithms and a few general prescribed rules. The first phase of the further development will be dedicated to expansion of the matching metadata and definition of the comprehensive set of rules using information related to the syntactic and semantic dimension of ontology inputs. The subsequent step will be connected with another important topic which we have not yet addressed in the paper - the matching execution - which requires an automatic composition of the candidate matching services to achieve the desired results.

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 7th Internationale Tagung Wirtschaftsinformatik 2005*, 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 28th VLDB Conference*, 2002.
4. A. Doan, P. Domingos, and A. Halevy. Reconciling Schemas of disparate Data sources: A Machine Learning Approach. In *Proc. of the the SIGMOD01*, 2001.

5. A. Doan, J. Madhavan, P. Domingos, and A. Halevy. Ontology Matching: A Machine Learning Approach. *Handbook on Ontologies*, pages 385–516, 2004.
6. Alfio Ferrara. Methods and Techniques for Ontology Matching and Evolution in Open Distributed Systems. In *Doctoral Consortium at the Conference on Advanced Information Systems Engineering (CAiSE'04)*, 2004.
7. J. Garbers, M. Niemann, and M. Mochol. A Personalized Hotel Selection Engine (Poster). In *3rd European Semantic Web Conference*, 2006 (to appear).
8. F. Giuchiglia and P. Shvaiko. Semantic Matching. *Knowledge Web Review Journal*, pages 265–280, 2004.
9. I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. `http://www.w3.org/Submission/SWRL`, 2004.
10. J. Madhavan, P. A. Bernstein, and E. Rham. Generic Schema Matching with Cupid. In *PROC. of the 27th VLDB Conference*, 2001.
11. J. Madhavan, P.A. Bernstein, P. Domingos, and A.Y. Halevy. Representing and Reasoning about Mappings between Domain Models. In *Proc. of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence (AAAI/IAAI)*, pages 80–86, 2002.
12. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *Proc. of the 18th International Conference on Data Engineering (ICDE02)*, 2002.
13. M. Mochol and E. Paslaru Bontas. A Metadata-Based Generic Matching Framework for Web Ontologies. Technical Report TR-B-05-03, FU Berlin, `http://www.inf.fu-berlin.de/inst/pubs/tr-b-05-08.abstract.html`, June 2005.
14. P. Mork and P. A. Bernstein. Adapting a Generic Match Algorithm to Align Ontologies of Human Anatomy. In *Proc. of the 20th International Conference on Data Engineering (ICDE'04)*, page 787, 2004.
15. E. Paslaru Bontas. Context-enhanced Ontology Reuse. In *Doctoral Consortium at the 5th International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT05)*, 2005.
16. E. Paslaru Bontas, M. Mochol, and R. Tolksdorf. Case Studies on Ontology Reuse. In *Proc. of the 5th International Conference on Knowledge Management*, 2005.
17. J. Poole and J.A. Campbell. A Novel Algorithm for Matching Conceptual and Related Graphs. *Conceptual Structures: Applications, Implementation and Theory*, pages 293–307, 1995.
18. E. Rham and P. A. Bernstein. A survey of approaches to automatic schema matching. *Journal of Very Large Data Bases*, 2001.
19. P. Shvaiko. Iterative schema-besed semantic matching. Technical Report DIT-04-020, University of Trento, `http://eprints.biblio.unitn.it/archive/00000550/01/020.pdf`, June 2004.
20. E. Sirin, J. Hendler, and B. Parsia. Semi-automatic Composition of Web Services using Semantic Descriptions. In *Web Services: Modeling, Architecture and Infrastructure workshop in ICEIS 2003*, 2003.
21. B. Srivastava and J. Koehler. Web Service Composition - Current Solutions and Open Problems. In *Workshop on Planning for Web Services, ICAPS 2003)*, 2003.
22. G. Stumme and M. Alexander. FCA-MERGE: Bottom-up merging of ontologies. In *Proc. of the 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pages 225–230, 2001.
23. K. Tu and Y. Yu. CMC: Combining Multiple Schema-Matching Strategies based on Credibility Prediction. In *Proc. of 10th International Conference on Database Systems for Advanced Applications (DASFAA 2005)*, 2005.