



Triggering Ontology Alignment Revalidation Based on the Degree of Change Significance on the Ontology Concept Level

Adrianna Kozierekiewicz  and Marcin Pietranik  

Faculty of Computer Science and Management,
Wroclaw University of Science and Technology,
Wybrzeze Wyspianskiego 27, 50-370 Wroclaw, Poland
{adrianna.kozierekiewicz,marcin.pietranik}@pwr.edu.pl

Abstract. Following a common definition, ontologies can be seen as a formal specification of a conceptualisation. However, it cannot be expected that there will be no changes applied to them. Obviously, any application build on top of some ontology needs to adjust to the introduced alterations. For example, a mapping designated between two ontologies (also called an ontology alignment) is valid only if participating ontologies are fixed. In this paper we present a function that can indicate, whether or not, the aforementioned alignment needs updating, in order to follow modifications done to participating ontologies, and to avoid mapping them again from scratch.

Keywords: Ontology evolution · Ontology alignment · Knowledge management · Consensus theory

1 Introduction

Ontologies, following a common definition, can be seen as a formal specification of a conceptualisation, which translates to a formal description of some selected area of knowledge using a set of concepts and relationships that hold between them. They can be used to enrich data mining tools, algorithms of fraud detecting and semantic publishing. Their biggest downside is a problem of heterogeneity, which entails that between two independently created ontologies there is no possibility to develop any framework which could assert a consistency between them.

One of the approaches to overcoming this difficulty is finding which parts of ontologies define the same or similar parts of the aforementioned selected area of knowledge. This issue is a widely investigated topic, and in a literature, designating such mappings, is referred to as an ontology alignment [4]. Most of the available sources emphasise the complexity of this procedure. Therefore, it is not possible to relaunch a selected aligning algorithm whenever a mapped

ontology has changed in order to revalidate that the available mapping is still valid.

A frequent simplification of this issue is based on an assumption that ontologies do not change in time and their authors do not update their contents. However, it is obvious that in modern applications it is impossible to build any kind of flexible knowledge base build on such fixed foundations. In our current research we want to concentrate on managing alterations applied to ontologies on a concept level, and to investigate how they may influence an alignment initially established between them. We claim that not all modifications that appear during the ontology's lifespan are significant enough to entail invalidation of ontology alignment that has been previously designated. For example, note, that small changes concerning some concept's label are not equally important and influential as a major update of its structure.

Formally, the research task can be described as follows: *For a given ontology O in its two consecutive states in time, denoted as $O^{(m)}$ and $O^{(n)}$, one should determine a function Ψ_C representing the degree of significancy to which concepts within it have been changed in time.* Informally speaking - our main goal is developing a function that could be used as an indicator of having to check if the alignment of concepts at hand may potentially need revalidating, in the light of changes applied to maintained ontologies. Such measure can be confronted with some accepted threshold in order to ascertain the necessity of updating ontology alignment.

The article is structured as follows. In Sect. 2 an overview of related researches is given. Section 3 includes a mathematical foundation for our work (which involves basic definitions etc.). The main contribution can be found in Sect. 4 which contains a description of the developed function Ψ_C . Experimental evaluation can be found in Sect. 5. The paper ends in Sect. 6 with a summary and brief overview of our upcoming research plans.

2 Related Works

An ontology can be understood as a structure which allows to store and process some knowledge. If our knowledge is distributed in many sources, then an ontology integration process should be applied in order to allow reasoning about the whole available knowledge. For this task an alignment between input ontologies is required to conduct such integration process (also referred to as merging). However, knowledge stored in ontologies could be out of date and therefore, an update process is required. The modification of ontologies may entail changes in the existing alignment. To the best of our knowledge, problems referring to the ontology alignment evolution are not well investigated so far.

Zablith and others [17] divide an ontology evolution process into five sub-problems:

1. Detecting the need for evolution which initiates the ontology evolution process by detecting a need for change.

2. Suggesting change representations and suggests changes to be applied to the ontology
3. Validating changes filters out those changes that should not be added to the ontology as they could lead to an incoherent or inconsistent ontology, or an ontology that does not satisfy domain or application-specific constraints.
4. Assessing impact measures the impact on external artefacts that are dependent on the ontology or criteria such as costs and benefits of the proposed changes.
5. Managing changes applies and records changes and keeps track of the various versions of the ontology.

To assert a proper ontology evolution management, all of the mentioned above subtasks have to be solved. However, most of the research available in the literature focus on detecting and managing changes implemented in an ontology. For example, [2] is devoted to a repository of large ontologies. Authors propose an algorithm called the Ontology Version Detector, which implements a set of rules analysing and comparing URIs of ontologies to discover versioning relations between ontologies.

The paper [13] is especially addressing a problem of detecting changes between versions of the same knowledge bases. Authors propose a formal framework that was used for the definition of their language of changes. The detection semantics of the defined language is used as the basis for a change detection algorithm.

The practical idea of an ontology version management and change detection is raised in [8]. Authors designed a system OntoView which is able to store an ontology, provide a transparent interface to different versions, specify relations between versions of ontologies, identify scheme for ontologies, and finally helps users to manage changes in online ontologies.

The ontology evolution management system is also designed by Khattak and others [9]. Authors describe a change history management framework for evolving ontologies. The paper addresses several subproblems such as ontology versioning (also covered in [7]), tracking a change's provenance, a consistency assertion, a recovery procedure, a change representation and a visualisation of the ontology evolution. Experimental results show that the proposed system has better accuracy against other existing systems.

In [18] authors propose a framework called temporal OWL 2 (τ OWL), which supports a temporal schema versioning, by allowing changing these components and by keeping track of their evolution through the conventional schema versioning and annotating document versions, respectively. Some tools for managing a temporal versions of an ontology could be found also in [5, 16].

The problem of the ontology evolution involves measuring and managing changes applied within ontologies, and assessing its impact on mappings between such ontologies. However, in many papers the alignment evolution is pushed to a background and not considered properly. Authors of [15] noticed that alignments originally established between ontologies can become stale and invalid when certain changes have been applied to maintained ontologies. Thus, they propose a

preliminary algorithm for a revalidation and preserving correctness of alignment of two ontologies.

In [3] an original method for identifying the most relevant subset of concept's attributes, which is useful for interpreting the evolution of mappings under evolving ontologies is designed. Such solution aims at facilitating a maintenance of mappings based on the detected attributes.

The alignment evolution is addressed from the different point of view in [10]. In this paper, a solution that allows query answering in data integration systems under evolving ontologies without mapping redefinition is provided. It is achieved by rewriting queries among ontology versions and then forwarding them to the underlying data integration systems to be answered. The changes among ontology versions using a high level language of changes are detected and described. They are interpreted as sound global-as-view mappings, and are used in order to produce equivalent rewritings among ontology versions.

COnto-Diff [6] is a rule-based approach which detects high-level changes according to a dedicated language of changes. The detection process is coupled with a mapping between the elements (concepts, properties) of two ontology versions. The application of the detected modifications (and their inverses) is also considered in this work.

In many real approaches dedicated to the ontology evolution, when maintained ontologies change, the mappings between them are recreated from scratch or are adjusted manually, a process which is known to be error-prone and time-consuming. In this paper we propose a function representing the degree of change significance, which allows detecting outdated alignments. In consequence it will provide us with an ability of automatic alignments revalidation.

3 Basic Notions

In our research, we assume a following formal definition of an ontology:

$$O = (C, H, R^C, I, R^I) \quad (1)$$

where C is a finite set of concepts; H denotes a concepts' hierarchy; R^C is a finite set of relations between concepts $R^C = \{r_1^C, r_2^C, \dots, r_n^C\}$, $n \in N$, such that every $r_i^C \in R^C$ ($i \in [1, n]$) is a subset of $C \times C$; I represents a set of instances' identifiers; $R^I = \{r_1^I, r_2^I, \dots, r_n^I\}$ denotes a set of relations between concepts' instances.

By "a real world" we call a pair (A, V) , where A denotes a set of attributes and V is a set of valuations of these attributes (their domains). A concept's $c \in C$ structure from the (A, V) -based ontology is defined as:

$$c = (id^c, A^c, V^c, I^c) \quad (2)$$

where: id^c is it's unique identifier, A^c denotes a set of its attributes ($A^c \subseteq A$) with their domains included in the set V^c (formally: $V^c = \bigcup_{a \in A^c} V_a$ where V_a is a domain of an attribute a taken from the set V), and I^c is a set of assigned

instances. For simplicity, we write $a \in c$ to denote that an attribute a belongs to the concept c (formally: $a \in c \iff a \in A^c$).

To ascribe a meaning to attributes included in some concept, we assume an existence of a sub-language of the sentence calculus (denoted as L_s^A) and a function $S_A : A \times C \rightarrow L_s^A$, which assigns logic sentences to every attribute from a concept c . For example, an attribute *DateOfBirth* within a concept *Person* obtains the following semantics: $S_A(\text{DateOfBirth}, \text{Person}) : \text{birthYear} \wedge \text{birthMonth} \wedge \text{birthDay} \wedge \text{age}$.

The overall meaning of a concept (further referred to as its context) is defined as a conjunction of semantics of each of its attributes. Formally, for a concept c , such that $A^c = \{a_1, a_2, \dots, a_n\}$, its context is as follows $\text{ctx}(c) = S_A(a_1, c) \wedge S_A(a_2, c) \wedge \dots \wedge S_A(a_n, c)$.

Due to the fact, that in this article we focus only on the concept level of ontologies, we do not provide detailed definitions of remaining elements from the Eq. 1. For broader explanations, please refer to our previous publications, such as [11].

In order to track changes applied to ontologies, we accept a notion of a universal timeline, which can be understood as an ordered set of discrete moments in time: $\overline{TL} = \{t_n | n \in N\}$. $TL(O)$ denotes a subset of this timeline for a selected ontology - it contains only those elements of \overline{TL} for which the ontology O has changed. By using a superscript $O^{(m)} = (C^{(m)}, H^{(m)}, R^{C^{(m)}}, I^{(m)}, R^{I^{(m)}})$ we denote the ontology O in a given moment in time $t_m \in TL(O)$. We also introduce the notion $O^{(m-1)} \prec O^{(m)}$ which represents a fact that $O^{(m)}$ is a later version of O than $O^{(m-1)}$. For simplicity we extend this notation for particular elements of the given ontology, e.g. $c^{(m-1)} \prec c^{(m)}$ denotes that a concept c has at least two versions, and $c^{(m-1)}$ is earlier than $c^{(m)}$. On top of these definitions, we define a repository of an ontology O , which is an ordered set of its subsequent versions in time, formally defined as $\text{Rep}(O) = \left\{ O^{(m)} | \forall m \in TL(O) \right\}$.

Assuming an existence of two independent, (A, V) -based ontologies, O and O' , an alignment on a concept level between them is defined as a finite set $\text{Align}(O, O')$ containing tuples of the form:

$$(c, c', \lambda_C(c, c'), r) \tag{3}$$

where: c and c' are concepts from O and O' respectively, $\lambda_C(c, c')$ is a real value representing a degree to which the concept c can be mapped into the concept c' , and r is one of types of relation that connects c and c' (equivalency, generalisation or contradiction). $\lambda_C(c, c')$ can be calculated using one of the similarity methods taken from a very broad literature concerning the ontology alignment. A robust overview of a current state of the art in this field can be found in [1]. Due to its simplicity and flexibility, we can use this notion also for time-tracked ontologies. For example, $\text{Align}(O^{(m)}, O'^{(n)})$ is an alignment of the ontology O in a state it had in a moment m , and the ontology O' in a state from a moment n . Obviously, both $m, n \in \overline{TL}$.

4 Ontology Change Significance on the Concept's Level

In order to compare two states of a single ontology O we introduce a function $diff_C$ which, when fed with its two successive states $O^{(m-1)}$ and $O^{(m)}$ (such that $O^{(m-1)} \prec O^{(m)}$), generates three sets containing concepts added, deleted and somehow altered. Formally, these sets are defined below:

$$diff_C(O^{(m-1)}, O^{(m)}) = \left\langle \begin{aligned} &new_C(C^{(m-1)}, C^{(m)}), \\ &del_C(C^{(m-1)}, C^{(m)}), \\ &alt_C(C^{(m-1)}, C^{(m)}) \end{aligned} \right\rangle \quad (4)$$

where:

1. $new_C(C^{(m-1)}, C^{(m)}) = \left\{ c \mid c \in C^{(m)} \wedge c \notin C^{(m-1)} \right\}$
2. $del_C(C^{(m-1)}, C^{(m)}) = \left\{ c \mid c \in C^{(m-1)} \wedge c \notin C^{(m)} \right\}$
3. $alt_C(C^{(m-1)}, C^{(m)}) = \left\{ (c^{(m-1)}, c^{(m)}) \mid c^{(m-1)} \in C^{(m-1)} \wedge c^{(m)} \in C^{(m)} \wedge c^{(m-1)} \prec c^{(m)} \wedge (A^{c^{(m-1)}} \neq A^{c^{(m)}} \vee V^{c^{(m-1)}} \neq V^{c^{(m)}} \vee I^{c^{(m-1)}} \neq I^{c^{(m)}}) \vee ctx(c^{(m-1)}) \neq ctx(c^{(m)}) \right\}$

The first two descriptors in the definition above are self-explanatory. The last one represents alterations applied to concepts from $O^{(m-1)}$, as a set of pairs of concepts' versions, that have been neither added nor deleted, but differ structure-wise or in terms of their contexts.

The function $diff_C$ describes changes applied to a certain ontology, however, it does not show how significant they were. For an ontology $O = (C, H, R^C, I, R^I)$ in its two subsequent states $O^{(m-1)}$ and $O^{(m)}$, such that $O^{(m-1)} \prec O^{(m)}$, and a concept difference function $diff_C$ defined above, a function calculating a **degree of change significance on the level of concepts** has a following signature:

$$\Psi_C : C^{(m-1)} \times C^{(m)} \rightarrow [0, 1] \quad (5)$$

Such function must meet a following two postulates:

- **P1.** $\Psi_C(C^{(m-1)}, C^{(m)}) = 0 \iff diff_C(C^{(m-1)} \times C^{(m)}) = \langle \phi, \phi, \phi \rangle$
- **P2.** $\Psi_C(C^{(m-1)}, C^{(m)}) = 0 \iff del_C(C^{(m-1)}, C^{(m)}) = C^{(m-1)} \wedge alt_C(C^{(m-1)}, C^{(m)}) = \phi$

P1 states that the change significance is minimal if no alterations on the concept level have been applied. Namely, no new concepts have appeared, no concepts have been removed, no concepts have been changed.

P2 describes that the change significance is maximal if the ontology has been completely modified, meaning, every concept from the earlier state has been deleted, every concept in a later state is new (or nothing has been added), and therefore no concepts have been altered.

Having the above postulates in mind, we define the function Ψ_C as follows:

$$\Psi_C(C^{(m-1)}, C^{(m)}) = \frac{|new_C(C^{(m-1)}, C^{(m)})| + |del_C(C^{(m-1)}, C^{(m)})|}{|C^{(m)}| + |del_C(C^{(m-1)}, C^{(m)})|} + \sum_{(c_1, c_2) \in alt_C(C^{(m-1)}, C^{(m)})} d_s(ctx(c_1), ctx(c_2)) + \frac{}{|C^{(m)}| + |del_C(C^{(m-1)}, C^{(m)})|} \quad (6)$$

The function above is build from three components. The first two are cardinalities of sets describing new and removed concepts. The last component utilises a function d_s which calculates a distance between two logic formulas - we initially transform the passed formulas (concepts' contexts) to a conjunctive normal form and incorporate the Jaccard's measure to calculate the distance between them. For details please refer to our previous publication [14].

In the next section we will describe an experiment that we designed and conducted in order to verify a usefulness of the developed function Ψ_C along with an analysis of obtained results.

5 Experimental Verification

5.1 Experiment's Setup and Procedure

The ontology alignment is a frequently covered topic. Ontology Alignment Evaluation Initiative (OAEI) is an organisation which annually organises a campaign aiming at assessing strengths and weaknesses of ontology matching systems and comparing their performances [1]. Participants of these campaigns designate mappings between preprepared ontologies that, for logistical reasons, are grouped into groups called tracks. Within every track, for every ontology pair, OAEI provides a reference alignment, with which a collected mappings are compared using a variety of different measures.

In order to verify the usefulness of a function Ψ_C in detecting a necessity of potential revalidating an alignment that became stale due to the ontology evolution, we needed a robust dataset and an independent ontology alignment tool. We chose "a Conference Track" consisting of 16 ontologies describing the domain of organising conferences, that was used in the OAEI'2017 campaign. We also decided to base our experiment on LogMap [12], which is an ontology alignment and alignment repair system. It is a highly scalable ontology matching solution with an integrated reasoning and inconsistency repair capabilities. It is capable of extracting mappings between concepts, relations and instances. More importantly, LogMap earned high positions in subsequent OAEI campaigns.

The experiment was divided into two parts. The first one was aimed at showing how different modifications of an ontology that may appear during its evolution can affect its alignments. It consisted of following phases:

1. Select a source ontology (called *confOf*) and a target ontology (*CMT*) from a Conference Track of OAEI'2017 campaign.
2. Generate a base alignment between the two selected ontologies using LogMap.
3. Apply random modifications to the source ontology according to every alteration scenario from Table 1
4. For the two versions of the source ontology, calculate a value of Ψ_C .
5. Using LogMap generate a new alignment between a modified version of the source ontology and the target ontology.
6. Calculate a Dice coefficient measure between the base and the new alignment, in order to illustrate differences between alignments of ontologies that have changes over time. This measure has been chosen as the very intuitive functions allowing to compare dissimilarity between two sets. This value can clearly show changes within ontologies (which significancy can be calculated using Ψ_C function) affect mappings between ontologies.

The goal of the second part was showing how the developed function, calculating the degree of concepts' change significancy, behaves when different ontologies and their evolutions are processed. This phase of the experiment had a following steps:

1. Select a source ontology (*confOf*)
2. Using LogMap, generate base alignments between the source ontology and every other ontology from the track (presented in Table 2)
3. Apply a random modification relying on adding and removing 5 related concepts to the source ontology and calculate the value of Ψ_C . Obviously, in this part of the experiment, the value of Ψ_C is constant and equals 0.128.
4. Using LogMap, generate new alignments between the modified source ontology and every other ontology from the track (except *confOf*).
5. Calculate a Dice coefficient measures between the base alignment and the new alignments between the source ontology and every other ontology from the track collected in the previous step.

Results collected during both parts of the experiment, along with their statistical analysis, can be found in a next section of the paper.

5.2 Results of the Experiment

As it was mentioned in the previous section, the experiment has been divided into two parts. In the first one, we prepared some scenarios which apply all possible ontology changes on the concept level like: adding, removing or modifying some or all concepts.

Let us suppose that there exists some correlation between the degree of significancy to which concepts within maintained ontologies have been changed in

time and a Dice distance between the base and the new alignment. We confirmed this hypothesis using a statistical analysis of gathered data obtained using a procedure described in the previous section and presented in the Table 1.

Before selecting a proper statistical test, we analysed the distribution of obtained data using a Shapiro-Wilk test. Because for both samples p – values are greater than $\alpha = 0.05$, we rejected the null hypothesis and claim that samples do not come from a normal distribution. Next, we calculated Spearman’s rank correlation coefficient. Comparing the obtained p – value equals 0.00642 with the assumed significance level α , we could draw a conclusion that there is a monotonic dependence between Ψ_C and calculated values of the Dice measure (Fig. 1). This dependence is directly proportional. The Spearman’s rank correlation coefficient was equal 0.71 and can be interpreted as a strong, monotonic relation between the examined samples. It allows us to claim that, the developed function Ψ_C can serve as a trigger of alignment revalidation in case of significant change that may appear during the ontology evolution.

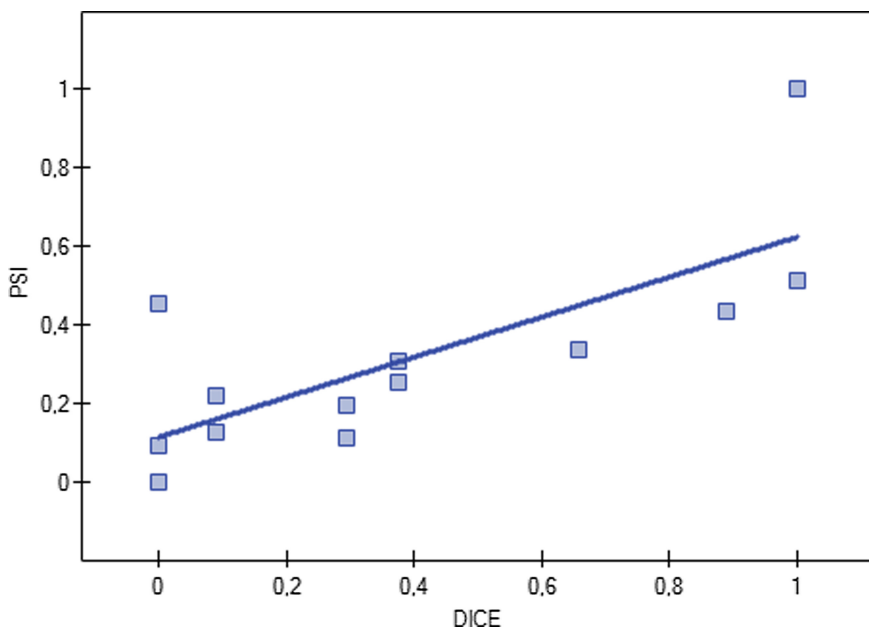


Fig. 1. A graphical representation of a monotonic dependence between Ψ_C and calculated values of the Dice measure

Based on the results presented in the Table 1, we observed that LogMap does not consider attributes and their modifications in its alignment determination process. Thus, for the second part of our experiment we focused only on adding and removing concepts for the chosen ontology where the Ψ_C was equal 0.294. Then, according to the experimental procedure described earlier, we calculated values of the Dice measure. Obtained results are shown in the Table 2.

Table 1. Different scenarios for a single pair of ontologies, 39 concepts in the base ontology *confOf*

No.	Description	Dice measure	Ψ_C
1	No changes	0	0
2	Removing 5 concepts	0.091	0.128
3	Adding 5 related concepts	0.294	0.114
4	Modifying 5 concepts	0	0.093
5	Adding and removing 5 concepts	0.375	0.256
6	Adding and modifying 5 concepts	0.294	0.196
7	Modifying and removing 5 concepts	0.091	0.221
8	Adding, removing and modifying 5 concepts	0.375	0.309
9	Removing 20 concepts	1.000	0.513
10	Adding 20 concepts	0.657	0.339
11	Modifying 20 concepts	0	0.455
12	Removing 40 concepts	1.000	1.000
13	Adding 40 concepts	0.889	0.435

For all target ontologies, the Dice coefficient value corresponds with Ψ_C , especially if a base alignment has a large number mappings. It is obvious, because larger alignments are proportionally less sensitive for changes appearing in participating ontologies. Therefore, we can draw a general conclusion that, based only on a value of Ψ_C (which can be calculated by processing a maintained, evolving ontology). It is possible to detect a necessity of significant changes that need to be applied in the corresponding alignments.

6 Future Works and Summary

An ontology integration task is a difficult, time-, and cost-consuming process. It starts with designating elements of ontologies that relate to the same objects from the selected universe of discourse. In a literature this task is called an ontology alignment. However, due to the fact that ontologies are complex structures, and in modern days it is not possible to assume that they won't change in time, therefore, predesignated mappings between two or more ontologies may become obsolete.

In this paper we have presented a component of an ontology alignment evolution framework. The developed tool can be used to check, whether or not, the aforementioned situation in which the mapping between ontologies is no longer valid. This trigger is build on top on an analysis of changes that appeared within ontologies over time. Its usefulness was proved on a basis of a statistical analysis of experimental results gathered from a procedure utilising a broadly accepted OAEI datasets, created for validating ontology alignment tools.

Table 2. The same modification scenario for different pairs of ontologies

Target ontology	Number of mapping in a base alignment	Dice measure
CMT	5	0.375
cocus	9	0.660
conference	14	0.226
confious	5	0.429
crs	6	0.375
edas	13	0.241
ekaw	15	0.278
iasted	4	0.385
linklings	6	0.474
micro	9	0.440
myReview	7	0.368
OpenConf	6	0.294
paperdyne	11	0.417
pcs	6	0.294
sigkdd	5	0.330

Our upcoming research plans are twofold. At first, we want to develop an algorithm that based solely on an analysis of an evolving ontology will be capable of updating existing mappings with other ontologies when such necessity occurs. Secondly, we will extend the created framework for other elements available within ontologies - relations and instances.

Acknowledgement. This research project was supported by grant No. 2017/26/D/ST6/00251 from the National Science Centre, Poland.

References

1. Achichi, M., et al.: Results of the ontology alignment evaluation initiative 2017. In: OM 2017–12th ISWC Workshop on Ontology Matching, pp. 61–113 (2017). No commercial editor
2. Allocca, C., d’Aquin, M., Motta, E.: Detecting different versions of ontologies in large ontology repositories. In: Proceedings of IWOD 2009, Washington, D.C., USA (2009)
3. Dinh, D., Dos Reis, J.C., Pruski, C., Da Silveira, M., Reynaud-Delaître, C.: Identifying relevant concept attributes to support mapping maintenance under ontology evolution. *Web Semant.: Sci. Serv. Agents World Wide Web* **29**, 53–66 (2014)
4. Euzenat, J.: Revision in networks of ontologies. *Artif. Intell.* **228**, 195–216 (2015)
5. Grandi, F.: Multi-temporal RDF ontology versioning. In: Proceedings of IWOD 2009, Washington, D.C., USA (2009)
6. Hartung, M., Groß, A., Rahm, E.: COnto-Diff: generation of complex evolution mappings for life science ontologies. *J. Biomed. Inform.* **46**(1), 15–32 (2013)

7. Heflin, J., Pan, Z.: A model theoretic semantics for ontology versioning. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 62–76. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30475-3_6
8. Klein, M., Fensel, D., Kiryakov, A., Ognyanov, D.: Ontology versioning and change detection on the web. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 197–212. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45810-7_20
9. Khattak, A.M., Latif, K., Lee, S.: Change management in evolving web ontologies. *Knowl.-Based Syst.* **37**, 1–18 (2013)
10. Kondylakis, H., Plexousakis, D.: Ontology evolution without tears. *J. Web Semant.* **19**, 42–58 (2013)
11. Kozierekiewicz, A., Pietranik, M.: The knowledge increase estimation framework for integration of ontology instances' relations. In: Lupeikiene, A., Vasilecas, O., Dzemyda, G. (eds.) DB&IS 2018. Communications in Computer and Information Science, vol. 838, pp. 172–186. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-97571-9_15
12. Jiménez-Ruiz, E., Cuenca Grau, B.: LogMap: logic-based and scalable ontology matching. In: Aroyo, L., et al. (eds.) ISWC 2011. LNCS, vol. 7031, pp. 273–288. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25073-6_18
13. Papavassiliou, V., Flouris, G., Fundulaki, I., Kotzinos, D., Christophides, V.: High-level change detection in RDF(S) KBs. *ACM Trans. Database Syst.* **38**(1), 1–42 (2013)
14. Pietranik, M., Nguyen, N.T.: A Multi-attribute based framework for ontology aligning. *Neurocomputing* **146**, 276–290 (2014). <https://doi.org/10.1016/j.neucom.2014.03.067>
15. Pietranik, M., Nguyen, N.T.: Framework for ontology evolution based on a multi-attribute alignment method. In: CYBCONF 2015, pp. 108–112 (2015). <https://doi.org/10.1109/CYBCConf.2015.7175915>
16. Sassi, N., Jaziri, W., Gargouri, F.: Z-based formalization of kits of changes to maintain ontology consistency. In: Proceedings of KEOD 2009, pp. 388–391 (2009)
17. Zablit, F., et al.: Ontology evolution: a process-centric survey. *Knowl. Eng. Rev.* **30**(1), 45–75 (2015)
18. Zekri, A., Brahmia, Z., Grandi, F., Bouaziz, R.: τ OWL: a systematic approach to temporal versioning of semantic web ontologies. *J. Data Semant.* **5**(3), 141–163 (2016)