iweightings: Enhancing Structure-based Ontology Alignment by Enriching Models with Importance Weighting

Alexandra Mazak, Monika Lanzenberger
Institute of Software Technology and Interactive Systems
Vienna University of Technology
Vienna, Austria
{*mazak,lanzenberger*}*@ifs.tuwien.ac.at*

Bernhard Schandl
Department of Distributed and Multimedia Systems
University of Vienna
Vienna, Austria
bernhard.schandl@univie.ac.at

***Abstract*—Structural ontology matching methods analyze mainly two factors: entity labels and relationships among entities. We propose to additionally consider an importance and relevance factor, which is determined by two indicators automatically calculated by a (simple) weighting method. This weighting factor represents the importance of a concept based on its information significance in the modeling context and, additionally, its relevance for structure-based alignment depending on the number of relationships this concept participates in quantified by the rweighting indicator. The method starts via a manually weighting annotation of relationships among concepts conducted by ontology engineers during the ontology development process. Our approach is an assistance mechanism to improve the ontology alignment process and to enhance the cognitive support for users. Thus, ontology alignment becomes already important ex ante when the ontology development process starts, unlike other alignment techniques, which consider only ex post knowledge.**

***Keywords*-Ontology alignment, Meta-Object Facility, ECore meta-model, importance and relevance weighting, candidate ranking.**

I. Introduction

Frequently, ontologies based on the same domain of interest are similar but also have many differences, which are also known as heterogeneity. The reason behind heterogeneity is rooted in diversity in ontology modeling based on different views people have on a domain. Heterogeneity cannot be avoided in distributed and open systems as, for instance, the Semantic Web. Moreover, ontology engineers may follow different modeling foci, e.g., due to different business goals.

Modeling heterogeneity can occur on various levels: *semantic heterogeneity* bases on different meaning whereas *semiotic heterogeneity* bases on the individual interpretation of ontology entities and the various relationships among them regarding to a certain context, i.e., the specific usage of the ontology. *Terminological heterogeneity* occurs due to variations in names referring to the same entities in different ontologies. In [1] several kinds of heterogeneity are introduced in detail.

Different ontology creators have different interests relating to the development of an ontology. Thus, there exists no global view in modeling a certain domain of interest. Quite contrary, there exist many subjective views, causing a variety of perspectives regarding to the certain context an ontology will be used in. Therefore, heterogeneity exists due to the intended usage of ontology entities, i.e., the importance and relevance of the concepts and the relationships among them. *Ontology alignment* is used to bridge these heterogeneities in order to make ontologies and corresponding instance data interoperable.

For instance, there might be two conference track ontologies (e.g., [20]) to align. In one ontology the modeling focus has been put on the *documents* published during conference, whereas in the other ontology the focus lies on the *organization* and the *events*. Thus, a problem in ontology alignment can be that a matching system identifies equal concepts in both ontologies (e.g., *author*), but when manually comparing both ontologies the concept *author* is not equally important and therefore, may convey different information content. The use of entities has significant impact on their importance and interpretation; therefore, matching entities which are not meant to be used in the same context is often error-prone.

A second problem in the alignment process occurs due to terminological heterogeneity: for instance, *contribution* and *article* might be used in the two ontologies to describe the same thing, i.e., a written contribution to a conference. The two terms are used synonymously but it is not straightforward to detect them as equal, neither by string-based techniques nor manually by users if they are not fully aware of the modeling context.

The result of heterogeneity causes difficulties when handling, matching, and reusing ontologies. For instance, in [16] an online user survey was conducted with the goal to understand what processes users are following to discover, track, and compute mappings. One user feedback to the mapping process stated that it would be a great benefit: "*to get into the brains of the original developers*"; for a better understanding of the semantics of the underlying ontologies, i.e., the meaning encoded in the schemas. Domain-related background knowledge is an important component in the ontology matching process [15]; thus cognitive support for ontology engineers is required.

Therefore the question arises, whether it is possible to make the modeling focus of the knowledge engineers explicit; for instance, for users at a later date, e.g., when starting an alignment process?

Further questions which arise in this context include:

- Which concepts convey significant information about the hidden modeling focus?
- How can knowledge engineers weight the importance of a concept considering its usage within the ontology during the conceptualization (development) process?
- How can users perform a ranking of concepts either based on their importance or on their number of relationships they participate in?
- How can users detect the core concepts of ontologies when starting an alignment process?

We assume that the meaning (importance and relevance) of a certain class or concept essentially depends on a certain usage and purpose, for which the ontology has been modeled. Based on this assumption, we present in this paper a contribution for answering these questions. We first present design considerations in Section 2. Next we describe our approach to importance and relevance weighting for ontology alignment in Section 3. In Section 4 we present related work. Finally, in Section 5, we discuss our concluding remarks and directions for future work.

II. Design Considerations

The first part of our approach suggested in this paper is based on the following idea: we explicitly encode the importance and relevance of concepts (classes) using two weighting indicators. The first one, denoted $iw_c(x)$ is a numerical value derived by weighting the local context of a concept x; the second one, $rw_c(x)$, additionally considers the number of outgoing relationships of x (cf. Sections III-B and III-C).

The local context of a concept within an ontology is described by the relationships in which it participates in; accordingly, the property domain and range axioms which constrain an object property (relation between the instances of two classes) are taken into account. These axioms constitute links among concepts and properties. Thus, our main focus is on the semantic connections among classes of an ontology.

The method starts via a manually weighting annotation of relationships among concepts conducted by ontology engineers during the ontology development process (see Figure 1).

Our approach is a contribution to improve the ontology alignment process. Thus, ontology alignment becomes already important ex ante when the ontology development process starts, unlike other alignment techniques which consider only ex post knowledge.

According to the classification in [1] our contribution can be subordinated to *structure-level techniques*, which consider the relations of entities with other entities. In addition, our approach can also be classified as an *element-level technique*, because the domain and range axioms specify the terminological correlation among classes and properties [5]; thus, we also take *terminological techniques* into account. Additionally, we consider *model-based techniques*, which handle the entities of ontologies based on their semantic interpretation.

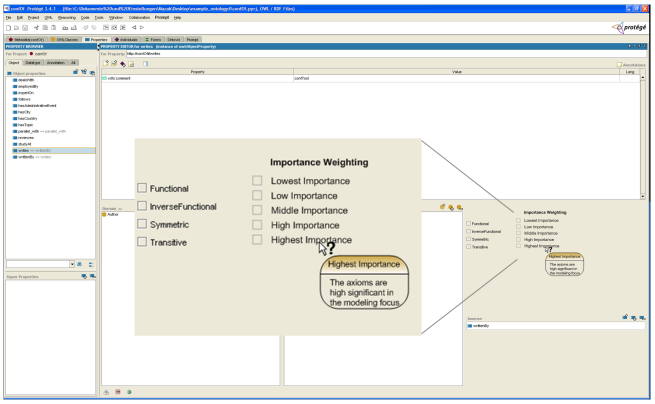

Figure 1: Example for a weighting annotation in Protégé

In analogy to the classification in [2] our contribution is assigned to the *ontology layer* at the level of semantic networks, where ontologies are viewed as graphs consisting of concepts and their relationships. Additionally, we are taking into account the *context layer* where the practical usage of the entities will be concerned in the context of the application.

In [4] our approach applies to the conceptual theory. This method works with concepts and compares their meanings in order to compute alignments. Moreover, our contribution relies on a simple statistical method, since a weighting average is calculated for each concept in the ontology, based on its outgoing edges, i.e., relations to other concepts.

In summary, our approach is a *hybrid-method* and according to [1] it can be seen as a "*cross-fertilization*" to gain more evidence for ontology alignment in the future.

III. Approach

A. Conceptual Design

An ontology is expressed in a specific ontology language. There a variety of languages allow users to write explicit, formal conceptualizations of domain models. The Web Ontology Language (OWL) is an ontology language recommended by the W3C [18]. OWL consists of three differently expressive representation formalisms: OWL Lite, OWL DL and OWL Full. Each of these sub-languages is an extension of its predecessor.

An ontology language contains different types of entities, the most important ones are called concepts or classes. OWL classes are sets of individuals in the domain of interest called

class extensions. All individuals of a domain are subsumed by the *owl:Thing* class. Relations are interpreted as a subset of the Cartesian product of the domain classes and are denoted in OWL as *owl:ObjectProperties*.

In our contribution we focus on the object properties and their domains and ranges based on the context of the application. Thus, we focus on the relational structure of the ontology and therefore on the schema axioms.

A domain ontology describes a particular domain of interest. In OWL DL we distinguish between the ABox, which contains *assertional knowledge*, and the TBox, which contains the *terminological knowledge* of the domain. The TBox contains the schema axioms (i.e., concepts and their relations), while the ABox contains the data axioms (i.e., the individuals of the concepts and the statements which they are belonging to). Thus, the TBox is assigned to the model-structure layer and the ABox to the information layer according to the four-layer metadata architecture [8] introduced by the Object Management Group (OMG).

For the realization of our approach we have applied the ECore meta-model [3]. ECore is the core meta-model in the Eclipse Modeling Framework (EMF), which supports the main concepts of the Model-Driven Architecture (MDA). The Meta-Object Facility (MOF) is a key standard in the MDA family. MOF is the basis of the OMG´s MDA. EMF is an open source model-driven software development platform and an efficient Java implementation of a core subset of the MOF API. Using the concepts of MOF we are able to define the abstract syntax of meta-languages like OWL or RDF Schema. MOF has two parts essential MOF (EMOF) and complete MOF (CMOF).

For our approach we use EMOF [8], which is implemented in the EMF´s ECore meta-model. Therefore, EMOF is compatible with ECore and it can be used to extend a meta-model like OWL DL.

For the realization of the importance weighting approach we use the ECore class *EReference*, which is a kind of pointer to represent the ends of a relation between classes. With EReference we are able to annotate the *owl:ObjectProperty* with an importance weighting value depending on its specific domains and ranges. The *EEnumerator* data type helps us to represent the weighting values for the EReference class *Weighting* by using literals.

In the four-layer architecture a concrete ontology is defined in the language *S1* (i.e., for instance, OWL DL). The language *S1* is described in the model at level M2, i.e., the OWL meta-model. The meta-model at M2 is a language-oriented meta-model based on the domain of interest modeled at M1. Real facts of the domain of interest are modeled at level M0, the information layer.

Figure 2 illustrates our extended OWL DL meta-model by using the constructs of the ECore meta-model at level M2.

The extensions of the meta-model, *Weighting* and *Importance*, are highlighted in order to distinguish them from the elements of the common OWL meta-model, which are defined in the *Ontology Definition Metamodel (ODM)* [9].

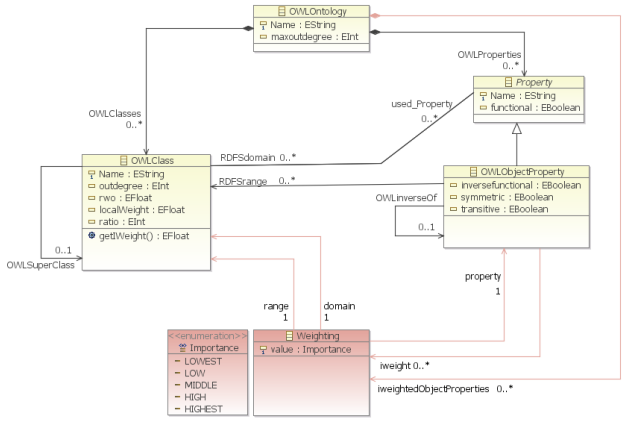

Figure 2: Extension of the OWL DL meta-model with the class Weighting and the class Importance.

B. Integration of iweightings in the Ontology Authoring Process

In our approach we describe ontology concepts by their importance and relevance. These features represent certain semantics based on the modeling context derived from the individual usage of the concepts within the ontology.

The mockup in Figure 1 shows a simple and user-friendly mechanism to assert weightings for object properties related to their particular domain and range axioms when an ontology engineer develops a domain ontology. The user interface of this mockup is based on the open source ontology editor Protégé [19] in the version 3.4.1.

For instance, in a first step the ontology engineer creates the object property *write* and defines its domain *author* and range *contribution*. Additionally, he/she assigns the iweighting value *Highest Importance* to the used object property and its domain/range combination. Thus, the iweighting value is determined by the certain domain and range axioms of the object property. We distinguish five weighting values, as presented in Table I. We think that users prefer to assign importance labels instead of importance values. Thus, the system automatically converts these importance labels to numerical values for further computation.

In a second step the system calculates a local weighting average for each class by considering the used object properties. We assume that the ontology developers accept the recommendation proposed in [6]: "*Each object property may have a corresponding inverse property.*" Thus, each range concept will be a domain concept of the inverse property and therefore, get an automatically calculated iweighted average value, the so-called *iweighting indicator*.

Table I: iweighting Importance Degrees

Importance Weighting	Description
Lowest Importance	The axioms are least significant in their meaning in the modeling focus.
Low Importance	The axioms are only lowly important in their meaning in the modeling focus.
Middle Importance	The axioms have a fair importance in their meaning in the modeling focus.
High Importance	The axioms are highly important in their meaning in the modeling focus.
Highest Importance	The axioms are highest significant in their meaning in the modeling focus.

More formally,

$$iw_c(x) = \frac{1}{|OP(x)|} \sum_{i \in OP(x)}^{|OP(x)|} iw_{OP_i}^{(x,y)} \quad (1)$$

where $iw_c(x)$ is the iweighting indicator of a concept x based on the average importance weights, which have been manually asserted by the ontology engineers during the ontology development process ($iw_{OP_i}^{(x,y)}$); where OP are the used OWL object properties of the *domain concept (x)* constrained by their particular *range (y)* axioms.

For instance, the mockup presented in Figure 3 shows a pan of the conference track ontology *confOf* [20] where the concept *Author* has an iweighting indicator of 0.95 and is therefore more important than its parent class *Person* with an iw_c(Person) of 0.12. Why is the difference between the super- and its subclass so high?

The answer lies in the meaning of the ontology concepts: *Person* acts only as a topic in the taxonomy; the more significant information is considered in the term *Author* and its usage in the modeling context. For instance, the main focus in the *confOf* ontology is on the concepts *Author* and *Contribution*.

C. Using iweightings in the Alignment Process

As introduced in Section I users need efficient cognitive support before starting the alignment process. We provide two indicators for each ontology concept: an *iweighting indicator* as described in the previous section, and an *rweighting indicator*, which additionally considers the number of the concept's object properties to determine its relevance. These indicators are two modes for ranking ontology concepts by their importance and/or relevance.

By selecting the iweighting indicator it is possible to rank all concepts by their importance, i.e., by their information significance in the modeling context; thus, all core concepts can be detected.

Why do users need a second indicator to rank concepts when starting an alignment process?

For instance, from concepts with a *Highest Importance* it cannot be derived that they participate in many relationships

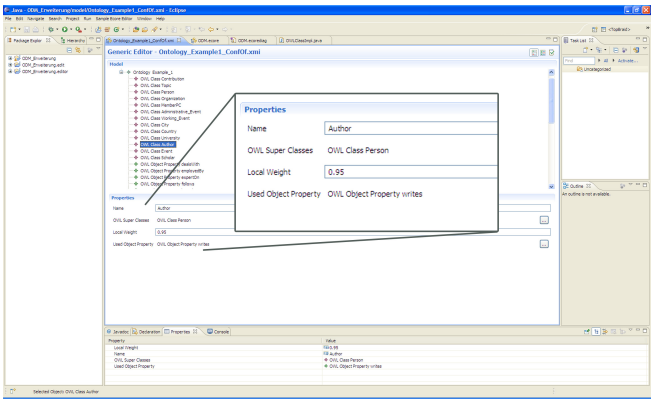

Figure 3: Example for the calculated weighting average for the OWL Class *Author* by the implemented weighting-algorithm using Eclipse.

to other concepts, which is an important fact for applying graph-based alignment tools: firstly, to detect efficient starting points (nodes), and secondly to traverse as many paths as possible in the sub-graphs. By selecting the rweighting indicator users can consider this fact.

The *rweighting indicator* is defined as

$$rw_c(x) = \alpha \cdot iw_c(x) + (1 - \alpha) \cdot \frac{|OP(x)|}{max|OP(x)|} \quad (2)$$

First the algorithm calculates the iw_c of a concept x according to (1); additionally, the algorithm calculates the outdegree for each concept relative to the maximum number of a concept's object properties (maximum outdegree of the ontology).

We enable the user to introduce their preference for candidate relevance ranking by using α defined in (2); α depends on whether the user primarily focuses on the $iw_c(x)$ or the number of relationships the concept x participates in. By default the algorithm assumes $\alpha = 0.5$, i.e., both terms are equally significant for ranking.

Describing the benefits in *feature engineering* and *search step selection* in the alignment process according to [2] we choose *Anchor-PROMPT* [10] for demonstration. Anchor-PROMPT is a tool for graph-based mapping and alignment operations, and its based on an algorithm using the graph structure of ontologies. Thus, Anchor-PROMPT considers ontologies as directed labeled graphs, in which it searches for correlations among concepts between two ontologies by parallel-traversing paths of a certain parameter length predefined by the user between originating and terminating points in each subgraph of the two ontologies. The notion of these initial points is *anchors* and the length of a path is the number of edges in the path. A path follows the links (directed labeled edges) between classes (nodes) defined by hierarchical relations or by slots and their domains and ranges. Thus, a non-local context is taken into account by the Anchor-PROMPT algorithm.

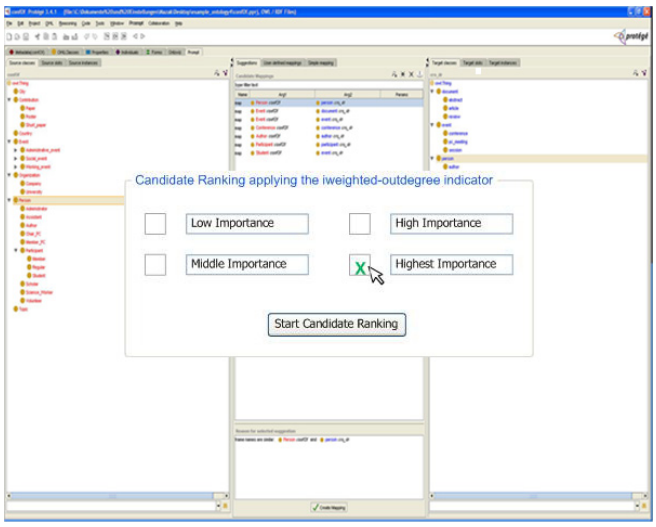

Figure 4: Example for starting a candidate ranking in Anchor-PROMPT.

The anchor pairs between two ontologies are defined either manually by the user, or automatically by lexical matching methods. For the user it is difficult to find useful sets of related terms between two ontologies especially if the models are very large. A large number of matches can be found by lexical matching methods, but often they are not significant; for instance, *house* and *mouse* have a string similarity score of 0.75 computed by the edit distance, but is this similarity value helpful?

Our weighting approach supports the user interaction with the system in the task of selecting the core concepts of the source ontologies; additionally, considering their number of relationships in which they participate in as domain concepts. Figure 4 shows an example for a concept ranking query using Anchor-PROMPT, which can be applied by configuring the PROMPT Tab Widget in Protégé.

Thus, an easy and quick finding of appropriate initial pairs can be accomplished with simple point-and-click interaction by the user. Additionally, the user is disburdened from the need to analyze the structure and concepts of the source ontologies, just to determine the originating and terminating points.

To demonstrate our approach, we use Anchor-PROMPT to align the ontologies *confOf* and *crs_dr* [20], which model a conference track as the domain of interest. The system suggests

- *Person* (confOf), *person* (crs_dr), and
- *Event* (confOf), *event* (crs_dr)

as initial points detected by lexical matching. The Anchor-PROMPT algorithm finds no correspondences between these anchor pairs because no relationships exist between them. Using the possibility of ranking the concepts by applying the rweighting indicator the system detects that *article* (crs_dr) and *Contribution* (confOf) have both a very high information significance and are therefore of highest importance; additionally, they are involved in more than one relationship to other concepts with an rw_c in the range of $(0.75; 0.95]$. Actually, there exist two relationships between the anchor points *Person* and *Contribution* from the *confOf* ontology and more than two relationships between *person* and *article* from the *crs_dr* ontology; therefore, the Anchor-PROMPT algorithm could traverse more potential paths, and more correspondences between the anchor points could be detected.

IV. Related Work

In [1] detailed information is given about techniques that, similar to other works, are using weights in their approaches; e.g., *statistical methods*, *semantic-based techniques* and other weighting methods. For instance, statistical methods consider the instance data of ontologies at the information layer M0. These methods need the instance data as representative samples to take measurements on which comparisons between two source ontologies can be established.

Semantic-based techniques use, e.g., intermediate formal ontologies to define a common context or background knowledge to bridge the gap caused by the lack of a common ground on which comparison can be based. The common ground can often be found in external resources and models; e.g., DOLCE, or WordNet. These methods help in handling the disambiguation of multiple possible meanings of terms.

Some methods measure semantic-similarity in an *IS-A taxonomy* based on the information content [12]; other methods measure similarity depending on the type of an entity and its features which make its definition [11], or they count the number of outgoing and incoming edges for weighting the edges to compute a *propagation coefficient* [13]. Automatically based ranking methods [17] identify the importance of concepts by counting the number of relations starting from one concept to another in a first step; also taking the importance of the other concept into account.

The techniques listed above share the lack of considering the importance and relevance of concepts in the modeling context. Such a method requires a non-trivial knowledge about the domain of interest [15]. We address this problem in our approach by starting with the weighting already in the ontology development process. Nobody can annotate a weighting factor better than the ontology engineers themselves.

V. Conclusion and Future Work

In the first part of our approach presented in this paper we generate a novel factor supporting a user's decision-making based on the importance and relevance of ontology concepts, e.g., by reducing complexity of large ontologies when starting an alignment process; improving the first two steps in

this process *feature engineering* and *search step selection*. Our weighting algorithm encodes the importance of each concept based on its weighted local context and its relevance for structure-based alignment methods by considering the concept's outdegree.

To further enhance our approach we aim to additionally include OWL datatype propertiesf in an analogous way, since these are not covered by typical graph-based matching methods. Additionally, we plan to support other steps in the ontology alignment process according to [2]: for instance, *similarity computation* can be enhanced by an algorithm that calculates an indicator based on the differences among the iw_c and/or rw_c of concepts between two ontologies to enhance one of the established alignment tools, or to provide users with a quick overview about possible correspondences aiding their cognitive support. Finally, we plan to conduct a detailed user evaluation of our approach.

ACKNOWLEDGMENT

We want to express our thanks to Manuel Wimmer from the Business Informatics Group at the Vienna University of Technology, who contributed his time and knowledge in EMF helping us to realize the meta-model extension.

REFERENCES

[1] J. Euzenat and P. Shvaiko, *Ontology Matching*, Springer-Verlag Berlin Heidelberg, 2007, pp. 37, 40-42, 65, 92-104.

[2] M. Ehrig, *Ontology Alignment, Bridging the Semantic Gap*, Springer Science+Business Media, LLC, New York 2007, pp. 26-28, 61-64, 76-77.

[3] F. Budinsky, D. Steinberg, E. Merks, R. Ellersick and T. J. Grose, *Eclipse Modeling Framwork*, Addision-Wesley, Boston (MA US), 2004, pp. 14-19, 95-103.

[4] S. Zanobini, *Semantic Coordination: The Model and an Application to Schema Matching*, PhD Thesis, International Doctorate School in Information and Communicatin Technology, University of Trento, Trento, 2007, pp. 71.

[5] [P. Hitzler, M. Krtzsch, S. Rudolphg and Y. Sure, *Semantic Web*, Springer-Verlag Berlin Heidelberg, 2008, pp. 76, 167.

[6] M. Horridge, S. Jupp, G. Moulton, A. Rector, R. Stevens and C. Wroe, *A Practical Guide To Building OWL Ontologies Using Protégé 4 and CO-ODE Tools*, Edition 1.1, The University Of Manchester, Manchester, 2007, http://www.co-ode.org/resources/tutorials/ProtegeOWLTutorial.pdf (checked online April-20-2009).

[7] R. C. Gronback, *Eclipse Modeling Project: A Domain-Specific Language Toolkit*, Addision-Wesley, Boston (MA US), 2009, pp. 32-33.

[8] Object Management Group (OMG), *Meta Object Facility (MOF) Core Specification*, Version 2.0, formal/06-01-01, Needham (MA US), 2006, http://www.omg.org/docs_formal/06-01-01.pdf (checked online July-31-2009).

[9] IBM Sandpiper Software Inc., *Ontology Definition Metamodel*, Third Revised Submission to OMG/RFP a/2003-03-40, ad/2005-08-01, Needham (MA US), http://www.omg.org/docs/ad/05-08-01.pdf (checked online August-11-2009).

[10] N. F. Noy and M. Musen, *Anchor-PROMPT: Using Non-Local Context for Semantic Matching*, IJCAI Workshop on Ontologies and Information Sharing, Seattle (WA US), 2001, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.7666 (checked online February-04-2009).

[11] J. Euzenat and P. Valtchev, *Similartiy-based Ontology Alignment in OWL-Lite*, ECAI European Conference on Artificial Intelligence, Valencia (Spain), 2004, http://www.citeulike.org/user/miguelfm/article/832236 (checked online May-07-2009).

[12] P. Resnik, *Using Information Content to Evaluate Semantic Similarity in a Taxonomy*, 14th International Joint Conference on Artificial Intelligence, Adelaide (Australia), 2001, http://arxiv.org/abs/cmp-lg/9511007 (checked online February-23-2009).

[13] S. Melnik, H. Garcia-Molina and E. Rahm, *Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching*, 18th ICDE International Conference on Data Engineering, San Jose (CA US), 2002, http://ilpubs.stanford.edu:8090/730/ (checked online September-10-2009).

[14] N. F. Noy and M. Musen, *The PROMPT Suite: Interacitve Tools for Ontology Merging and Mapping*, International Journal of Human-Computer Studies, Elsevier Ltd., Stanford (CA US), 2003, pp. 983-1024, http://portal.acm.org/citation.cfm?id=965952 (checked online April-14-2009).

[15] K. Kotis and M. Lanzenberger, *Ontology Matching: Status and Challenges*, IEEE Intelligent Systems, AAAI sponsered Journal, New York (NY US), 2008, pp. 84-85.

[16] S. M. Falconer, N. F. Noy and M. A. Storey, *Ontology Mapping — A User Survey*, Proc. 2nd Intl Workshop Ontology Matching (OM 07), CEUR-WS, 2007, vol. 304, 2007, http://www.dit.unit.it/~p2p/OM-2007/5-446ontology_mapping_survey.pdf (checked online September-27-2009).

[17] G. Wu, J. Li, L. Feng and K. Wang, *Identifying Potentially Important Concepts and Relations in an Ontology*, 7th International Semantic Web Conference (ISWC 08), Karlsruhe (Germany), 2008, http://data.semanticweb.org/conference/iswc/2008/paper/research/63/html (checked online November-05-2009).

[18] OWL Web Ontology Language Guide, W3C Recommendation 10 February 2004, http://www.w3.org/TR/owl-guide/ (checked online September-20-2009).

[19] Protégé: Open Source Ontology Editor and Knowledge-base Framework developed by Stanford Center for Biomedical Informations Research at the Stanford University, Stanford (CA US), http://protege.stanford.edu/ (checked online June-18-2009).

[20] Ontology Alignment Evaluation Initiative - OAEI-2009 Campaign, Conference track, http://nb.vse.cz/~svabo/oaei2009/ (checked online August-28-2009).