

Alignment Conservativity Under the Ontology Change

Yahia Atig, EEDIS Laboratory, Sidi Bel Abbes University, Algeria & University Center of Naama, Algeria*

Ahmed Zahaf, University Dr. Tahar Moulay of Saida, Algeria

Djelloul Bouchiha, University Center of Naama, Algeria

Mimoun Malki, LabRI-SBA Laboratory, Ecole Supérieure en Informatique, Sidi Bel-Abbes, Algeria

ABSTRACT

Recently, many methods have appeared to solve the problem of the evolution of alignment under the change of ontologies. The main challenge for them is to maintain consistency of alignment after applying the change. An alignment is consistent if and only if the ontologies remain consistent even when used in conjunction with the alignment. The objective of this work is to take a step forward by considering the alignment evolution according to the conservativity principle under the change of ontologies. In this context, an alignment is conservative if the ontological change should not introduce new semantic relationships between concepts from one of the input ontologies. The authors give methods for the conservativity violation detection and repair under the change of ontologies and they carry out an experiment on a dataset adapted from the Ontology Alignment Evaluation Initiative. The experiment demonstrates both the practical applicability of the proposed approach and shows the limits of the alignment evolution methods compared to the alignment conservativity under the change of ontologies.

KEYWORDS

Alignment Evolution, Alignment Repair, Conservativity Principle Violations, Ontological Change

INTRODUCTION

The proliferation in the development of ontologies has led to the appearance of ontology repositories such as Bioportal¹ and AgroPortal² to store and share ontologies and alignments. The utility of these repositories depends not only on ontologies quality, but also on alignments between them. Indeed, the evolution of ontologies following changes in knowledge domains may affect and make obsolete the alignment between them. Thus, alignments must be evolved and maintained in order to keep up with the change in ontologies.

Recently, many methods have appeared to solve the problem of alignment evolution under ontology change (Zahaf, 2017). The main challenge for them is to maintain the alignment consistency after applying the change. An alignment is consistent if and only if the ontologies remain consistent even when used in conjunction with the alignment. The notion of consistency is approached by alignment evolution methods according to two different levels: structural consistency and logical consistency. The structural consistency ensures that the alignment obeys the constraints of its underlying representation structure (Martins & Silva, 2009). The logical consistency considers the

DOI: 10.4018/JITR.299923

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

semantics of the alignment, meaning that it does not introduce contradicting knowledge in ontologies (Euzenat, 2015; Zahaf & Malki, 2016).

The objective of this work is to take a step forward by considering the alignment evolution according to the conservativity principle under ontological changes. Jiménez-Ruiz et al. (2011) proposed the conservativity as a principle to minimize the number of potentially unintended logical consequences in the alignment. They state that mappings must not introduce new semantic relationships between concepts of one of the input ontologies. In other words, alignment should allow interaction between ontologies rather than providing a new description of the domain. In the context of alignment evolution under ontology change, an alignment is conservative if the ontological change should not introduce new semantic relationships between concepts from one of the input ontologies. We consider these relationships as violations of the conservativity principle following ontological changes.

To our knowledge, the problem of *Alignment Conservativity under Ontology Change* has not been studied yet. We built on top of previous work on repairing automatically evolved alignments under ontology change (Zahaf & Malki, 2016) and extend this work in several directions:

1. Zahaf & Malki (2016) don't look for eventual violations of these axioms. Instead, they rely on the state-of-the-art of reasoners to detect alignment inconsistency regardless ontological change. In this work, we propose a method for detecting conservativity violations following the change in input ontologies. The method is based on two patterns depending on the type of the ontological change. The former detects violations following the addition of new axioms to new ontology version, while the latter detects violations following the removal of axioms from this version. While existing methods of repairing conservativity violations (Solimando et al., 2016) only detect violations of the subsumption and equivalence types, our method detects violations of all types of axioms. We differentiate two possible situations in which the alignment can fall into conservativity violation depending on whether the violation appeared before or after the ontological change. In the context of the evolution of alignment under the ontology change, we are concerned with the second situation, that is, the violation of conservativeness caused by the ontological change.
2. The alignment repair method proposed in (Zahaf & Malki, 2016) is designed to deal with any arbitrary unwanted axiom to restore the alignment consistency. We adapt this method to repair the conservativity violations in the context of alignments evolution under ontology change.
3. We conduct an experiment on a dataset adapted from the Ontology Alignment Evaluation Initiative (OAEI)³. The experiment demonstrates both the practical applicability of the proposed approach and shows the limits of ontology matching tools when dealing with alignment evolution problem w.r.t conservativity principle. We notice that the outputs of these tools suffer from this problem and propose our method as a complementary solution to cope. In fact, we do not consider our approach as a turnkey method to evolve an alignment, but rather as an additive component for this kind of approach, dealing with the conservativity violations upon ontological change.

In the remainder of this paper, section 2 summarizes the relevant related work. Section 3 presents the basic concepts and definitions we will rely on along the paper. In Section 4, we formally state the problem of alignment conservativity under ontology change. Section 5 presents the proposed approaches concerning the detection and repair process. We evaluate and analyze our findings in Section 6. Finally, Section 7 gives some conclusions and future work lines.

RELATED WORKS

Recently, many approaches have appeared to solve the problem of alignment evolution under the ontology change. We can identify two types of classes. The first calculates a new alignment from scratch by using ontology matching tools, while the second reuses as much as possible the old alignment

by adapting it to the ontological change. The main challenge for both classes is to maintain the consistency of alignment after applying the change (Euzenat, 2015). An alignment is consistent iff the ontologies remain consistent even when used in conjunction with the alignment. Haase & Stojanovic (2005) distinguish three types of consistency: structural, logical and user defined consistency. The structural consistency is determined by a set of conditions w.r.t the underlying models of ontologies, while logical consistency ensures no contradiction can be entailed from those ontologies. The user-defined consistency refers to user requirements that need to be expressed outside of the ontology language itself. Other methods like (Jiménez-Ruiz et al., 2011), and (Solimando et al., 2016) have taken a step forward, and treat the conservativity of alignment. The conservativity is a general form of logical consistency by preventing any unwanted axioms from being a logical consequence of the alignment. To our knowledge, no method has previously addressed the conservativity problem under ontological change. In this context, an alignment is conservative if the ontological change should not introduce new semantic relationships between concepts from one of the input ontologies. In the following, we discuss the approaches of the two classes according to the type of consistency they ensure during the evolution of the alignment.

Ontology Matching Methods

We consider ontology matching methods as a solution to deal with alignment evolution under ontology change by calculating a new alignment from scratch. These tools differ in the nature of the knowledge encoded in the ontology, and techniques used in the identification of correspondences (Euzenat et al., 2011). Almost all existing matching systems combine different techniques (terminological, structural, instances, etc.) to fulfill lacks of every category type and maintain the consistency of alignment after applying the change (Euzenat, 2015). An alignment is consistent iff the ontologies remain consistent even when used in conjunction with the alignment.

Structural consistency is targeted by a first set of tools like ALIN (Da Silva et al., 2020) through an interactive phase based on expert feedback to produce the so called mappings suggestions. After each expert feedback, ALIN modifies the mapping suggestions set using structural analysis of ontologies and alignment anti-patterns. SANOM (Mohammadi, 2019) combines lexical (Jaro-Winkler and WordNet) and structural metrics to map entities of two ontologies. It seems effective in dealing with structural consistency, but has no guarantees towards logical consistency.

Another set of approaches deal with logical consistency. For example, last version of Lily (Tang et al., 2018) combines alignment and ontologies in single (*is-a*) graph and uses three matching calculation strategies: Generic/ Large-scale/ Instance ontology matching, and two repair strategies: Ontology mapping debugging/ tuning. The choice to delete or change suspicious mappings is left to the user. YAM ++ (Bellahsene et al., 2017) is based on ALCOMO (Meilicke & Stuckenschmidt, 2007) which proposes four patterns to detect logical consistency in subsumption/equivalence correspondences between classes/ properties. Two type of diagnosis are projected: Global optimal diagnosis removes the slightest amount of confidence, and Local optimal diagnosis, by an incremental check of the correspondences set.

Techniques presented so far deal well with logical consistency problem according to the contradictory axioms notion, which causes unsatisfiability within a set of alignment correspondences. However, this performance does not ensure the alignment conservativity following the ontological changes. In this context, an alignment is conservative iff the ontological change should not introduce new semantic relationships between concepts in the input ontologies. These relationships are considered as conservativity violations following ontological changes. As a solution, a third set of approaches have emerged under the name of LogMap-Family (Jiménez-Ruiz, 2019). They aim to calculate a new alignment from scratch while respecting conservativity principle.

Jiménez-Ruiz et al. (2011) use a specific pattern to detect conservativity violations in UMLS-Meta4 thesaurus. If an input ontology does not imply a relation between two of its entities, and this relation arose via alignment, then the related correspondences are in conflict and one of them may be incorrect. Conflicting matches are detected using an ontology reasoner and diagnosed by removing

the slightest confident correspondence in each conflict. This technique is similar to our elimination choice, but different in the size of conflict sets since it is applied only to pairs of mappings, while our diagnosis deals with larger conflict sets. Note also that it takes only the ontology source and the alignment. Yet, this can be a subject of many neglected logical consequences when discarding the target ontology. To argue this, we have provided a counterexample in a previous work (Atig et al., 2016). Solimando et al. (2016) suggest that an aligned ontology must not introduce new subsumption relationships between concepts within the input ontologies. They reduce the conservativity violations detection to an alignment inconsistency detection using the disjointness assumption (Schlobach, 2005). Mappings identified as violations are repaired in the same way as the previous approach using confidence value as a differentiating factor. The same work offers another technique to detect the called equivalence violation. It is about a sets of correspondences that form a cycle. The reparation uses the weighting in the arcs of problematic cycles to eliminate the arc with the lowest weight so that respect the minimality of change principle. Here also, this technique considers only subsumption axioms as conservativity violations since equivalence violation is treated as a two-way subsumption, while our approach detects all types of violations. It is clear that this kind of techniques wastes all efforts provided before the ontological change. What cause the emergence of methods aiming to adapt the alignment following ontological changes instead of calculating a new one from scratch.

Alignment Adaptation Methods

Starting from the idea that alignment calculation is not a trivial task, and since change in ontologies may trigger change in alignment, a set of approaches has emerged to deal with the alignment adaptation problem. In this context, alignment evolution methods aim to reuse as much as possible the old alignment by adapting it to the ontological change. Similar to ontology matching context, adaptive methods differ according to the performance targeted in the outputting alignments and are differentiated according to the same three levels: structural consistency, logical consistency and conservativity methods.

A first set of approaches aims to guarantee structural consistency. For example, Khattak et al. (2015) use a change history log (Khattak et al., 2008) to delete the affected correspondences from alignment. Then, the changed elements in the evolved ontology are automatically matched with the complete current new ontology version to replace the deleted correspondences while the rest is reused in the new alignment. Compared to fully re-computing from scratch techniques, this approach reduces significantly the time required to maintain alignments. By cons, seeking new match for changed entities can only ensure a structural consistency.

Groß et al. (2013) propose two approaches: the composition-based and diff-based adaptation approaches. Using the ontology matching tool GOMMA (Kirsten et al., 2011) and ontology versions, the first approach converts the implicit change to an alignment. The second approach uses COnTo-Diff tool (Hartung et al., 2013) to identify changes between evolved ontology versions, then converts the changes to a set of semantic relations between concerned entities. Both approaches compute new correspondences between added concepts and concepts of the target ontology to improve the alignment. We have implemented the composition of an alignment between ontology versions with an initial alignment in a previous work (Atig et al., 2013), and we can testify on the simplicity of its implementation, but nothing ensures that resulting alignment is valid. Moreover, the correctness of this operation depends on the correctness of the composed alignments. Both proposals rely on heuristic rules to generate an alignment between versions. Thereby, no guarantees are given to ensure the logical consistency of alignment. To overcome this disability, a second set of approaches have emerged to ensure a logically consistent evolution of the alignment following the changes made in the input ontologies.

Euzenat (2015) study the problem of restoring consistency in a network formed by a set of ontologies connected by a set of alignments. He defines two types of inconsistencies: local inconsistency which is an ontology or alignment inconsistency and global inconsistency which arises in the network but ontologies and alignments are consistent in isolation. The local revision of ontology or alignment is the proposed solution for local inconsistencies and these local operations can be used independently to resolve the

global inconsistency. This framework is considered as impracticable and cannot be incorporated easily into a computational framework since closed sets are either infinite or very large and (Peppas, 2008).

Zahaf & Malki (2016) aim to preserve ontological change meaning and ensuring consistent alignment evolution under ontology change and offer two operations: Alignment revision which restores the consistency following adding new axioms in input ontologies, and Alignment contraction which ensures not entailing again the removed axioms. The authors draw a set of constraints that an alignment should satisfy in order to be correctly evolved. Then, based on diagnosis theory (Reiter, 1987), they propose an automatic method to reach this objective. A conflict set of correspondences responsible for violations is calculated and diagnosed from the alignment. Despite the difference in the violations nature between consistency and conservativity principles, this approach is similar to ours in terms of violation detection process for contracted axioms but completely different in the case of added axioms (revision).

To our knowledge, the problem of alignment conservativity under ontology change has not been studied yet. We believe that the current work is the first to address it. Thus, while waiting for other approaches to emerge in the same context, we consider this proposal as primary step to perfect the task of alignment evolution following the change in the input ontologies. To conclude, we note that approximately, all studied alignment repair systems adopt a common principle which suggests that the input ontologies are immune during the repair phase, except restoring ontology local inconsistency in (Euzenat, 2015) and the work in (Pesquita et al, 2013) which proposes updating ontologies during the automatic calculation of repairs. Furthermore, most of alignment revision techniques use the conflict set and diagnosis notions inspired by diagnosis theory. We follow the same strategies in this work whether through immunization input ontologies in the repair process, or concerning conflict set and diagnosis.

PRELIMINARIES AND NOTATIONS

This section contains the basic notions that will be used throughout this paper. The first and most important one is the OWL2⁵ ontology viewed as logical theory.

Definition 1 (OWL2 Ontologies). An OWL2 ontology O is a couple (S, A) , with A is a set of axioms that constraint the intended interpretation of a vocabulary S also called signature $Sig(O)$ in a domain of discourse. The vocabulary provides legal names for the entities appearing in the ontology, while axioms act as semantic constraints to define these entities.

An interpretation which satisfies all axioms of an ontology constitutes a model of that ontology. The model notion establishes a logical consequence relation between an ontology and axioms expressed in the language of this ontology.

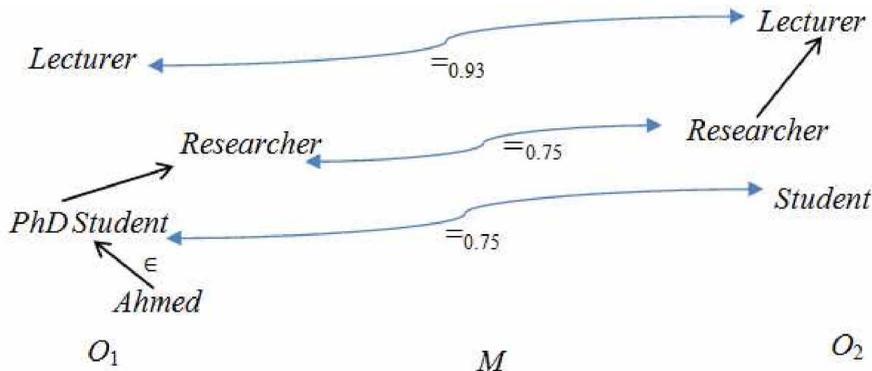
Definition 2 (Ontology Consequence). An axiom δ is a logical consequence of an ontology O (noted $O \models \delta$) iff every model of O satisfies δ .

Ontology alignment is the task to detect links between elements from two ontologies. These links are referred as correspondences and express semantic relations. While a matchable element can be any arbitrary entity, we consider here only alignments of matchable entities that belong to ontologies. We adapt the definition of Euzenat & Shvaiko (2007) as follows .

Definition 3 (Ontology alignment). Given two ontologies O_i and O_j , let $Q(O_i)$ (respectively $Q(O_j)$) be the set of matchable entities of O_i (respectively O_j). A correspondence between O_i and O_j is a 4-tuple (e_i, e_j, r, n) such that, $e_i \in Q(O_i)$, $e_j \in Q(O_j)$, r is a semantic relation, and $n \in [0; 1]$ is a confidence value. An alignment M between O_i and O_j is a set of correspondences between O_i and O_j .

Example 1: Considering the alignment M of Figure 1. We use DL like syntax to describe both ontologies. Also, we use the index number in ontologies notation as name space to designate entities. Alignment M is created by the ontology matching system YAM++6

Figure 1. An alignment M between two educational domain ontologies O_1 and O_2



$$O_1 = \left\{ \begin{array}{l} PhDStudent \sqsubseteq Researcher \\ PhDStudent(Ahmed) \end{array} \right\}; O_2 = \left\{ Researcher \sqsubseteq Lecturer \right\}$$

$$M = \left\{ \begin{array}{ll} 1 : PhDStudent \stackrel{0.93}{=} 2 : Student \\ 1 : Researcher \stackrel{0.75}{=} 2 : Researcher \\ 1 : Lecturer \stackrel{0.75}{=} 2 : Lecturer \end{array} \right\}$$

There is no standard for alignment semantics. In (Borgida & Serafini, 2003), distributed description logics semantics have been proposed. Another approach called reductionist semantics interprets correspondences of the alignment as axioms in some merged ontology (Meilicke & Stuckenschmidt, 2009). The merged ontology is called aligned ontology. In this work, we use an example of this semantic called natural semantic. It involves building a merged ontology through the union of the two ontologies to align and axioms obtained by translating relations of the alignment. We introduce this semantic through its aligned ontology.

Definition 4 (Natural Semantics). Given an alignment M between two ontologies O_1 and O_2 and $trans: M \rightarrow A$, a function that transforms a correspondence to an axiom. The natural semantics of M is defined by the following aligned ontology:

$$O_1 \cup_M O_2 = O_1 \cup O_2 \cup trans(M).$$

We introduce the notion of alignment consequence according to natural semantics as follows:

Definition 5 (Alignment consequence). An axiom δ is an alignment consequence of an alignment M between two ontologies O_1 and O_2 if δ is a logical consequence of the aligned ontology $O_1 \cup_M O_2$.

An axiom which is an alignment consequence either represents an ontological axiom or the image of a correspondence by the transformation function of the alignment.

Definition 6 (ontology signature isomorphism). Given two ontologies $O_1=(S_1, A_1)$ and $O_2=(S_2, A_2)$, an ontology signature isomorphism is a particular alignment $M : S_1 \rightarrow S_2$ such that $A_2 \models M(A_1)$ and $A_1 \models M^-(A_2)$, i.e., all models of A_2 are models of the image of A_1 by M and vice versa. The image of an axiom is obtained by systematically replacing signature elements of this axiom by their correspondents, according to the signature isomorphism M .

PROBLEM STATEMENT

Zahaf & Malki (2016) are based on the belief base revision theory to introduce two postulates, namely: ontology change preservation and logical consistency for alignments repair following the ontology change. The change preservation ensures that deleted axioms should no longer be logical consequences of the alignment. While logical consistency guarantees that ontological change does not generate contradictory knowledge in ontologies. Note that in monotonic logics, an inconsistency can only occur if certain types of axioms have been added. We reconsider these two postulates in the context of the conservativity principle under ontological change. We reformulate the former and generalize the latter to integrate any type of added axioms, and we define the general concept of conservation in the context of alignment evolution under ontology change. In this context, the alignment is conservative if the ontological change does not have to introduce new semantic relationships between the concepts of an introductory ontology.

Jiménez-Ruiz et al. (2011) identified the conservativity principle as a conservativity extension (Lutz et al., 2007) problem by calculating the deductive difference between one ontology and its extension consisting in adding alignment to this ontology (i.e., $diff(O_i, O_i \cup_M M)$). Solimando et al. (2016) generalize this proposition and state that deductive difference $diff(O_i, O_i \cup_M O_j)$ between any ontology O_i , such that $i \in \{1, 2\}$, and the aligned ontology must be empty w.r.t the signature of that ontology. The deductive difference between O_i and $O_i \cup_M O_j$ is the set of entailments $\{\delta\}$ formulated over $Sig(O_i \cup O_j)$ that do not hold in O_i , but do hold in $O_i \cup_M O_j$. Formally:

$$diff(O_i, O_i \cup_M O_j) = \{\delta \mid O_i \not\models \delta \text{ and } O_i \cup_M O_j \models \delta \text{ and } Sig(\delta) \subseteq Sig(O_i \cup O_j)\}$$

We differentiate two possible situations in which the alignment can fall into conservativity violation depending on whether the violation appeared before or after the ontological change. In the context of the evolution of alignment under the ontology change, we are concerned with the second situation, that is, the violation of conservativeness caused by the ontological change. Thus, we define the alignment conservativity violations under ontology change as the set theoretical difference between the alignment conservativity violations before and after the change. Formally,

Definition 7 (Alignment Conservativity Under Ontology Change). Let O_{i1} and O_{i2} two versions of the evolved ontology O_i . M an alignment between two ontologies O_i and O_j is conservative under ontology change iff There are no violations after the change, except for those before the change:

$$diff(O_j, O_{i2} \cup_M O_j) = diff(O_j, O_{i1} \cup_M O_j)$$

$$diff(O_{i2}, O_{i2} \cup_M O_j) = diff(O_{i1}, O_{i2} \cup_M O_j)$$

Example 2 shows two conservativity violation situations: Figure 2 illustrates the problem before the change and Figure 3 illustrates it after the change.

Example 2: Following example 1, we note that, before the evolution of the input ontologies as illustrated in Figure 2, the set of conservativity violations is the deductive difference $diff(O_2, O_1 \dot{\cup}_M O_2) = \{2:Student \dot{\cup} 2:Researcher\}$. Therefore, the axiom $2:Student \dot{\cup} 2:Researcher$ (dashed green arrow) represents a violation of the conservativity before the change.

Assuming now that one of the two input ontologies has been evolved and let $O_1' = O_1 \dot{\cup} \{1:PhDStudent \dot{\cup} 1:Lecturer\}$ be the new version of O_1 following the addition of the new axiom $1:PhDStudent \dot{\cup} 1:Lecturer$ (solid red arrow). This change can be requested for example by applications using ontology O_1 , since the added axiom is entailed when using O_1 in conjunction

Figure 2. Conservativity violation before ontology change

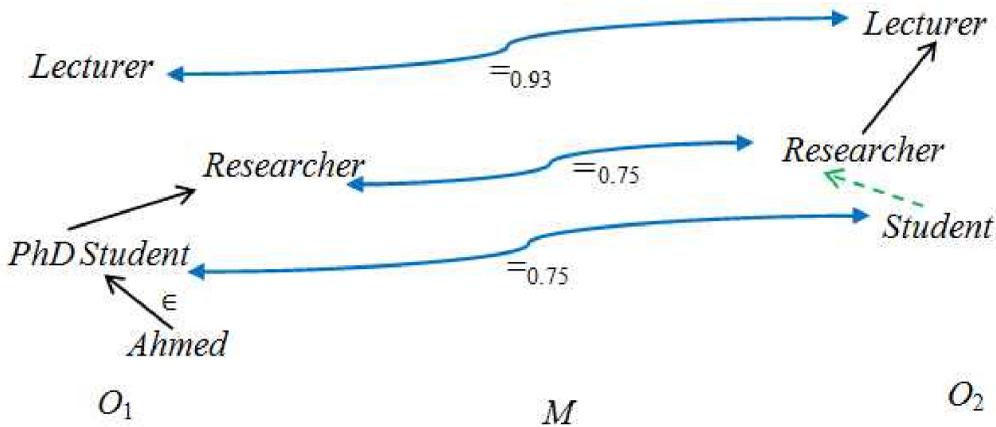
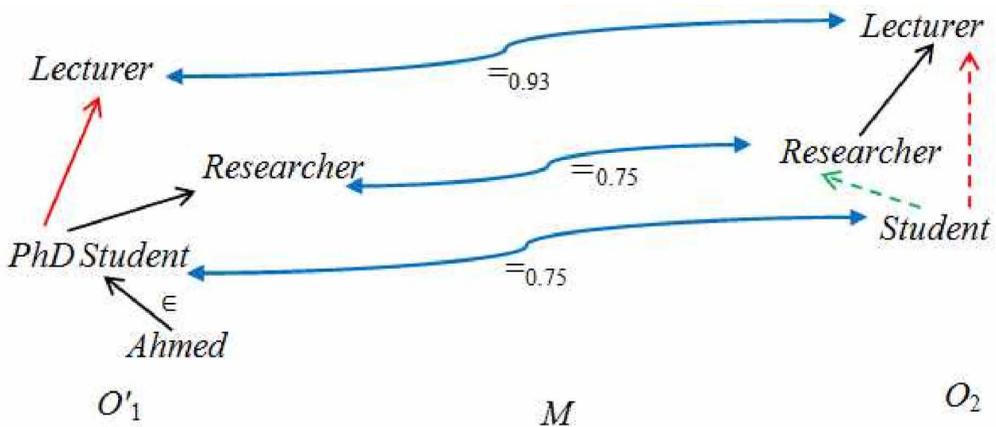


Figure 3. Conservativity violation after evolving the ontology O_1 to O'_1



with O_2 and alignment M , which leads ontology O_1 owners to explicitly evolve it by adding a new axiom $\{1:\text{PhD Student} \sqsubseteq 1:\text{Lecturer}\}$. In this situation, $\text{diff}(O_2, O'_1 \dot{\bar{E}}_M O_2) = \{\{2:\text{Student} \sqsubseteq 2:\text{Lecturer}\}, \{2:\text{Student} \sqsubseteq 2:\text{Researcher}\}\} \neq \text{diff}(O_2, O_1 \dot{\bar{E}}_M O_2)$. So, according to definition 7, alignment M violates the conservativity under evolving O_1 .

The problem of conservativity violations involves two major challenges, namely: *Detecting conservativity violations* and *Repairing alignment following conservativity violations*.

DETECTION OF CONSERVATIVITY VIOLATIONS UNDER ONTOLOGY CHANGE

The conservativity principle as a deductive difference already suffers from two major drawbacks (Lutz et al., 2007) and (Lutz & Wolter, 2010), namely: lack of algorithm available for computing deductive difference for DL logics, and the massive, up to infinite, number of entailments in this difference. In order to avoid these drawbacks, we suggest an approximation of the deductive difference in the context of alignment evolution under ontology change.

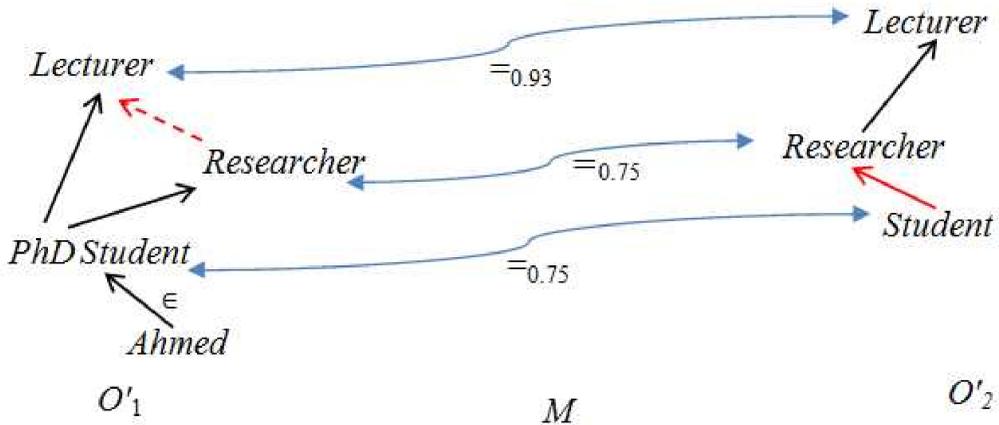
We only consider here the alignments with the equivalence relations. This is not a disadvantage of our approach because it is always possible to find a subset of the alignment with only equivalent relations. An equivalence relation expresses that linked entities represent the same thing in the domain of discourse. In this case, the alignment constitutes an isomorphism of ontological signature (Kalfoglou, & Schorlemmer, 2003) connecting the vocabulary of two ontologies so that the axioms specifying the linked entities are preserved or conserved. This conservativeness must remain valid throughout the ontologies life cycle. Otherwise, we must register conservativity violations.

Definition 8 (Conservativity Violations Under Ontology Change Detection Patterns). Let O_i be an ontology which have evolved to a new version O_{i2} and M an alignment between two ontologies O_i and O_j might manifests conservativity violations under ontology change if:

- For all added axiom δ^+ such that $sig(\delta^+) \dot{\in} Q(O_{i2})$, we have $O_{i2} \dot{\in}_M O_j M(\delta^+)$ but $O_j \not\in M(\delta^+)$
- For all deleted axiom δ^- such that $sig(\delta^-) \dot{\in} Q(O_{i2})$, we have $O_{i2} \dot{\in}_M O_j M(\delta^-)$

Note that conservativity violations only concern axioms whose signature is fully involved in the alignment, which means that the signature elements of any axiom are matchable entities. Example 3 illustrates a situation in which the images of the added axioms in O_{i2} must exist as a logical consequence of the ontology O_j .

Figure 4. Evolution of ontology O_1 into O_2



Example 3: Following example 2, let $O_2' = O_2 \dot{\in} \{2:Student \dot{\in} 2:Researcher\}$ be a new version of O_2 by applying the new change 2: Student $\dot{\in}$ 2: Researcher (solid red arrow in Figure 4). The current change restores the conservativity of a subset of M (i.e., $\{1:PhDStudent \xrightarrow{0.75} 2:Student; 1:Researcher \xrightarrow{0.75} 2:Researcher\}$), since that, $O_1 \dot{\in} M(2:Student \dot{\in} 2:Researcher) = \{1:PhDStudent \dot{\in} 1:Researcher\}$. However, M is not fully conservative, since that, $O_1 \not\in M(2:Researcher \dot{\in} 2:Lecturer) = \{1:Researcher \dot{\in} 1:Lecturer\}$ represented by a dashed red arrow in Figure 4.

Example 4 considers another situation where the images of the deleted axioms from O_{i1} must not exist as a logical consequence of the ontology O_j .

Example 4: Following example 3, assuming now that ontology O_1 has been evolved once again to $O_1'' = O_1 / \{1:PhDStudent \dot{\in} 1:Researcher\}$ by contracting the axiom 1:PhDStudent $\dot{\in}$ 1:Researcher from O_1 . The current change breaks the conservativity of the subset $\{1:PhDStudent \xrightarrow{0.75} 2:Student; 1:Researcher \xrightarrow{0.75} 2:Researcher\}$ of M , since that, $O_2 \dot{\in} M(1:PhDStudent \dot{\in} 1:Researcher) = \{2:Student \dot{\in} 2:Researcher\}$.

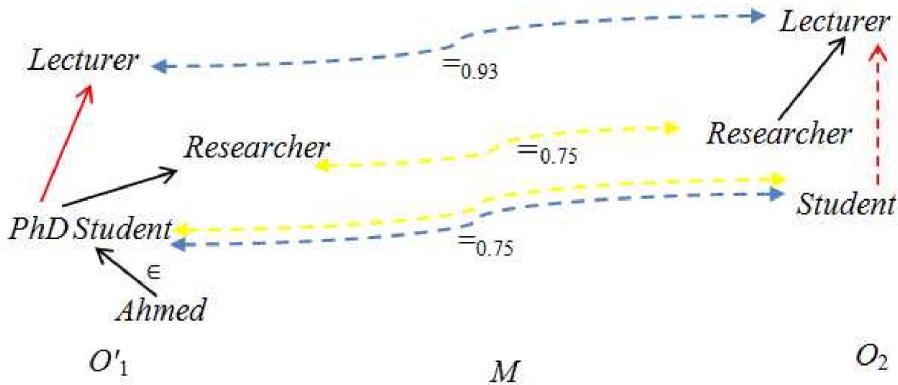
REPARATION OF CONSERVATIVITY VIOLATIONS UNDER ONTOLOGY CHANGE

Unlike existing methods (Jiménez-Ruiz et al., 2011), and (Solimando et al., 2016) which are dedicated to repairing certain types of conservativity violations such as subsumption and equivalence violations, Zahaf & Malki (2016) design an alignment repair method that treats any arbitrary unwanted axiom. We reuse this method to repair conservativity violations in the context of alignment evolution under ontology change. According to Zahaf & Malki (2016), an alignment repair is a diagnosis task that aims to compute and eliminate from the alignment a subset of correspondences called a diagnosis to fix the conservativity violation. First, the method determines the set of correspondences called the conflict set that causes the conservativity violation. Formally,

Definition 9 (Conflict Set). Let δ is a conservativity violation caused by an alignment M between two input ontologies, namely O_i and O_j . A subset C of M is a conflict set if $O_i \cup_c O_j = \delta$.

C is *minimal* if for all subset C' of C , we have $O_i \cup_c O_j \neq \delta$

Figure 5. Two conflict sets for a single conservativity violation



Example 5: Following example 2, The conflict sets responsible of the conservativity violation of alignment M upon $O'_1 = O_1 \cup \{1 : PhDStudent \subseteq 1 : Lecturer\}$, are:

$C_1 = \{1 : PhDStudent =_{0.75} 2 : Student ; 1 : Researcher =_{0.75} 2 : Researcher\}$ (dashed yellow arrows in Figure 5).

$C_2 = \{1 : PhDStudent =_{0.75} 2 : Student ; 1 : Lecturer =_{0.93} 2 : Lecturer\}$ (dashed blue arrows in Figure 5).

Since there can be several conflict sets for the same violation, the method constructs the diagnosis by selecting the correspondences with the lowest confidence value in each conflict set.

Definition 10 (Alignment Diagnosis). Given, MC a set of minimal conflict sets of an alignment M w.r.t a conservativity violation δ , Δ is a diagnosis for an alignment M w.r.t MC if:

$$\left\{ \begin{array}{l} (i) \Delta \subseteq \cup MC \\ (ii) \text{if } \emptyset \neq X \in MC, \text{ then } X \cap \Delta \neq \emptyset \\ (iii) \text{if } c = (e, e', r, n) \in \Delta, \text{ then there exists } CMC \text{ such that,} \\ \quad cC \text{ and } n = \min \{n_i | (e_i, e'_i, r_i, n_i) \in C\} \end{array} \right.$$

Example 6. Following example 5, the diagnosis of the alignment M is:

$$\Delta = \{1 : PhDStudent =_{0.75} 2 : Student\}$$

Finally, the alignment reparation process discards the diagnosis from the original alignment in order to restore its lost conservativity upon input ontologies evolution. The result of this reparation is a repaired sub-alignment w.r.t the conservativity principle.

Definition 11 (Alignment Repair). Let δ is a conservativity violation caused by an alignment M between two input ontologies, namely O_i and O_j and Δ is a diagnosis for δ . The alignment repair is the contraction operation of Δ from the alignment M , noted $M \setminus \Delta$.

Example 7. Following example 6, the repaired alignment by the obtained diagnosis Δ is:

$$M \setminus \Delta = \{1 : Lecturer =_{0.93} 2 : Lecturer; 1 : Researcher =_{0.75} 2 : Researcher\}$$

Table 1 summarizes the method. At worst, it repairs the alignment in logarithmic time.

Table 1. Alignment repair method

Method: alignment repair
<p>AlignmentRepair($M, o1, o2, \delta$) Input: $o1, o2$// two ontologies M// M is an alignment between $o1$ and $o2$ δ// δ is a conservativity violation Output: M// repaired alignment</p> <ol style="list-style-type: none"> 1. $\Delta \leftarrow \emptyset$ 2. while $O_i \cup_M O_j = \delta$ 3. do 4. $CS \leftarrow \text{MinConflictSet}(M, o1, o2, \delta)$ 5. $Clv \leftarrow \text{CorrespWithLowestConfidValue}(CS)$ 6. $\Delta \leftarrow \Delta \dot{\cup} \{Clv\}$ 7. $M \leftarrow M, \{Clv\}$ 8. Return M

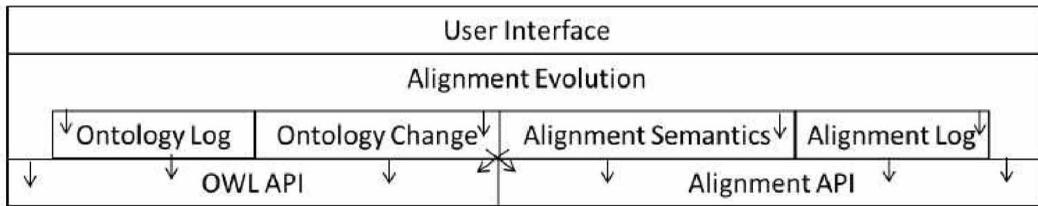
EXPERIMENTAL EVALUATION

Zahaf & Malki (2016) distinguish two classes of alignment evolution methods. While methods of the former reuse as much as possible the old alignment, methods of the latter compute from scratch the new alignment. Also, none of the approaches of both classes guarantee the preservation of the ontology change in alignment evolution task. The preservation of the ontology change is a special case of the conservativity principle which only concerns deleted axioms. Through this experience, we will test some methods to consolidate this argument by extending it to all cases of the conservativity principle. Mainly, the selected methods rely on ontology matching techniques for evolving alignments. Besides, they embed debugging techniques to diagnosis alignments for eventual consistency problems. By selecting these methods we want to show neither ontology matching nor alignment debugging methods fit well for the problem of conservativity violations in the context of ontology alignment evolution under ontology change.

Implementation

We built on top and extended the automatically evolved alignment repair under ontology change platform (Zahaf, 2017) to address ontology alignment conservativity violations under ontological change. The platform embeds the OWL-API (Horridge & Bechhofer, 2009) and Alignment-API (Euzenat, 2004) libraries as a baseline for managing OWL ontologies and alignments between ontologies. Figure 6 illustrates the platform architecture. Ontology change component is responsible

Figure 6. Architecture of the alignment evolution system (Zahaf, 2017).



for identifying and representing the ontology change. Alignment log component embeds services for representing, storing, and tracking the alignment change. Alignment evolution component implements the alignment evolution under ontology change repair. Alignment semantics component relies on the state-of-the-art of reasoners to check alignment consistency and entailments. We have extended the alignment evolution component to implement the method for detecting conservativity violations following the ontological change. By considering the alignment as an isomorphism, the method checks the entailment of the image in one ontology of the axiomatic change in the other ontology.

Dataset

Ontologies and Change

OAEI⁷, a coordinated international initiative carries out annual campaigns for the evaluation of ontology matching tools. It uses a benchmark dataset for identifying strengths and weaknesses of matching systems. The benchmark dataset consists of a large set of artificial tests. These tests alter an initial ontology about the topic of scientific publications and the task is to match it to the modified ontology. Modifications consist of inserting or deleting some features, e.g., replacing by random labels, deleting or inserting classes in the hierarchy, ect. The ontologies are described in OWL-DL and serialized in RDF/XML format. The initial ontology is that of test #101. It contains 33 named classes, 24 object properties, 40 data properties, 56 named individuals and 20 anonymous individuals.

Zahaf & Malki (2016) adapted a subset of the systematic benchmark for evaluating alignments evolution methods under ontology change. In what concerns ontological changes, they rearrange tests #101, #103, #104, #203, #223, #230, and #233 to form the new tests # 101-103-104, #101-203-223, and #101-230-233 according to the assessment requirements. They also consider ontologies 104, 223 and 233 as a version of 103, 203 and 230, respectively.

To generate the ontological change, we have used the method developed in (Zahaf, 2012) to compute the difference between versions. This method, considers the ontological change operation as the set theoretical difference between signatures and axioms, respectively. Since the conservativity principle is a logical property which might concern only axioms whose signature is fully implied in alignments, we only consider the axiomatic change of matchable signatures. Table 2 summarizes the obtained change.

Table 2. Ontological change between versions of the dataset

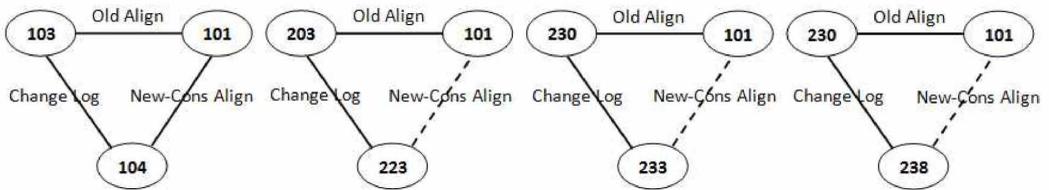
Difference Versions	Added Axioms	Deleted Axioms
103-104	0	11
203-223	1	9
230-233	0	220
230-238	182	71

The axioms removed from 103 compared to 104 are domains for object and data-properties. Besides adding new entities and related axioms to version 223, definitions of other entities have changed by adding axioms. The same holds for definitions of some entities in version 203 by removing axioms. Removed axioms are domains, ranges and some restrictions on properties. Since both do not have hierarchies, no axioms added between 230 and 233. Deleted axioms are due to the removal of object and data-properties. As the ontological change generated is mainly of suppression type, we extend the set with the additional test #101-230-238 to enrich it with addition type. Comparison between the versions 230 and 238 shows the removal of instance, and related axioms and adding other entities and axioms.

Alignments

Concerning alignments to be repaired, we consider as old alignments, those between the following ontologies pairs: 101-103, 101-203 and 101-230, while the alignments between the following ontologies pairs: 101-104, 101-223, 101-233 and 101-238 are the evolved alignments after change. Figure 7 schematizes this dataset.

Figure 7. Dataset



Accuracy Measures

The considered dataset does not contain reference alignments to measure accuracy w.r.t conservativity principle, which restricts the use of traditional precision methods. Therefore, to compare the performances of evolution methods in ontology matching context, we use the number of conservativity violations by changed axioms. In addition, we compare the elapsed time, as well as the rate of violations reparation for all methods. The violations reparation rate of an alignment M is defined by $\%Rep=(\Delta/M)*100\%$. where Δ is a diagnosis of initial alignment M .

Ontology Matching Tools

In the ontology matching context, this experimentation requires alignments between new ontology versions and ontology 101. In order to calculate these alignments, we consider the matching tools referenced in the OAIE's annual workshop. The workshop knows the participation of many competitive ontology matching tools. Without exception, all of them perform well in the track of systematic benchmark test and register high precision that is close to 1.00. Some of them are open software and they are available to download from the web. Even others are not open software; their outputs for the systematic benchmark test are available on their web sites. We have selected YAM++, Lily⁸ and ASMOV⁹ since these systems embed semantic check components for bugs diagnosis. Regarding Lily, we use its version2 available for downloading on its website. Lily presents a user friendly interface to configure some parameters. We choose 15 as the size of semantic sub-graph and we enabled similarity propagation option. Since we deal with semantics properties of alignments in this step, these parameters setting are more than necessary to fit the systems with their full potentialities. We use both YAM and

Lily to generate alignments between 101-104, 101-223, 101-233 and 101-238. ASMOV presents outputs alignments between these ontologies on its website and are available for downloading.

Experimentation

The experimentation process was conducted in two steps. In the first step, we exploit the change logs between the original ontologies (103, 203 and 230) and their respective new versions (104, 223, 233 and 238) to detect the set of conservativity violations for the original alignments upon input ontologies evolution. In the second step, we use our method to show the performances and limits of the selected alignment evolution methods to avoid conservativity violations.

Step 1 (Violations Detection Process)

To detect conservativity violations upon ontology evolution, we use logs (103-104, 203-223, 230-233 and 230-238, respectively). These logs contain two types of information: added and removed axioms. We only consider axioms whose signatures represent matchable entities. Then, for each change, we apply the appropriate detection pattern. After obtaining alignments between new ontologies versions and the ontology 101, we count the number of conservativity violations caused by the related ontological changes. Table 3 presents the detailed values of results for each test and each tool in this experiment. The first column designates the selected method, while second shows every test named by its related ontologies. The third and fourth columns show respectively the number of correspondences and conservativity violations in the old alignment.

Step 2 (Methods Performances and Limitations)

This step aims to show the limits of the selected methods to avoid alignment conservativity violations upon ontology change. We compare the performance of YAM++, Lily and ASMOV in the alignment evolution context. The fifth column of Table 3 shows the number of correspondences in every diagnosis. The sixth column shows the size of new alignments generated by the selected methods/test.

When we applied the repair method (see Table 1) on initial alignments, we observed similarities in the results, and the number of conservativity violations seems to be the same for all methods for each test. However, this similarities does not confirm that all methods register the same score when dealing

Table 3. Ontological change between versions of the dataset

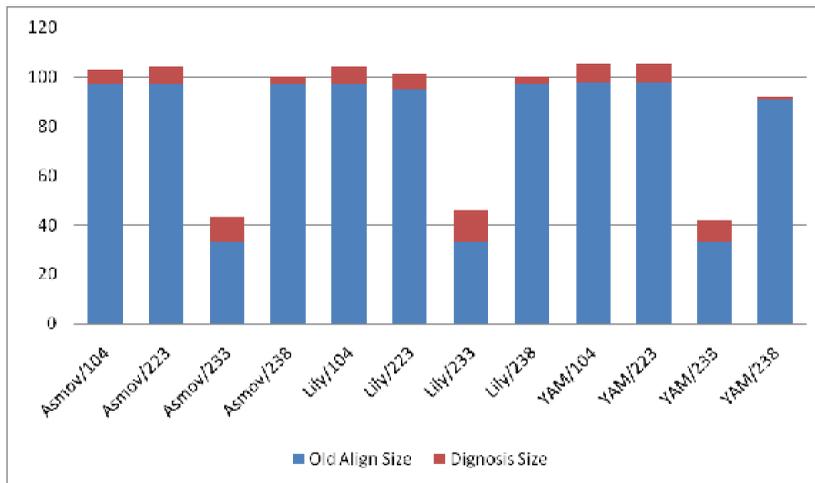
Method	Test	#OldAlgn	#Viol	#Diagnosis	#NewAlgn	#Time ns	%Rep
Asmov	101-103-104	97	5	6	91	0.6	6.18
	101-203-223	97	10	7	90	0.75	7.21
	101-230-233	33	23	10	23	0.4	30.3
	101-230-238	97	3	3	94	0.41	3.09
Lily	101-103-104	97	5	7	90	0.7	7.21
	101-203-223	95	9	6	89	0.6	6.31
	101-230-233	33	23	13	20	0.51	39.39
	101-230-238	97	4	3	94	0.32	3.09
YAM	101-103-104	98	5	7	91	1.5	7.14
	101-203-223	98	9	7	91	1.09	7.14
	101-230-233	33	23	9	24	0.34	27.27
	101-230-238	91	1	1	90	0.11	1.09

with this problem. As a matter of fact, alignment quality depends on its content and its size. For instance, an empty alignment avoids completely the conservativity violation but it doesn't present any interest.

The selected ontologies and reference alignments between them in each dataset, are mainly designed to compare precision and recall of tools in ontology matching problem. However, in alignment evolution context, we haven't these references alignment. Hence, it's not possible to use the same traditional accuracy measures. Instead, we use the violations repair rate with the related elapsed time. These measures show for each method, at what degree our proposed method reuses the original alignment while respecting the conservativity principle upon ontology change. The two last columns of Table 3 show the results of these measures. The seventh column shows the elapsed time measured in nanosecond to repair old alignments. The eighth column shows the repair rate compared to old alignments size. Figure 8 summaries this comparison. It shows the repair rate for every method/test. Note that the test is designated here by the evolved ontology name.

Even if the used approaches represent tools of the ontology alignment problem, in three quarters of the tests, the violations repair did not exceed 7.21%. This represents a reuse of 92.79% of the original alignments. The remaining quarter represents the test 101-230-233 with all tools. This is due to the nature of ontological changes applied in this test. According to Table 2, about 220 axioms were removed from ontology 230, which represents a large number of changes for a reduced amount of correspondences (33 for the three tools). It is obvious that in such cases, another experiment is required to fix a threshold which separates between the adaptation approach and calculating a new alignment from scratch. Despite this, we find that this situation drastically confirms that the selected tools suffer from the problem of conservativity principle violation upon ontology change, and require an additive component to deals with this problem.

Figure 8. Comparative results of methods in the contexts of alignment evolution and ontology matching problems



CONCLUSION AND FUTURE WORKS

According to a structural or logical level, alignment consistency was the main challenge of evolution methods following ontological changes. In this work, we tried to take a step forward by considering the alignment evolution under ontology change w.r.t conservativity principle. We were able to position ourselves as the first work to tackle such a problem since, to our knowledge, it has not been studied yet. We proposed two patterns to detect conservativity violations according to the different changes that could affect ontology axioms. The first pattern deals with the case of adding an axiom to a version

of an input ontology. Whereas, the second pattern deals with the case of removing an axiom from it. The results of the detection process is then used to adapt the initial alignment to these changes. In this context, we reuse a method proposed in (Zahaf & Malki, 2016) to repair the detected conservativity violations. This choice seems reasonable since this method treats any arbitrary unwanted axiom unlike other existing methods (Jiménez-Ruiz et al., 2011), and (Solimando et al., 2016) which are dedicated to repairing certain types of conservativity violations such as subsumption and equivalence violations.

The conducted experiment demonstrates the practical applicability of the proposed approach to ensure a conservative evolution of the alignment following the input ontologies evolution. Actually our method is not a turnkey, but can serves as an add-on component to alignment evolution methods. It is concerned in adaptation techniques which either add or remove correspondences or change the confidence values compared to those which change the semantic relationships in these correspondences. Furthermore, the results of this experiment shed light on many ways to improve our method. For instance, we must consider the minimal change principle to refine our repair process. Also, an examination must be carried out for studying the current problem in the context of adaptation approaches affecting the semantic relationships. The impact of this kind of mapping change can, for instance, sweep away a subset of conservativity violations in the evolved alignment. In all the cases, a main conclusion that can be drawn from these experiences is that the problem of alignment evolution has not received a lot of importance and many fundamental as well as methodological aspects of this problem must be carried out.

FUNDING AGENCY

The publisher has waived the Open Access Processing fee for this article.

REFERENCES

- Atig, Y., Amine, A., Zahaf, A., & Kumar, A. (2013). Alignment Evolution between Ontologies Using a Change Log. *International Journal Of Data Mining And Emerging Technologies*, 3(2), 81–87. doi:10.5958/j.2249-3220.3.2.011
- Atig, Y., Zahaf, A., & Bouchiha, D. (2016). Conservativity Principle Violations for Ontology Alignment: Survey and Trends. *International Journal of Information Technology and Computer Science*, 8(7), 61–71. doi:10.5815/ijitcs.2016.07.09
- Bellahsene, B., Emonet, V., Ngo, D., & Todorov, K. (2017) YAM++ Online: A Web Platform for Ontology and Thesaurus Matching and Mapping Validation. In *ESWC: European Semantic Web Conference* (pp.137-142), doi:10.1007/978-3-319-70407-4_26
- Borgida, A., & Serafini, L. (2003). Distributed description logics: Assimilating information from peer sources. *Journal on Data Semantics*, 1, 153–184.
- Da Silva, J., Revoredo, K., Baião, F., & Euzenat, J. (2020). Alin: Improving interactive ontology matching by interactively revising mapping suggestions. *The Knowledge Engineering Review*, 35, E1. doi:10.1017/S0269888919000249
- Euzenat, J. (2004). An API for ontology alignment. *Proc. 3rd conference on international semantic web conference (ISWC)*, 698–712.
- Euzenat, J. (2015). Revision in networks of ontologies. *Artificial Intelligence*, 228, 195–216. doi:10.1016/j.artint.2015.07.007
- Euzenat, J., Meilicke, C., Stuckenschmidt, H., Shvaiko, P., & Trojahn, C. (2011). Ontology Alignment Evaluation Initiative: Six years of experience. *Journal on Data Semantics*, XV, 158–192. doi:10.1007/978-3-642-226304_6
- Euzenat, J., & Shvaiko, P. (2007). *Ontology Matching*. Springer Verlag.
- Groß, A., Dos Reis, J. C., Hartung, M., Pruski, C., & Rahm, E. (2013). Semi-automatic adaptation of mappings between life science ontologies. In *International Conference on Data Integration in the Life Sciences*, (pp. 90-104). Springer Berlin Heidelberg. doi:10.1007/978-3-642-39437-9_8
- Haase, P., & Stojanovic, L. (2005). Consistent evolution of OWL ontologies. In *European Semantic Web Conference*, (pp. 182-197). Springer Berlin Heidelberg.
- Hartung, M., Groß, A., & Rahm, E. (2013). COnto-Diff: Generation of complex evolution mappings for life science ontologies. *Journal of Biomedical Informatics*, 46(1), 15–32. doi:10.1016/j.jbi.2012.04.009 PMID:22580476
- Horridge, M., & Bechhofer, S. (2009). The OWL API: A Java API for Working with OWL 2 Ontologies. *Proc. OWLED 2009 - OWL Experienced and Directions Workshop*.
- Jean-Mary, Y. R., Shironoshita, E. P., & Kabuka, M. R. (2009). Ontology Matching With Semantic Verification. *Journal of Web Semantics*, 7(3), 235–251. doi:10.1016/j.websem.2009.04.001 PMID:20186256
- Jiménez-Ruiz, E. (2019). LogMap Family Participation in the OAEI 2019. In *Proceedings of the 14th International Workshop on Ontology Matching*. CEUR-WS.org.
- Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., & Berlanga, R. (2011). Logic-based Assessment of the Compatibility of UMLS Ontology Sources. *J. Biomed. Semant.*, 2(Suppl 1), S2. doi:10.1186/2041-1480-2-S1-S2 PMID:21388571
- Kalfoglou, Y., & Schorlemmer, M. (2003). Ontology mapping: The state of the art. *The Knowledge Engineering Review*, 18(1), 1–31. doi:10.1017/S0269888903000651
- Khattak, A. M., Latif, K., Khan, S., & Ahmed, N. (2008). Managing change history in web ontologies. In *Semantics, Knowledge and Grid, 2008. SKG'08. Fourth International Conference* (pp. 347-350). IEEE.
- Khattak, A. M., Pervez, Z., Khan, W. A., Khan, A. M., Latif, K., & Lee, S. Y. (2015). Mapping evolution of dynamic Web ontologies. *Information Sciences*, 303, 101–119. doi:10.1016/j.ins.2014.12.040

- Kirsten, T., Gross, A., Hartung, M., & Rahm, E. (2011). GOMMA: A component-based infrastructure for managing and analyzing life science ontologies and their evolution. *Journal of Biomedical Semantics*, 2(6), 1–24. doi:10.1186/2041-1480-2-6 PMID:21914205
- Lutz, C., Walther, D., & Wolter, F. (2007). Conservative Extensions in Expressive Description Logics. *International Joint Conference on Artificial Intelligence (IJCAI)*, 7, 453–458.
- Lutz, C., & Wolter, F. (2010). Deciding Inseparability and Conservative Extensions in the Description Logic EL. *Journal of Symbolic Computing*, 45(2), pp194–pp228. doi:10.1016/j.jsc.2008.10.007
- Martins, H., & Silva, N. (2009). A User-driven and a Semantic-based Ontology Mapping Evolution Approach. *ICEIS*, (1), 214–221. doi:10.5220/0002011002140221
- Meilicke, C. (2011). *Alignments Incoherency in Ontology Matching* (Ph.D. thesis). University of Mannheim.
- Meilicke, C., & Stuckenschmidt, H. (2009). An efficient method for computing alignment diagnoses. In *International Conference on Web Reasoning and Rule Systems* (pp. 182-196). Springer Berlin Heidelberg. doi:10.1007/978-3-642-05082-4_13
- Mohammadi, M., Atashin, A. A., Hofman, W., & Tan, Y. H. (2019). SANOM results for OAEI 2019. *CEUR Workshop Proceedings*, 2536, 164–168.
- Peppas, P. (2008). Belief revision. In F. Van Harmelen, V. Lifschitz, & B. Porter (Eds.), *Handbook of knowledge representation* (pp. 317–359). Elsevier. doi:10.1016/S1574-6526(07)03008-8
- Pesquita, C., Faria, D., Santos, E., & Couto, F. M. (2013). To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. *Ontology Matching Workshop (OM)*, 13–24.
- Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1), 57–95. doi:10.1016/0004-3702(87)90062-2
- Schlobach, S. (2005). Debugging and Semantic Clarification by Pinpointing. In *Eur. Sem. Web Conf. (ESWC)*, (pp. 226–240). Springer.
- Solimando, A., Jiménez-Ruiz E., & Guerrini G. (2016). Minimizing Conservativity Violations in Ontology Alignments: Algorithms and Evaluation. *Knowledge and Information Systems*.
- Tang, Y., Wang, P., Pan, Z., & Liu, H. (2018). Lily results for OAEI 2018. *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference*, 179–186.
- Zahaf, A. (2012). *Alignment between versions of the same ontology*. ICWIT.
- Zahaf, A. (2017). *Alignment Evolution under Ontology Change: A formal Framework and Tools* (Ph.D. thesis). University of Sidi Bel Abbès. Algeria.
- Zahaf, A., & Malki, M. (2016). Alignment Evolution under Ontology Change. *International Journal of Information Technology and Web Engineering*, 11(2), 14–38. doi:10.4018/IJITWE.2016040102

ENDNOTES

- ¹ <http://bioportal.bioontology.org>
- ² <http://agroportal.lirmm.fr/>
- ³ <http://oaei.ontologymatching.org/>
- ⁴ <https://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html>
- ⁵ <https://www.w3.org/TR/owl2-overview/>
- ⁶ <https://www.lirmm.fr/yam-plus-plus/>
- ⁷ <http://oaei.ontologymatching.org/>
- ⁸ <https://cse.seu.edu.cn/people/pwang/lily.psp>
- ⁹ <http://infotechsoft.com/products/asmov.aspx>

Yahia Atig is an Assistant Professor at the Institute of Sciences and Technologies, Dept. Mathematics and Computer Science, of University Center of Naama, Algeria. He obtained his engineering degree in 2009 from Saïda University, Algeria and a magister's degree in computer science in 2013 from ESI Algiers, Algeria. He is currently pursuing his doctorate of science in computer science at Sidi Bel Abbes University in Research Team at the "Evolutionary Engineering and Distributed Information Systems Laboratory", EEDIS. His research interests include semantic web, ontology engineering, information and knowledge management.

Ahmed Zahaf received his engineer degree in computer science from Oran University, Algeria, in 1994, and doctorate of science in computer science from Sidi Bel Abbes University, Algeria, in 2017. Currently, he is a lecturer at Dr Tahar Moulay, University of Saida, Algeria. His research interests include semantic web, Linked data, ontology engineering, knowledge management and information systems.

Djelloul Bouchiha received his engineer degree in computer science from Sidi Bel Abbes University, Algeria, in 2002, and M. Sc. in computer science from Sidi Bel Abbes University, Algeria, in 2005, and Ph. D. in 2011. Between 2005 and 2010, he joined the Department of Computer Science, Saida University, Algeria, as a Lecturer. He became an Associate Professor since September 2013. Currently, he is a Full Professor at the University Center of Naama. His research interests include semantic Web services, Web reverse-engineering, ontology engineering, knowledge management and information systems.

Mimoun Malki is a full professor at the Ecole Supérieure en Informatique de Sidi Bel Abbes, Algeria, since 2015. He is the head of the LabRI-SBA Laboratory. He served as a full professor of computer engineering and the head of Evolutionary Engineering and Distributed Information Systems Laboratory (EEDIS Lab.) at Djillali Liabes University of Sidi Bel-Abbes, Algeria (2002- 2015). His research interests include Databases, Information Systems Interoperability, Ontology Engineering, Semantic Web Services, Linked Data Services, Web Reengineering, Enterprise Mashup, Cloud Computing, Semantic Web of Things and Big Data Privacy. He has published papers in well-known journals (IEEE TKDE, Springer WWW, Elsevier FGCS, Springer Computing, etc.).