Ontology Alignment and Applications in 90 Minutes

Vadim Ermolayev[1] and Maxim Davidovsky[1]

[1] Department of IT, Zaporozhye National University
Zhukovskogo st. 66, 69063 Zaporozhye, Ukraine

vadim@ermolayev.com, m.davidovsky@gmail.com

Abstract. In this paper, we describe the structure and outline the content of a short tutorial on Ontology Alignment. The tutorial is planned in three parts within an overall timeframe of 90 minutes. Part 1 covers the fundamentals of ontology alignment and offers basic definitions, problem statements and problem classification based on the span, dynamics, direction, and distribution settings. This material is illustrated by: (i) using a walkthrough example of two elementary ontologies in Bibliographics domain; and (ii) offering a deeper discussion of one of the exemplar problems of ontology alignment – ontology instance migration – which has a practical utility for real world applications. The second part presents a software solution for ontology instance migration problem. The solution is demonstrated on the pair of Bibliographic ontologies of our walkthrough example. Part 3 puts ontology alignment in the context of several categories of applications which are important for the industries and the knowledge economy as a whole. The applications of ontology alignment in those categories are overviewed and requirements to the solutions are extracted.

Keywords. Ontology, ontology alignment, knowledge-based application, agent, argumentation, negotiation, information flow, ontology instance migration

Key terms. KnowledgeRepresentation, KnowledgeManagementMethodology, KnowledgeManagementProcess, KnowledgeTechnology, ICTTool

1 Introduction

This paper outlines the tutorial on the basics and problems of Ontology Alignment. The material is illustrated by our agent-based solution for ontology instance migration problem – one of practically important sub-problems in ontology alignment. The demand for applications of ontology alignment in real world applications is also presented. The tutorial, though given for the first time, is based in parts on our previous tutorial on Agent-Based Ontology Alignment [1]. This tutorial differs from [1] in the following: (i) it is broader in scope as covers not only agent-based approaches to align ontologies; and (ii) it is more oriented to reviewing industrial applications of ontology alignment and analyzing their requirements to the technology.

1.1 Structure and Timeframe

Part 1 targets a broad audience of those who are interested in the problems of ontology alignment in general and starts at a relatively basic level. It begins with informal definition of ontology alignment and puts the problem into the context of the other knowledge harmonization and integration problems. It further explains the motivation to study the methods of ontology alignment. Further the basic formalisms for ontology alignment are introduced and explained using an incremental approach. The generic ontology alignment problem is stated first and illustrated by the walkthrough example. This generic problem statement is further refined by offering a classification of the types of ontology alignment problems. A particular attention is paid to the ontology instance migration problem as a sub-problem of ontology alignment. The time frame for the first part of the tutorial is 30 minutes[1]. A standard configuration of presentation equipment is required: 1 beamer, 1 presentation screen, 1 microphone for the presenter, 1 additional microphone for the questions from the audience.

Part 2 offers a more practical material as it is focused on the presentation of the agent-based software solution for the ontology instance migration problem. The material of this part covers the presentation of the: (i) solution architecture; (ii) methodology shaping out the workflow; (iii) software demonstration that migrates instances from one to the other ontology of our walkthrough example. The time frame for the second part of the tutorial is also 30 minutes. Part 2 uses two independent presentation channels: one for the tutor and the other for software demonstration. Therefore it requires an enhanced configuration of presentation equipment: 2 beamers, 2 presentation screens, 2 microphones for the presenters, 1 additional microphone for the questions from the audience.

Part 3 is focused on the discussion of the importance of ontology alignment technology for real world applications. It starts with revisiting the motives to have this technology in place and proves the necessity of having the solutions for several categories of ICT applications, particularly in information and knowledge processing. In fact a review of applications, their specific requirements, and available solutions is given in this concluding part of the tutorial to provide a holistic, cross-domain view on the role of ontology alignment as a fundamental technology for today's knowledge economy. Similarly to parts 1 and 2, the time frame for part 3 of the tutorial is 30 minutes. Similarly to part 1, part 3 requires a standard set of presentation equipment.

The whole tutorial is therefore given in 90 minutes. A small break could be planned after Part 2 if the audience wishes to do so for having some discussions or posing in-depth questions. Though questions are allowed to be posed at any time, all three parts are planned with 5-minute question and answer sessions at their ends.

[1] Timings are given approximately. Small deviations could occur depending on the number of questions coming from the audience.

1.2 A Walkthrough Problem and Example

An example problem of ontology alignment that is used throughout the tutorial for detailed discussions is the **ontology instance migration** problem. The problem statement for ontology instance migration is presented in Section 2. The approach and software for solving this problem is demonstrated in Section 3. The applications that require the migration of ontology instances are mentioned among those discussed in Section 4.

Besides that, a very simple and artificial example of two different `Biblio` ontologies is used for illustrations throughout the tutorial. The structural schemas and assertional parts of these ontologies are provided in the support material at http://isrg.kit.znu.edu.ua/a-boa/index.php/A-BOA_Walkthrough_Problem_and_Example.

1.3 Support Materials, Discussions, and Contributions

For additional support materials a reader is advised to visit the A-BOA Wiki (isrg.kit.znu.edu.ua/A-BOA/) which has been developed for our previous tutorial on Agent-Based Ontology Alignment [1]. A-BOA Wiki is a Semantic MediaWiki based collaborative platform and a resource providing teaching content and discussion functionality.

1.4 Motivation to Study Ontology Alignment

The world around us is multi-faceted and polysemic in a sense that a model of the world developed in the mind of an individual or by a social group may be different from the model of the others. Knowledge-based systems reflect this fact in their knowledge representations. However, we do many things across several facets or even across subject domains. So, the knowledge representations of the corresponding facets of knowledge representation have to be brought into a harmonized or aligned state to enable proper communication, coordination or information processing.

`Biblio` ontologies give a simple example of such different facets, or knowledge representations, for the same body of knowledge about conference papers. Imagine that `Biblio-2` is the knowledge representation of a conference management system, while `Biblio-1` is the model for a paper repository used by a publisher for book production. The descriptions of the papers that have been accepted for a conference have to appear in the publisher's paper repository. Similarly, the publisher's information about the page limits has to be given to the conference management system to instruct the authors at proper time. Knowledge representations of `Biblio-1` and `Biblio-2` have therefore to be aligned for enabling seamless transformation and transfer of individual records between these two distributed knowledge-based systems. The tutorial will teach how such alignments could be done and what the complications in that activity are.

An attendee will learn that an alignment is essentially a result of applying a set of formal transformations to a knowledge representation – to its structure and individual assertions. An alignment allows interpreting knowledge that is external to the interpreter in the same way it interprets its own knowledge schema and assertions. For example, if an alignment of `Biblio-2` to `Biblio-1` exists, the publisher, who is the owner of `Biblio-1` may seamlessly import the assertions about the accepted papers to its production repository. Similarly, an alignment of `Biblio-1` to `Biblio-2` is required by conference organizers to get the publisher's information about publication constraints like page limits.

In a summary, ontology alignment has to be a technology at hand for all those who develop distributed constellations of knowledge-based systems that require collaboration across the nodes. Building ontology alignments efficiently and effectively is also important for the management and maintenance of such systems. Indeed, the fact that you have developed a perfect ontology alignment for your system does not yet allow you to retire. World changes and these changes are reflected in some facets of knowledge representations sporadically and without informing the other nodes. Hence the alignment activity has to be repeated in order to bring the whole system to a harmonized state.

2 Basics and Problems of Ontology Alignment

This section of the tutorial presents the formal problem statement and classification of ontology alignment problems, discusses one of the problem statements – for the ontology instance migration problem in more detail.

Following Euzenat and Shvaiko [2], an **ontology** is formally denoted as a tuple $O = \langle C, P, I, T, V, \leq, \perp, \in, = \rangle$ where C is the set of *concepts* (or *classes*); P is the set of *properties* (*object* and *datatype properties*); I is the set of *individuals*(or *instances*); T is the set of *datatypes*; V is the set of *values*; $\leq$ is a *reflexive*, *anti-symmetric* and *transitive relation* on $(C \times C) \cup (P \times P) \cup (T \times T)$ called *specialization*, that form partial orders on C and P called *concept hierarchy* and *property hierarchy* respectively; $\perp$ is an *irreflexive* and *symmetric relation* on $(C \times C) \cup (P \times P) \cup (T \times T)$ called *exclusion*; $\in$ is a relation over $(I \times C) \cup (V \times P)$ called *instantiation*; $=$ is a relation over $I \times P \times (I \cup V)$ called *assignment*; (the sets C, P, I, T, V are pairwise disjoint). It is also assumed (c.f. [3]), that an ontology O comprises its schema S and the assertional part A (see also Fig. 2):

$$O = \langle S, A \rangle; S = \langle C, P, T \rangle; A = \langle I, V \rangle \quad (1)$$

Ontology schema is also referred to as a **terminological component** (TBox). It contains the statements describing the concepts of *O*, the properties of those concepts, and the axioms over the schema constituents. The **set of individuals**, also referred to as an **assertional component** (ABox), is the set of the ground statements about the individuals and their attribution to the schema – i.e. where these individuals belong.

Ontology matching is denoted as a process of discovering the correspondences (or *mappings*) between the elements of different ontologies. A *mapping* (or a *mapping rule* [2]) is a tuple $m = \langle e, e', \Re, n \rangle$, where: e, e' are the elements of C, R, I, T, V of the respective ontologies O and O'; $\Re = \{\subset, \subseteq, \equiv, \supset, \supseteq\}$ is a set of relations; and n is a confidence value (typically in the range of $[0,1]$).

Finally, **ontology alignment** is denoted as the result of applying the discovered set of mapping rules to the respective ontologies. A generic ontology matching process and ontology alignment are described and pictured in more detail at http://isrg.kit.znu.edu.ua/a-boa/index.php/Basic_Definitions_and_Generic_Problem_Statement.

Based on the features of participating ontologies and the span of e, e' across C, P, I, T, V -s of O and O' a classification of the problems of finding ontology alignments could be outlined and formally stated. Graphical interpretation of some of these problems is described in more detail at http://isrg.kit.znu.edu.ua/a-boa/index.php/Classification_of_Ontology_Alignment_Problems. The dimensions along which the problems are classified are:

Complete (C), structural (S), or assertional (A) alignment
Static (S) versus dynamic (D) aligned ontologies
Bi-directional (B) versus uni-directional (U) alignment
Fully distributed (D) settings versus the presence of a central (C) referee ontology

A generic ontology alignment process may therefore be classified as a complete static bi-directional alignment using central referee ontology (CSBC). Our walk-through problem of ontology instance migration could be classified as assertional, static, uni-directional, distributed (ASUD) ontology alignment problem.

Yet another important feature for classifying ontology alignment processes is the presence of iterations for the refinement of alignments. All the processes discussed above are one-shot. However, the resulting alignments may appear to be of insufficient quality after their evaluation. Iterative ontology alignment processes aim at improving this shortcoming by incorporating the evaluation step and the refinement cycle in the process – please refer to (http://isrg.kit.znu.edu.ua/a-boa/index.php/Classification_of_Ontology_Alignment_Problems) for a graphical illustration. Iterative ontology instance migration process is discussed in more detail below. Our agent-based software prototype toolset for solving this problem is presented in Section 3.

One of the practically important ontology alignment problems, especially in fully distributed and dynamic settings, is the problem of transferring the individuals of one (source) ontology to the empty assertional part of the other (target) ontology [4].

Let us consider two arbitrary ontologies $O^s = (S^s, A^s)$ and $O^t = (S^t, A^t)$ conceptualizing the semantics of the same universe of discourse U – for example O^s and O^t are the two ontologies describing the same subject domain. U could be regarded as a collection of ground facts: $U = \{f\}$. Essentially, O^s and O^t are the interpretations of U. These ontologies would be considered identical if and only if:

$$\forall f \in U \; \mathrm{int}_{I^s}(f) \equiv \mathrm{int}_{I^t}(f) , \qquad (2)$$

where $\mathrm{int}_I(f)$ is the interpretation of the fact f by the individuals from I of ontology O.

Consequently, an abstract metric of interpretation difference $idiff(U, O^s, O^t)$ could be introduced. The value of *idiff* will be equal to zero for identical ontologies and will increase monotonically to one with the increase of the number of $f \in U$ such that $\neg(\mathrm{int}_{I_s}(f) \equiv \mathrm{int}_{I_t}(f))$. Hence, $idiff = 1$ iff $\forall f \in U \neg(\mathrm{int}_{I^s}(f) \equiv \mathrm{int}_{I^t}(f))$. $(1 - idiff)$ may further be interpreted [4] as balanced F-measure.

Ontologies O^s and O^t are structurally different if their schemas differ: $S^s \neq S^t$. This structural difference may be presented as a transformation $\mathrm{T} : S^s \to S^t$. Transformation T may be sought in the form of the set of nested transformation rules over the constituents of S^s resulting in the corresponding constituents of S^t.

Let us assume now that, given two structurally different ontologies O^s and O^t, the ABox of O^s contains individuals ($I^s \neq \varnothing$), while the ABox of O^t is empty ($I^t = \varnothing$). The problem of minimizing $idiff(U, O^s, O^t)$ by: (i) taking the individuals from I^s; (ii) transforming them correspondingly to the structural difference between O^s and O^t using T; and (iii) adding them to I^t – is denoted as **ontology instance migration problem**.

Theoretically ontology instance migration problem can be solved in one shot. In practice however each of the sub-tasks (ii-iii) may result in the loss of assertions [4]. Therefore an iterative refinement of the solution could yield results with a lower resulting *idiff* value. Hence, the problem has to be solved using an iterative ontology alignment process. Essentially, an iterative solution of ontology instance migration problem develops a sequence of O^s states $O^s_{st_i}$ in a way to minimize the $idiff(U, O^s, O^t)$ in a way that:

$$idiff(U, O^s_{st_i}, O_t) < idiff(U, O^s_{st_j}, O_t) \to i > j, \tag{3}$$

where: $O^s_{st_i}$ is O^s in the state after accomplishing iteration *i*; *i*, *j* are iteration numbers.

3 A Solution for Ontology Instance Migration Problem

This section demonstrates our agent-based solution for the ASUD ontology alignment problem stated above as **ontology instance migration problem**. This problem has been chosen as it possesses significant practical interest in real world applications, in particular for Ontology Engineering and Management in distributed and dynamic settings [4]. Instance migration in our solution is performed iteratively, so the alignment is refined from iteration to iteration.

Many influential publications, for example [5], envision that intelligent software components, like agents, need to be used together with ontologies for making semantic technologies accepted and effective in open and decentralized scenarios. For such agent based solutions, comprising industrial applications, the heterogeneity problem is the challenge that has to be faced. Ontology alignments are a means to solve the challenge. From the other hand agents, being the recipients of ontology alignment solutions, may help solving ontology alignment problems.

For a graphical illustration and more details of a simplified agent-based architecture for solving a generic ontology alignment problem please refer to http://isrg.kit.znu.edu.ua/a-boa/index.php/Theoretical_Foundations_and_Demonstration. The architecture introduces the wrapper agents W and W' for ontologies O and O' respectively. Agent R wraps the central referee ontology O^r and helps W and W' finding the proper mappings using O^r (a matchmaker function). W and W' produce their own sets of mappings M and M' in collaboration with each other (a fully distributed problem setting) or also in collaboration with R (the problem setting with a central referee ontology). At the Apply Mappings step M and M' are autonomously applied by W and W' to O and O'. A problem in developing such an agent-based solution is how do the agents collaborate and develop these mappings.

The presented solution is based on automated meaning negotiations between agents [6] as a way to discover structural differences between the schemas of O and O'. Similarly to [7], this approach aims at aligning ontologies by parts (*contexts*) that are relevant to a particular negotiation encounter. Negotiations imply iterative monotonic reduction of *semantic distances* between the contexts. An agent uses *propositional substitutions* which may reduce the distance and support them with *argumentation*. The process is stopped when the distance reaches a commonly accepted threshold or the involved parties exhaust their propositions and arguments. As opposed to the Argumentation Framework based approaches, this approach addresses the entire process of semantic reconciliation between ontologies and does not require off-the-shelf mappings.

The methodology used in our solution comprises several steps in the workflow. Steps (**I**) and (**II**) correspond to Discover Mappings, step (**III**) is for Applying Mappings, step (**IV**) corresponds to the step of evaluation and making decision about undertaking one more iteration. Iteration loop however does not involve mappings discovery in our solution. Instead, the mappings are revised manually by a knowledge engineer based on the list of migration failures in the migration log. Step (**V**), though important in practice, is not demonstrated.

`Biblio-1` and `Biblio-2` are used as examples of O and O'. The demonstrated agent-based solution is evaluated by comparing to our former work [4] where Ontology Difference Visualizer (ODV) tool [8] was used for discovering the structural difference between aligned ontologies.

Ontology instance migration process starts with the step (**I**) of discovering the structural difference between O and O'. Only TBoxes of the ontologies are used as the sources. Structural difference is discovered by the SDiff Discovery Engine (SDDE) [9] – a system of collaborative software agents negotiating on semantic contexts [10] for finding mappings $M' : S \rightarrow S'$. For demonstration purposes discovered structural difference is visualized using UML extension [8]. The mappings are further written down by SDDE as instance transformation rules [4] at the subsequent step (**II**). Instance Migration Engine (IME) is invoked at step (**III**) to perform the instance transformations according to these transformation rules. All the cases in which IME fails to perform the transformation are recorded to the instance migration log. Step (**IV**) involves a knowledge engineer who checks the migration log and decides if a refinement is required. If so, he starts the new iteration by refining the set of the transformation rules based on his analysis of the failure cases and using the rule editor of IMS at step (**II**). The refined set of rules is fed to the IMS at step (**III**). The loop continues until the knowledge engineer decides that further refinement is not possible, or all the instances of I^s are migrated to I^t .

4 Applications of Ontology Alignment

In this part of the tutorial a few selected categories of applications that require aligning information or knowledge representations are analyzed. A broader spectrum of applications is surveyed in [11]. In particular, attention is paid to the requirements related to ontology alignments that are posed by the applications in each category. A particular ontology alignment problem fitting to these requirements is also outlined.

A good survey of ontology-based applications is [12]. Ontology matching and alignment applications are discussed in [2]. Another comprehensive summary of ontology matching techniques and applications is [13]. In addition to these surveys, the publications surveying or reporting ontology alignment approaches are for example Chuttur [14], Vázquez-Naya et al. [15], Zhdanova et al. [16], Euzenat et al. [17]. Based on these inputs the following several typical application categories are analyzed in the tutorial with a focus on real world applications.

4.1 Distributed Information Retrieval

Distributed Information Retrieval (DIR) is an important category of applications that assist retrieving and fusing information from heterogeneous, distributed, and independent information resources. Ontologies in DIR are used for representing the structures of information at different nodes and for translating or transforming user queries and system responses. In particular, ontologies in DIR are important for extracting information or knowledge satisfying the semantics and the context of a user query. Ontology alignments are required:

- At query transformation step – for correlating query structure and semantics with different information resource schemas and metadata and building respective partial queries
- At query result fusion step – transforming and putting together the retrieved information instances

Hence, a solution of an SSUD ontology alignment problem is required for query transformation and of an ASUD problem for results fusion and delivery to a user. A **critical requirement** at the latter step **is high recall** as it is important not to miss any potentially relevant information while irrelevant individuals can be filtered out using other techniques. One more important requirement to an ontology alignment solution in DIR is its **scalability** in terms of the complexity and number of aligned ontologies.

4.2 Human-Machine Dialogues

Ontology alignments are used in human-machine interaction for providing mutual understanding between a user and a processing node. A software agent may represent a processing node in such interactions as an intelligent wrapper. Ontologies and their alignments can be used to obtain a formalizable set of requirements, structures, queries, etc. from informal or poorly structured user descriptions. As a rule such dialogs are run in iterative way. Hence, **iterative ontology alignment** methods fit to this category of applications better.

Brasoveanu et al. [18] argue the importance of using generic multimodal ontologies on the Semantic Web and propose an approach to enhance human-agent interaction based on multimodal ontologies. Guzzoni et al. [19] propose a toolkit-based approach for modeling human-agent interaction. Their toolset provides a means to model different aspects of an intelligent assistant such as: ontology-based knowledge structures; service-based primitive actions; composite processes and procedures; natural language and dialog structures. Tijerino et al. [20] report a framework for human-agent collaboration for the purpose of problem solving on the Semantic Web. In human-machine dialogue scenarios the most critical requirements are **adaptability**, **integrativity**, and **scalability** that allow enhancing human-machine mutual understanding.

4.3 Ontology Evolution, Versioning, Refinement

Ontology evolution, versioning, and refinement are important problems in Ontology Engineering (OE) and Management (KM) applications. Solutions are required for adequately representing knowledge in changing domains. Ontology alignment is one of the enabling technologies in these applications. Indeed, all three problems cope with transforming a source ontology revision to a target state (revision) that fits to the requirements causing the transition. Important aspects of this transition are that the target revision has to: (i) be consistent; (ii) re-use the source as much as allowed by the requirement of being consistent

Ontology alignments are used both to ensure consistency and maximal possible degree of re-use. Provided that the source revision is consistent, for proving that the resulting ontology revision is consistent it is sufficient to build the complete static bi-directional alignment (CSBC or CSBD problems). For the proper re-use of the source revision the solution of a uni-directional alignment problem will fit. For example a typical sub-task in an ontology refinement process is ontology instance migration from the source revision to the target revision [4]. A balanced **combination of** appropriately high **recall and precision** is an essential requirement for the instance migration solution.

4.4 Service Composition

The automation of web service composition or orchestration at run time is a challenging problem in Service Science which is intensively researched in the last decade. The complexity of the problem is caused by the inherent distributed character of software systems based on the use of services (for example Web services), the openness of these systems, and the dynamic character of their configurations and constellations. A sub-stream of research in the field develops the frameworks for services that intensively use ontologies as service descriptions – Semantic Web Services. Two prominent examples of these frameworks are OWL-S [21] and WSMO/L/X [22] which however do not fully solve runtime service composition problem. More advanced approaches exploit collaborative agents as service wrappers for managing services and service brokers or mediators for manipulating their descriptions in a runtime composition process (for example [23]). Like in Ontology Engineering and Management, a balanced **combination of** appropriately high **recall and precision** is an essential requirement for service composition. The **scalability** of the solution is also important.

The aspects of ontology reconciliation with respect to Web services and their composition are elaborated in [24, 25, 26]. An important requirement for such systems is the capability of adaptation and integration for providing compliant access and making the use of aggregate and atomic services more convenient.

5 Learning Outcomes

By the end of the tutorial the participants will:

- Learn the basics of ontology alignment that will enable them to understand the notions of an ontology, ontology mapping, the process of ontology matching, and the alignment as a result of matching process
- Learn the generic ontology alignment problem and the classification of its flavors based on the features of distributedness, the span of alignment, the direction of alignment, and the dynamic character of the source ontologies. Specifically, learn about the ontology instance migration problem as one of the ontology alignment problems.
- Be able to differentiate between one-shot and iterative ontology alignment methods and judge about the appropriateness of using this or that kind of a method in a particular setting
- Learn about one of the agent-based solutions for ontology alignment (ontology instance migration problem)
- Learn that ontology alignment is a very important, enabling technology for several kinds of the applications of distributed knowledge-based systems. In particular, learn which of the requirements of these applications make ontology alignment a challenging task.

References

1. Ermolayev, V., Davidovsky, M.: Agent-Based Ontology Alignment: Basics, Applications, Theoretical Foundations, and Demonstration. Tutorial Paper. In: Dan Burdescu, D., Akerkar, R., Badica, C. (eds.) Proc. WIMS 2012, 11-22, ACM (2012)

2. Euzenat J., Shvaiko P.: Ontology Matching. Berlin Heidelberg, Springer-Verlag (2007)
3. Nardi, D., Brachman, R. J.: An Introduction to Description Logics. In: Baader, F., Calvanese, D., McGuinness, D. L., Nardi, D., Patel-Schneider, P. F. (eds.) The Description Logic Handbook. Cambridge University Press New York, NY, USA (2007)
4. Davidovsky, M., Ermolayev, V., Tolok, V.: Instance Migration between Ontologies having Structural Differences. Int. J. on Art. Int. Tools. 20(6), 1127-1156 (2011)
5. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 284, 28–37 (2001)
6. 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)
7. Atencia, M., Schorlemmer, M.: Formalising Interaction-Situated Semantic Alignment: The communication product. In: Proc. of the Tenth International Symposium on Artificial Intelligence and Mathematics (ISAIM'08), Fort Lauderdale, Florida, USA (2008)
8. Ermolayev, V., Copylov, A., Keberle, N., Jentzsch, E., Matzke, W.-E.: Using Contexts in Ontology Structural Change Analysis. In: Ermolayev, V., Gomez-Perez, J.-M., Haase, P., Warren, P., (eds.) Proc. CIAO 2010, CEUR-WS, vol. 626 (2010)
9. Davidovsky, M., Ermolayev, V., Tolok, V.: Agent-Based Implementation for the Discovery of Structural Difference in OWL DL Ontologies. In: Mayr, H. C., Ginige, A., Liddle, S. (eds.) Proc. 4th Int. United Information Systems Conference (UNISCON 2012), LNBIP 137, Springer-Verlag, Berlin Heidelberg (2013)
10. Ermolayev, V., Ruiz, C., Tilly, M., Jentzsch, E., Gomez-Perez, G. M., Matzke, W.-E.: A Context Model for Knowledge Workers. In: Ermolayev, V., Gomez-Perez, J.-M., Haase, P., Warren, P. (eds.) Proc. CIAO 2010, CEUR-WS, vol. 626 (2010)
11. Davidovsky, M., Ermolayev, V., Tolok, V.: A Survey on Agent-Based Ontology Alignment. In: Proc. 4th Int. Conf. on Agents and Artificial Intelligence ICAART'12, pp. 355–361 (2012)
12. Gargantilla, J., Gomez-Perez, A.: OntoWeb: A Survey on Ontology-Based Applications. Deliverable 1.6. OntoWeb Consortium IST Project IST-2000-29243 (2004)
13. Scharffe, F., Euzenat, J., Le Duc, C., Mocan, A., Shvaiko, P.: Analysis of Knowledge Transformation and Merging Techniques and Implementations. KWEB/2004/D2.2.7/0.8 (2007)
14. Chuttur, M. Y.: Challenges Faced by Ontology Matching Techniques: Case Study of the OAEI Datasets. J. of Information Technology, 3(1), 33–42 (2011).
15. Vázquez-Naya, J. M., Romero, M. M., Loureiro, J. P., Sierra, A. P.: Ontology Alignment Overview. Encyclopedia of Artificial Intelligence 2009, pp. 1283–1289 (2009)
16. Zhdanova A,.V., de Bruijn, J., Zimmermann, K., Scharffe, F.: Ontology Alignment Solution. Deliverable D14 v2.0 (2004)
17. Euzenat, J., Laera, L., Tamma, V., Viollet, A.: Negotiation and Argumentation Techniques among Agents Complying to Different Ontologies. Deliverable D2.3.7, KWEB, v1.0 (2006)
18. Brasoveanu, A., Manolescu, A., Spânu, M. N.: Generic Multimodal Ontologies for Human-Agent Interaction. Int. J. of Computers, Communications & Control, 5(5), 625–633 (2010)
19. Guzzoni, D., Baur, C., Cheyer, A.: Modeling Human-Agent Interaction with Active Ontologies. Artificial Intelligence, SS-07-04, 52–59 (2007)
20. Tijerino, Y.A., Al-Muhammed, M., Embley, D.W.: Toward a Flexible Human-Agent Collaboration Framework with Mediating Domain Ontologies for the Semantic Web. In: Proc.

ISWC 2004 Workshop on Meaning Coordination and Negotiation, Hiroshima, Japan, pp. 131–142 (2004)
21. Martin, D., Paolucci, M., McIlraith, S., Burstein, M., McDermott, D., McGuinness, D., Parsia, B., Payne, T., Sabou, M., Solanki, M., Srinivasan, N., Sycara, K.: Bringing Semantics to Web Services: The OWL-S Approach. In: Cardoso, J., Sheth, A. (eds.) Proc. SWSWPC 2004, LNCS 3387, pp. 26–42 (2004)
22. Roman, D., Keller, U., Lausen, H., de Bruijn, J., Lara, R., Stollberg, M., Polleres, A., Feiera, C., Bussler, C., Fensel, D.: Web Service Modeling Ontology. Applied Ontology, 1(1), 77–106 (2005)
23. Ermolayev, V., Keberle, N., Kononenko, O., Plaksin, S., Terziyan, V.: Towards a Framework for Agent-Enabled Semantic Web Service Composition. Int. J. of Web Services Research, 1(3), 63–87 (2004)
24. Li, L., Yang, Y.: Agent Negotiation Based Ontology Refinement Process and Mechanisms for Service Applications. Service Oriented Computing and Applications, 2, 15–25 (2008)
25. Paurobally, S., Tamma, V., Wooldridge, M.: A Framework for Web Service Negotiation. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 2(4) (2007)
26. Huang, J., Zavala, R., Mendoza, B., Huhns, M. N.: Reconciling Agent Ontologies for Web Service Applications. In: Proc. of Multiagent System Technologies: Third German Conference (MATES-05). LNAI 3550, pp. 106–117 (2005)

Biographies

Vadim Ermolayev is an associate professor at the Department of Information Technologies (IT) of Zaporozhye National University and the lead of Intelligent Systems Research Group. He is also a research consultant in Semantic Technologies, Intelligent Software Systems, Distributed Artificial Intelligence. The research projects he took part in were focused on: intelligent systems and knowledge representations for enterprises; business and informal process dynamics; intelligent distributed information retrieval; the confluence of agent-based systems and Semantic Web services; ontology engineering, evolution, and refinement; performance management in engineering design. Alignment of knowledge representations was one of important topics in those projects.

Maxim Davidovsky is a PhD candidate at the Department of Mathematical Modeling (MM) of Zaporozhye National University. He also works for the Laboratory of Web-based Technologies and Distance Learning and is the member of Intelligent Systems Research Group at the Department of IT. Maxim received his MSc degree in applied mathematics and accomplished his postgraduate course in mathematical modeling and computational methods at the Department of MM. His research interests are in distributed and decentralized knowledge-based systems and software development. The focus of his current research activity is agent-based ontology alignment and instance migration specifically in distributed and decentralized settings.