Agent-Based Implementation for the Discovery of Structural Difference in OWL-DL Ontologies

Maxim Davidovsky[1], Vadim Ermolayev[2] and Vyacheslav Tolok[3],

[1] Zaporozhye National University, Center of Information Technologies, Zhukovskogo Str. 66,
69600 Zaporozhye, Ukraine
[2] Zaporozhye National University, Department of Information Technologies, Zhukovskogo Str. 66,
69600 Zaporozhye, Ukraine
[3] Zaporozhye National University, Department of Mathematical Modelling, Zhukovskogo Str. 66,
69600 Zaporozhye, Ukraine

m.davidovsky@gmail.com, vadim@ermolayev.com, vyacheslav-tolok@yandex.ru

Abstract. Various knowledge-based information systems contain distinct knowledge representations reflecting different domains of interest and different viewpoints across domains of discourse. For efficient use of knowledge-based systems it is necessary to know semantic relations or alignment between different knowledge representations. One of the promising approaches is the use of intelligent software agents where agents communicate in order to align respective knowledge representations. The paper presents an approach for ontology alignment based on implementation of meaning negotiation between intelligent agents. In the approach, negotiation is conducted in iterative way. At each step agents compare ontological contexts and use propositional substitutions in order to reduce semantic distance between the contexts. The focus of the paper is the implementation of agents' negotiation strategy.

Keywords: Ontology, ontology alignment, intelligent agent, meaning negotiation, implementation.

1 Introduction

Complexity and dynamism of modeling domains lead to heterogeneous and polysemous knowledge models and representations. Hence, knowledge-based information systems inevitably face with knowledge heterogeneity problem. Today the mainstream of knowledge representation is ontologies. Thus, for efficient use of knowledge-based systems it is necessary to know semantic relations between knowledge representations or alignment between respective ontologies. A perspective approach for ontology alignment is the one based on the use of intelligent software agents.

The approach is promising both from the efficiency standpoint and in terms of appropriateness for a broad range of applications. Today, ontologies are widely used for knowledge management, information retrieval and sharing, e-commerce, and many other applications. However, ontologies are often used in the open, decentralized and distributed systems where different nodes possess distinct though semantically overlapping ontologies. Also, ontologies can evolve independently of each other. Thus, for the interaction between parts of a distributed system and to ensure proper operation of the entire system the harmonization of ontologies must be implemented. In such circumstances, the alignment should be carried out in automatic mode. Moreover, the alignment should be performed dynamically in response to changes of the system, or "on demand" – during the interaction of previously non-interacting parties or those which do not have permanent relation or contact. The capability of migration of system nodes according to the structure changes over time is one more important aspect. It is obvious that the above requirements demand some intelligence from the parties involved in the alignment. Moreover, it must be “distributed”, “mobile” and “dynamic” intelligence. To the authors' opinion, such direction of research within the bounds of distributed artificial intelligence as the agent paradigm is one of the most consistent with and appropriate to the aforementioned requirements.

Another topical issue is the use of ontologies in such an open and decentralized system as the Semantic Web (SW) [1]. Ontologies are one of the pillars of the SW where they are used as the structural frameworks for knowledge representation and organizing of information. Another important component of the SW is the use of intelligent software agents which also makes the use of the agent paradigm the natural and organic means for aligning ontologies in the SW. Thus, intelligent agents and Semantic Web services that serve as a functional framework of the SW can interact seamlessly by aligning respective ontologies or asking for alignment services to the third party agent(s). A more detailed analysis of the use cases and respective specific requirements for ontology alignment is presented in our recent paper [2] as well as several known to date agent-based solutions.

In this paper we present an approach for ontology alignment based on implementation of meaning negotiation between intelligent agents. The negotiation strategy implies aligning ontologies by parts (conceptual subgraphs or contexts) that are relevant to a particular negotiation encounter. Negotiation is conducted in an iterative manner and is aimed at the reduction of a semantic distance between the contexts. Agents use propositional substitutions which may reduce the distance and support them with argumentation. The process is stopped when the distance reaches some commonly accepted threshold or the parties exhaust their propositions and arguments.

The paper is organized in the following way. In section 2 we describe the entire alignment process by using a characteristic task of ontology instance migration. Section 3 is devoted to implementation details. Section 4 outlines the set-up for future evaluation experiments. And the 5th section concludes the paper with a brief description of intermediate results and our plans for future work.

2 Ontology Alignment Process

Ontology alignment process (referred also as ontology matching [3]) is a process of discovering the correspondences (or mappings) between the elements of different ontologies. In the presented agent-based implementation the process embodies the strategy of automated meaning negotiation proposed by Ermolayev et al. in [4] for efficient information retrieval in open and distributed environments. The presented work is a part of the research activity aiming at development and implementation of an efficient methodology of ontology alignment and instance migration in decentralized settings (see also [5] for details on instance migration part of the methodology). The methodology (depicted on figure 1) assumes (semi-) automated iterative process of ontology alignment and instance migration with possible human intervention for checking the correctness and setting up the process. The presented implementation realizes the first necessary step of the methodology – discovery of mappings between respective ontological elements and presentation of results (r.t. TaskKind shapes No I and II on the figure) in the form suitable for further (automatic) processing and (semi-automated or manual) checking and correction.

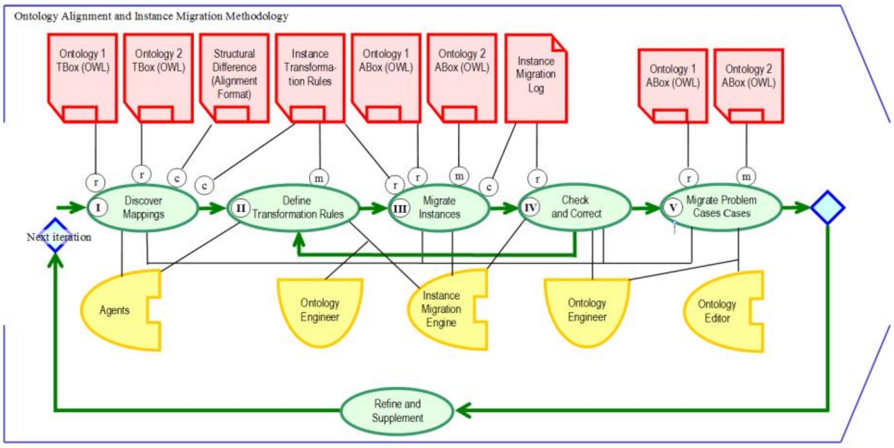

Fig. 1. Ontology alignment and instance migration methodology specified in ISO/IEC 24744 notation for describing methodologies [6].

Let's consider the task of ontology instance migration – a typical and characteristic case for knowledge sharing and reuse scenarios. Suppose we have an ontology O_1 conceptualizing knowledge of a domain P. In general, an ontology, as a rule, contains terminological part (or TBox) and assertional part (or ABox). TBox represents conceptual knowledge of a domain. ABox is a set of facts about the domain or, in other words, the instances of respective concepts and concrete values of their properties. Hence, we can represent the ontology O_1 as a tuple $O_1 = \langle C_1, I_1 \rangle$, where C_1 is a set of classes and respective properties representing concepts and relationships between them. I_1 is a set of instances.

Assume we have some ontology $O_2 = \langle C_2, I_2 \rangle$ reflecting some other perspective of the domain P or modeling some other domain $\tilde{P}$ semantically overlapping with P. Thus, O_2 contains the sets $C_2 \neq C_1 \neq \emptyset$ and $I_2 \neq I_1$. Obviously, for more complete and efficient use of knowledge we have to enrich the ontology O_2 with assertional knowledge of O_1 (i.e., some set of instances $\widetilde{I_1} \subseteq I_1$) that is relevant and valid for O_2.

We also assume that the ontologies O_1 and O_2 exist in decentralized settings and are associated with some operational agents Ag_1 and Ag_2 respectively. Then the process of replenishing knowledge of the ontology O_2 consists of two steps: (1) matching of terminological parts C_1 and C_2 by the agents and obtain some structural difference ΔC; and (2) being based on the structural delta migrate the set of instances $\widetilde{I_1}$, and obtaining a new set of instances $\widetilde{I_2} = I_2 \cup \widetilde{I_1}$ for ontology O_2.

Consider the first step of the process (Fig. 2). At the first step the agent Ag_2 initiates the process of ontology alignment and starts negotiation with the agent Ag_1. Then Ag_2 iterates over all concepts $c2_n \in C_2$ along with the respective contexts $Ctx2_k$ (where $n = \overline{1, N}$, $k = \overline{1, K}$, N and K are the numbers of concepts and contexts in O_2 respectively). Here, we define a context of concept c as a tuple $\langle \widehat{C}, \hat{I} \rangle$, where $\hat{C}$ is a set of TBox elements directly related with c, and $\hat{I}$ is a set of respective ABox elements. Then Ag_2 sends the contexts to the agent Ag_1. Agent Ag_1 applies the obtained context to the ontology O_1 and forms hypotheses of equivalence of the obtained concept and concepts of the ontology O_1. The hypotheses are weighted with confidence ratios on the basis of the measurement of contexts' semantic distances SD, which are calculated as sums of the semantic distances between the respective contexts' constituents. To calculate the semantic differences SD the variety of metrics can be used (e.g., see an overview of such metrics in [7]). In our implementation SD is implemented as an interface (see Section 3 Implementation) that allows substituting different concrete implementations of SD or their combination for more efficient discovering of correspondences.

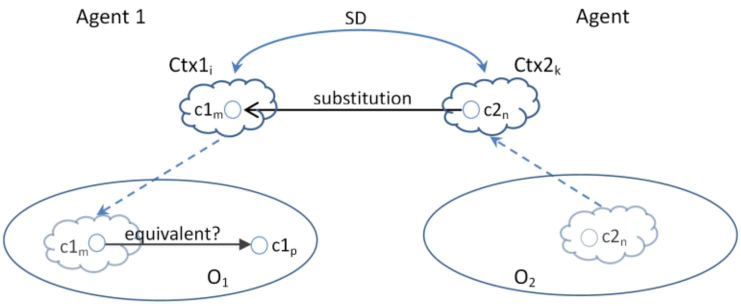

Fig. 2. Illustration of the agent-based ontology alignment process.

To test the approach we used the following metrics:

1. Lexical metric. Let R_A, R_B be the sets of roots of the words which constitute the names of concepts A and B respectively, then lexical similarity may be computed as:

$$Sim_L = \frac{R_A \cap R_B}{R_A \cup R_B} \tag{1}$$

2. Instance similarity metric. Two concepts, A and B are similar if $I_A \cap I_B \neq \emptyset$, where I_A and I_B are the sets of instances of A and B respectively. Instance Similarity is measured by symmetric Jaccard coefficient [8]:

$$Sim_I(A, B) = \frac{P(I_A \cap I_B)}{P(I_A \cup I_B)} \tag{2}$$

where $P(I)$ is the probability that a randomly chosen instance belongs to I.

3. Contextual or Feature Similarity. Similarity between feature sets may be computed by Tversky metrics [9]. The set of similarity measures of the object properties and related concepts s_j used as the feature set. Hence, feature similarity may be computed as an integrative metric for a pair of concepts as follows:

$$Sim_C = \frac{1}{m} \sum_{j=1}^{m} s_j \tag{3}$$

The particular negotiation round stops when the agents exhaust their argumentation and all of the respective contexts' constituents have been used. The whole process stops when there are no more contexts for all agents according to the alignment task.

Further, agents produce output of the resulting alignment of the two kinds (Fig. 2). The first type of result is a representation of the alignment in the Alignment format[1] [10]. This representation is quite intuitive and allows to conveniently assess the correctness and completeness of the resulting alignment. Also, this format is the de facto standard for representing alignments for participants of OAEI[2] and the presentation of results in this format allows us to estimate the system using OAEI benchmarks and makes the results comparable to those provided by OAEI. The second type of output is the representation of the alignment in the form of structural difference ΔC and generation of respective transformation rules for the next step – the migration of ontology instances (the problem statement of ontology instance migration and the description of solution based on transformation patterns and rules are described in detail in [5]).

[1] http://alignapi.gforge.inria.fr/format.html

[2] http://oaei.ontologymatching.org/

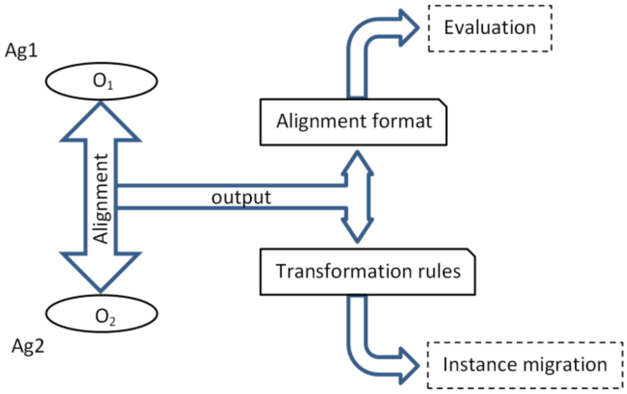

Fig. 3. Two kinds of alignment representation.

At the second step agents use Instance Migration Engine in order to transfer instances between ontologies O_1 and O_2 based on the transformation rules generated at the first step. Problem cases occurring during the migration are recorded into the migration log (possible problems and collisions during the migration are considered and analyzed in [11] and [5]). Finally, the target ontology agent asks a reasoner for the consistency check of the obtained ontology O_2. Entities that break the consistency are not added and are written to the log. All problem cases can be analyzed and resolved in subsequent iterations (if any) of the ontology alignment process.

3 Implementation details

To implement the described ontology alignment process we have developed a system of two intelligent agents. We use JADE[3] (Java Agent Development Environment) as an agent platform. Each agent contains an ontology processing module – ontology handler, which allows to parse an ontology and iterate over ontological elements and a context handler for the management of ontological contexts. Structural delta between the contexts is determined by the use of special determinator and computed based on various implementations of semantic distance metrics. The Agents carry out the alignment of ontologies presented in the OWL[4] language and use the OWL API[5] for coping with OWL-ontologies. In Fig. 4 is depicted a UML diagram fragment of the proof-of-concept software (please note that the figure shows not all the classes and methods in details, but the functional skeleton of classes and interfaces for a better understanding of the relationships between them).

[3] http://jade.tilab.com/
[4] http://www.w3.org/TR/owl2-overview/
[5] http://owlapi.sourceforge.net/

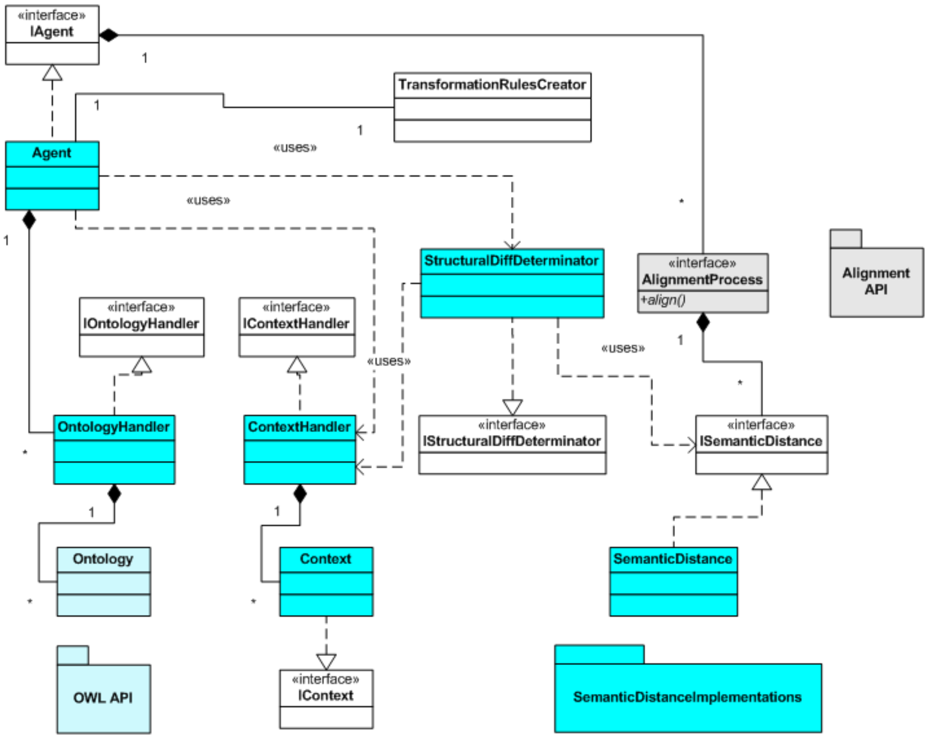

Fig. 4. An implementation UML diagram fragment.

The Ontology Alignment API [10] has been used for representing alignments in the Alignment format. The API provides a set of interfaces that can be implemented using different methods and metrics. A typical example of such an interface is the *AlignmentProcess* interface that provides an abstract method *align()* which can be implemented by special ontology matching techniques. In our implementation, the agents exchange ontological contexts rather than the whole ontologies, and the method implements alignment of individual contexts. The resulting alignment is obtained by combining the alignments obtained in separate negotiation iterations and applying the cut-off *trim()* operation. This operation cuts off the alignment in accordance with a predetermined threshold, removing the elementary alignments with confidence values below the threshold. As noted above, on the basis of the obtained alignment an agent forms the set of transformation rules to convert and transfer the ontology instances. More information about the transformation pattern and rule language can be found in [5]. For consistency checking in our implementation we use the Pellet reasoner[6].

[6] http://clarkparsia.com/pellet/

4 Evaluation set-up

The implemented software is planned to be evaluated for quality and completeness of the resulted alignments. The experimental set-up is depicted in Fig. 4. According to the two-step character of the alignment process we can measure the quality after each step on separate iterations. For these purposes we plan to use *Precision*, *Recall* and *F-measure* metrics (the description of these metrics for measuring the quality and completeness of ontology instance migration is given in [5]; the use of the metrics for assess ontology alignments is quite similar). We plan to use OAEI ontologies and datasets as a test-bed for our experiments.

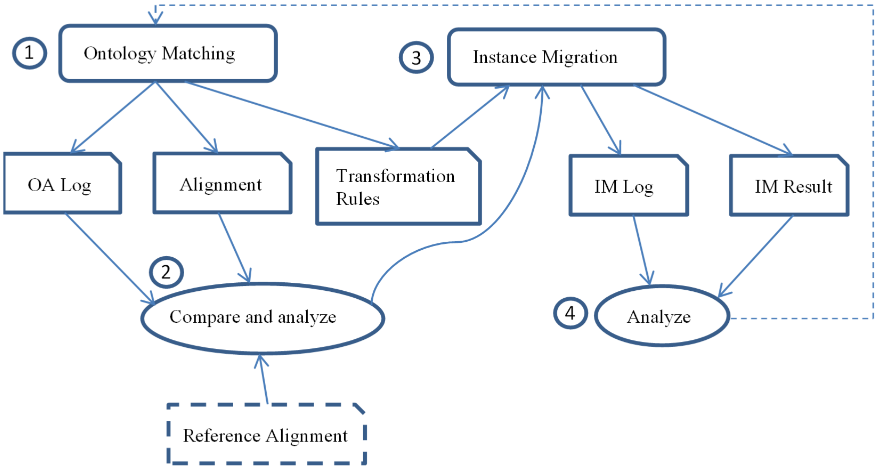

Fig. 5. Set-up for ontology alignment evaluation experiment (used abbrs.: OA – Ontology Alignment, IM – Instance Migration).

After the first step we evaluate the obtained alignments by comparing them with the reference ones. Then we proceed with instance migration and consequently analyze the obtained results. Being based on the analysis we can refine the set of alignments. Thus, we can assess the effectiveness of separate software components (used for matching and instance migration subtasks) as well as the iterative methodology in general.

5 Conclusions and future work

The paper presents the implementation of agent-based ontology alignment. The implementation is a continuation of work on realization of the iterative ontology alignment and instance migration methodology with the use of intelligent software agents. This methodology provides an apparatus for solving knowledge heterogeneity problem in decentralized environment and addresses a number of problems related to knowledge sharing and reuse. In the future we plan to conduct series of experiments with the developed software.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web (Berners-Lee et. al 2001). Scientific American, vol. 284, 28-37 (2001)
2. Davidovsky, M., Ermolayev, V., Tolok, V.: A Survey on Agent-Based Ontology Alignment. In: Proc. Int Conf on Agents and Artificial Intelligence (ICAART 2012), Vilamoura, Algarve, Portugal, 6-8 Feb. 2012, pp. 355-361 (2012)
3. Euzenat J., Shvaiko P.: Ontology Matching. Berlin Heidelberg (DE), Springer-Verlag (2007)
4. Ermolayev V., Keberle, N., Matzke, W.-E., Vladimirov, V.: A Strategy for Automated Meaning Negotiation in Distributed Information Retrieval. In: Y. Gil et al. (Eds.): ISWC 2005 Proc. 4th Int. Semantic Web Conference (ISWC'05), 6-10 November, Galway, Ireland. LNCS 3729, pp. 201 – 215 (2005)
5. Davidovsky, M., Ermolayev, V., Tolok, V.: Instance Migration Between Ontologies having Structural Differences. International Journal on Artificial Intelligence Tools. Vol. 20, No. 6, 1127–1156 (2011)
6. Henderson-Sellers, B. Gonzalez-Perez, C.: Standardizing Methodology Metamodelling and Notation: An ISO Exemplar. In: UNISCON 2008, eds. R. Kaschek, C. Kop, C. Steinberger, and G. Fliedl, LNBIP, Vol. 5, pp. 1–12. Springer, Berlin/Heidelberg, (2008)
7. Euzenat, J., Le Bach, T., Barrasa, J., Bouquet, P., De Bo, J., Dieng, R., Ehrig, M., Hauswirth, M., Jarrar, M., Lara, R., Maynard, D., Napoli, A., Stamou, G., Stuckenschmidt, H., Shvaiko, P., Tessaris, S., Van Acker, S., and Zaihrayeu, I.: State of the art on ontology alignment. KnowledgeWeb project deliverable D2.2.3, v.1.2. August 2, 2004. URL: http://knowledgeweb.semanticweb.org/ (2004)
8. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to match ontologies on the Semantic Web. International Journal on Very Large Data Bases, vol. 12(4), 303-319 (2003)
9. Tversky, A.: Features of Similarity. Psychological Review, vol. 84(4) (July 1977), 327-352 (1977)
10. David, J., Euzenat, J., Scharffe, F., Trojahn dos Santos, C.: The Alignment API 4.0. Semantic Web, vol. 2(1), 3-10 (2011)
11. Davidovsky, M., Ermolayev, V., Jentzsch, E., Matzke, W.-E.: Evaluation of Semi-Automated Ontology Instance Migration. In: M. Essaaidi et al. (Eds.): Intelligent Distributed Computing IV, SCI 315, pp. 179–190, Springer-Verlag Berlin Heidelberg (2010)