

Conservativity Principle Violations for Ontology Alignment: Survey and Trends

Yahia Atig

Ctr Univ Naama, Inst. Sciences and Technologies, Dept. Mathematics and Computer Science, Algeria

E-mail: y_atig@esi.dz

Ahmed Zahaf and Djelloul Bouchiha

University Dr. Tahar Moulay of Saida, Algeria

Ctr Univ Naama, Inst. Sciences and Technologies, Dept. Mathematics and Computer Science, Algeria

E-mail: Ahmed.zahaf@univ-saida.dz, bouchiha.dj@gmail.com

Abstract—Ontology matching techniques are a solution to overcome the problem of interoperability between ontologies. However, the generated mappings suffer from logical defects that influence their usefulness. In this paper we present a detailed analysis of the problem so-called conservativity principle; alignment between ontologies should never generate new knowledge compared to those generated by reasoning solely on ontologies. We also study the sub-problems; Ontology change and Satisfiability preservation problems and compare the related works and their way to detect and repair conservativity principle. At the end we present a set of open research issues.

Index Terms—Conservativity Principle Violations, Ontology Alignment, Ontology Matching, Semantic Web.

I. INTRODUCTION

The alignment between ontologies is a crucial task in many application domains [3]. As not exhaustively, we can cite: Semantic web, communication in MAS (Multi-Agent System), data warehouse, integrating schema/ontologies, etc. Ontology is defined as the conceptualization of objects recognized as existing in a domain, with their properties and linking relationships. The problem is that given the same domain or related domains, it is possible that several ontologies are available (developed simultaneously by several different communities). The comparison of two ontologies, passing the one to the other or integrating them becomes therefore necessary.

This necessity does not make alignment faultless and impeccable, since mappings can lead to many undesirable logical consequences in the aligned ontologies and therefore the domain covered by these ontologies. In [13] three principles were proposed to minimize the number of potentially unintended consequences, namely: (i) consistency principle, the mappings should not lead to unsatisfiable classes in the integrated ontology, (ii) locality principle, the mappings should link entities that have similar neighborhoods, (iii) conservativity principle, the mappings should not introduce new semantic

relationships between concepts from one of the input ontologies. These principles have been actively investigated in the last years (e.g., [18], [25], [10], [13], [12], [17], [21]). The conservativity principle has been identified for instance in [13] as an alignment which allows the interaction between ontologies, rather than providing a new description of the domain. However, [23] proposes a different variant of the conservativity principle where the integrated ontology O_u must not introduce new subsumption relationships between concepts within the input ontologies.

In this paper we focus on the conservativity principle for ontology alignment. Actually, we achieved a thorough survey and make the following contributions:

- We formally define and illustrate the conservativity principle problem, highlighting the complexity of the problem. We modify and adapt an example presented in [23] which is a use case based on the Optique's¹ application domain.
- We systematically review the literature on the conservativity principle problem, offering a complete state-of-the-art by presenting, comparing and discussing the existing approaches.
- We analyze lacks of existing approaches, discussing general open issues which make difficult to deal with conservativity principle violations. This allows us to underscore open research challenges.

We structure the remainder of this paper as follows: Section 2 summarizes the basics concepts and definitions we will rely on along the paper. In Section 3, we introduce our problematic after analyzing some definitions mentioned in literature. This section is also an examination of the conservativity principle problem studied in several related works. Section 4 is a comparison of different surveys performed about alignment maintenance on basis of the studied sub-problems. Section 5 present some statistics presents on one side revealing the importance of this field and the

¹ <http://optique-project.eu/>

other side a numerical comparison between approaches and surveys studied here. Finally, Section 6 discusses our findings and challenges of different nature, representing open research issues and wraps up with concluding remarks and outlines future work.

II. PRELIMINARIES AND NOTATIONS

In this section, we define the edges of the conservativity principle problem. So we define some important notions for our work.

The concept of Ontology can be seen as a logical theory [14]. So it is a pair (S, A) , where S is the signature describing the vocabulary, and A is a set of axioms specifying the intended interpretation of the vocabulary in a domain of discourse. The signature is the set $S = C \cup P \cup I$. C represents the vocabulary to designate concepts. P is the vocabulary to designate properties and I is the vocabulary to designate individuals. We distinguish between the origins axioms A and their logical consequences A^* (also called closure). Theory (S, A) is called the presentation of (S, A^*) . In this work, we limit ourselves only to $S = C \cup P$ and we designate by ontological entity a concept or a property.

Ontology alignment is the task to detect links between elements from two ontologies. These links are referred as correspondences and express semantic relations. According to Euzenat and Shvaiko [6] we define a correspondence as follows and introduce an alignment as set of correspondences.

Definition 1 (Correspondence and Alignment). Given two ontologies O_1 and O_2 , let Q a function that defines sets of matchable elements $Q(O_1)$ and $Q(O_2)$. A correspondence between O_1 and O_2 is a 5-tuple (id, e_1, e_2, r, n) such that, id a unique identifier, $e_1 \in Q(O_1)$, $e_2 \in Q(O_2)$, r is a semantic relation, and $n \in [0; 1]$ is a confidence value. An alignment M between O_1 and O_2 is a set of correspondences between O_1 and O_2 . We restrict r to be one of the semantic relations from the set $\{\subseteq, \supseteq, \equiv, \perp\}$

In order to reason about alignment, two classes of approaches have been introduced. The first class is based on model theory. IDDL [29] and DDL [2] are two examples of approaches of this class. Based on an axiomatic approach, the second class called reductionist semantics [16] is to interpret correspondences of the alignment as axioms in some merged ontology. In this paper, 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 merged ontology.

Definition 2 (Merged Ontology). 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 merged ontology is defined by $O_1 \cup_M O_2 = O_1 \cup O_2 \cup trans(M)$.

After defining the most important notions for the

conservativity principle problem, we illustrate the problem itself.

III. ANALYSIS AND EXAMINATION

In order to analyze the conservativity problem through all its sides, we discuss in the present part of the paper the problem statement which will allow us to: first, defining the principle, and thereafter, comparing our definition against others approaches in the literature.

A. Problem Statement

This section is organized according to the following points: example of motivation, problem definition and comparison with other definitions mentioned in the literature.

1) Motivating example

As a motivating example we modify and adapt an example described in [23] which is a use case based on the Optique's application domain. As adaptation of the Optique's ontology O_1 , we add the following property: α_{10} : $Well \sqsubseteq \exists hasOwner.Owner$, to indicate that each *Well* has an *Owner*.

Table 1 shows the fragments of two ontologies in the context of the Oil and Gas industry. The ontology O_1 has been directly bootstrapped from a relational database in Optique, and it is linked to the data via direct ontology-to-database mappings. The ontology O_2 , instead, is a domain ontology, based on the NPD FactPages, preferred by Optique end-users to feed the visual query formulation interface².

The integration via ontology matching of O_1 and O_2 is required since the vocabulary in O_2 is used to formulate queries, but only the vocabulary of O_1 is connected to the database. Consider the set of mappings M in Table 2 between O_1 and O_2 generated by an off-the-shelf ontology alignment system. As described in Section 2, mappings are represented as 5-tuples; for example the mapping m_1 suggests an equivalence relationship between the entities $O_1:Well$ and $O_2:Well$, with confidence 0.9.

2) Problem definition

In this paper we propose a general definition of the conservativity of alignment, covering any violations of the principle for which the alignment must not introduce any new entailments to the input ontologies.

Definition 3 (Conservatif Alignment). An alignment A between two ontologies O_1 and O_2 is *conservatif* iff $(O_1 \cup_A O_2) \models \delta \rightarrow \exists i \in \{1, 2\} O_i \models \delta \vee \delta \in A$, i.e. any reasoning on the set $\{O_1 \cup_A O_2\}$ that leads to logical consequences δ must not surpassed the set of entailments generated by reasoning on $\{O_1, O_2\}$ separately.

The reasoning on the set $\{O_1 \cup_A O_2\}$ however,

² In Optique OWL2 QL ontologies are used for query rewriting. The query formulation may be based on much richer OWL2 ontologies. The axioms that fall outside the OWL2 QL profile are either approximated or not considered for the rewriting.

violates the conservativity principle, according to our definition of conservativity of alignment (Definition 3), and introduces new entailments (see Table 3) to the input ontologies O_1 and O_2 .

Table 1. Fragments of the ontologies used in Optique

Ontology O_1	Ontology O_2
α_1 WellBore \subseteq \exists belongsTo.Well	β_1 Exploration_well \subseteq Well
α_2 WellBore \subseteq \exists hasOperator.Operator	β_2 Explorborehole \subseteq Borehole
α_3 WellBore \subseteq \exists locatedIn.Field	β_3 Appraisal_exp_borehole \subseteq Explor_borehole
α_4 AppraisalWellBore \subseteq WellBore	β_4 Appraisal_well \subseteq Well
α_5 ExplorationWellBore \subseteq WellBore	β_5 Field \subseteq \exists hasFieldOperator.Field_operator
α_6 Operator \subseteq Owner	β_6 Field_operator \cap Owner \subseteq Field_owner
α_7 Operator \subseteq Company	β_7 Company \subseteq Field_operator
α_8 Field \subseteq \exists hasOperator.Company	β_8 Field_owner \subseteq Owner
α_9 Field \subseteq \exists hasOwner.Owner	β_9 Borehole \subseteq Continuant \cup Occurrent

Table 2. Ontology mappings for the vocabulary in O_1 and O_2

Alignment A				
id	e_1	e_2	n	ρ
m_1	O_1 :Well	O_2 :Well	0.9	\equiv
m_2	O_1 :WellBore	O_2 :Borehole	0.7	\equiv
m_3	O_1 :ExplorationWellBore	O_2 :Exploration_well	0.6	\subseteq
m_4	O_1 :ExplorationWellBore	O_2 :Explor_borehole	0.8	\equiv
m_5	O_1 :AppraisalWellBore	O_2 :Appraisal_exp_borehole	0.7	\equiv
m_6	O_1 :Field	O_2 :Field	0.9	\equiv
m_7	O_1 :Operator	O_2 :Field_operator	0.7	\supseteq
m_8	O_1 :Company	O_2 :Company	0.9	\equiv
m_9	O_1 :hasOperator	O_2 :hasFieldOperator	0.6	\equiv
m_{10}	O_1 :Owner	O_2 :Owner	0.9	\equiv

Table 3. Example of conservativity principle violations

σ	Entailment:	follows from:	Violation?
σ_1	O_2 :Explor_behole \subseteq O_2 :Exploration_well	m_3, m_4	YES
σ_2	O_1 :AppraisalWellBore \subseteq O_1 :ExplorationWellBore	β_3, m_4, m_5	YES
σ_3	O_2 :Field_operator \subseteq O_2 :Field_owner	$\alpha_6, \beta_6, m_7, m_{10}$	YES
σ_4	O_1 :Company \equiv O_1 :Operator	$\alpha_7, \beta_7, m_7, m_8$	YES
σ_5	O_1 :Company \subseteq O_1 :Owner	σ_4, α_6	YES
σ_6	O_2 :Company \subseteq O_2 :Field_owner	σ_3, σ_5	YES
σ_7	O_2 :Well \subseteq O_2 :Owner	m_1, m_{10}, α_{10}	YES

We have shown that the alignment violating the conservativity principle leads to non-desired entailments to the input ontologies. Therefore, a comparison between the different works on the conservativity principle can be considered as very important.

3) Comparison of definitions

In order to position our definition of conservativity principle problem, we present in this part of the paper a comparison between several definitions provided in the literature. Since the Satisfiability preservation and Ontology change preservation are two instances of conservativity problem, this comparison is a classification of approaches in three dimensions: i. Approaches defining the Satisfiability preservation problem, ii.

Approaches defining the Ontology change preservation problem and iii. Approaches defining the Conservativity problem.

i. Satisfiability preservation problem

The satisfiability preservation of the alignment between ontologies was the subject of study in several works ([27], [25], [17] and [12]). In [27] the authors of Lily address the problem of debugging ontology mappings to improve the quality of mapping result. They define two types of inconsistencies:

- **Mappings that form a circle:** such type of unsatisfiability means that the mapping should not destroy the hierarchy structure (*is-a* structure) in the ontology, for example: let's take $(e_1, e'_1) \in$ ontology O_1 , $(e_2) \in$ ontology O_2 . The following mappings form a circle leading to alignment inconsistency: $m_1: e_1 \subseteq e'_1$ and $m_2: e'_1 \subseteq e_2$ and $m_3: e_1 \equiv e_2$. Here, the equivalent mapping is treated as bidirectional *is-a* relation. The *is-a* circle destroys the hierarchy of ontology O_1 .
- **Mappings that do not meet the equivalentClass/disjointWith axioms:** in such case, alignment between ontologies should not introduce equivalences between disjoint elements in the inputs ontologies, for example: let's take $(e_1) \in$ ontology O_1 , $(e_2, e'_2) \in$ ontology O_2 . The following mappings lead to alignment inconsistency: $m_1: e_2 \perp e'_2$ and $m_2: e_1 \equiv e_2$ and $m_3: e_1 \equiv e'_2$. Here, the behavior of the alignment is inconsistent since it leads to two contradictory

mappings $m_1: e_2 \perp e'_2$ and $m_1: e_2 \perp e'_2$ and $m_4: e_2 \equiv e'_2$.

Stuckenschmidt et al. [25] proposed a theory for reasoning about ontology mappings. This work identified four properties that reflect the quality of a mapping, namely *containment*, *minimality*, *consistency* and *embedding*. The consistency principle claims that a mapping is consistent if it does not make a satisfiable concept in the target terminology unsatisfiable.

Meilicke [17] identifies the (in) coherence of ontology as: an ontology is called incoherent when there exists an unsatisfiable named concept or property; otherwise the ontology is called coherent. A concept C is defined to be unsatisfiable *iff* each model I of O maps C to the empty set, i.e., an instance of C cannot exist for logical reasons. Thus, a named concept or property $C_{\#i}$ with $i = \{1, 2\}$ is unsatisfiable due to A with respect to O_1 and O_2 *iff* $C_{\#i}$ is satisfiable in O_i and unsatisfiable in A . the definition which interests us is the alignment incoherence definition: Given an alignment A between ontologies O_1 and O_2 with signatures S_1 and S_2 respectively, A is incoherent with respect to O_1 and O_2 *iff* there exists $C_{\#i} \in S_i$ with $i = \{1, 2\}$ that is unsatisfiable due to A with respect to O_1 and O_2 . Otherwise, A is coherent with respect to O_1 and O_2 .

LogMap [12] introduces the notion of Logical inconsistencies. Indeed, the ontology $O_1 \cup O_2 \cup M$ resulting from the integration of O_1 and O_2 via mappings M may entail axioms that don't follow from O_1 , O_2 or M alone.

ii. Ontology change preservation problem

[28] defines the notion of *conserving the changed meaning* to refer the control of the propagation of knowledge from one version to another which is one of the known activity of alignment. If this propagation is not controlled, it can affect the meaning of ontological elements. An alignment M between two versions O_1 and O_2 conserves the changed meaning *iff* M verifies the following two properties:

$$\forall \delta \in A^-, (O_1 \cup_M O_2) \not\models M(\delta)$$

$$\forall \delta \in A^+, (O_1 \cup_M O_2) \not\models M^-(\delta)$$

Such as: A^- is the set of deleted axioms. A^+ is the set of added axioms and M^- is the set of deleted mappings between two versions of the same ontology.

iii. Conservativity problem

In this section we explore various definitions of the conservativity problem of alignment between ontologies, for instance, in [13] an interesting definition of the conservativity principle was proposed. This definition required that, given an ontology source (say, O_1) and the mappings M , the union $(O_1 \cup M)$ should not introduce new semantic relationships between entities from O_1 . This definition takes only the ontology source and the alignment and don't take the target ontology in

consideration. However, this can be a subject of many neglected logical consequences when discarding the target ontology. Indeed, the following example presents a concrete case.

Example. After the adaptation of the Optique's ontology O_1 , with the property: $\alpha 10: Well \subseteq \exists hasOwner.Owner$, to indicate that each *Well* has an *Owner*. Thus, in such case, the union $(O_2 \cup M) \not\models O_2: Well \subseteq O_2: Owner$.

Another definition of conservativity principle [23] is given based on the definition cited in [13]. In this work the authors propose a different variant of the conservativity principle where they require that the integrated ontology O_u (i.e., $O_u = O_1 \cup O_2 \cup M$) does not introduce new subsumption relationships between concepts from one of the input ontologies, unless they were already involved in a subsumption relationship or they shared a common descendant. As it is clear, this definition deals with conservativity principle violations only at the concept hierarchy level within the input ontologies, thereby it is also considered as incomplete to cover all types of conservativity principle violations.

To achieve our survey about different works on the conservativity principle, we present in the following (Table 4) a comparative table between several works according to their problem definitions.

Table 4. Comparison between approaches according to their problem definitions

Approach	Satisfiability preservation	Ontology change preservation	Conservativity
[27]	+	-	-
[17]	+	-	-
[10]	+	-	-
[12]	+	-	-
[28]	+	+	-
[13]	+	+	+(*)
[23]	+	-	+(*)
Our definition	+	+	+

(*): Here, the conservativity of alignment between ontologies is an incomplete process for reasons already discussed in the last section (*iii. Conservativity problem*).

We have shown that most systems ([27], [17], [10] and [12]) deal only with the satisfiability preservation problem. However, [28] addresses a more complicated problem: Ontology change preservation, which needs more sophisticated violations detection processes. the rest of the compared systems here ([13] and [23]) solve the conservativity problem with different degrees, since that [23] deals with conservativity principle violations at only the concept hierarchy level within the input ontologies, and therefore it cannot covers all types of violations even those concerning ontology change preservation. Whereas, [13] deals with the conservativity problem in a partial manner as discussed in the last section (*iii. Conservativity problem*). Our definition (*III.A.2 Problem definition*) is more general, covering any violations of the

conservativity principle for which the alignment must not introduce any new entailments to the input ontologies. After this comparison, the analysis of the violations detection processes adopted by the mentioned systems arises as an important task to ensure a complete survey.

B. Conservativity violation detection

The present section highlights violation detection by analyzing at first approaches that address the general problem (*conservativity principle*) then, its instances (*ontology change* and *satisfiability preservation*).

Violations detection of conservativity principle was subject of study in [23] and [13]. As mentioned in the section above (*iii. Conservativity problem*), [13] states that the conservativity principle is based on the purpose of M , which is to enable the interaction between O_1 and O_2 , rather than to provide a new description of the domain. Indeed, the authors use a specific pattern to detect conservativity principle violations; this pattern is based on the following observation:

The OWL2 alignment M that encodes the contents of UMLS-Meta³ contains only axioms of the form *EquivalentClasses*(e_1 e_2) where e_1 is mentioned only in O_1 and e_2 is mentioned only in O_2 (note that different ontology sources use different namespaces to refer to their entities). This observation is used to significantly simplify the problem in the following way: O_1 violates conservativity *iff* there exist axioms *EquivalentClasses*(e_1 e_2) and *EquivalentClasses*(e'_1 e_2) in M , with e_1 and e'_1 different entities in O_1 , such that O_1 alone does not imply the axiom *EquivalentClasses*(e'_1 e_2). If this is the case, then the mappings *EquivalentClasses*(e_1 e_2) and *EquivalentClasses*(e'_1 e_2) from M are in conflict and one of them may be incorrect.

In order to identify such conflicting mappings, it suffices to (syntactically) check in M whether two entities from one of the sources are mapped to the same entity in the other source, and then check (semantically) whether these two entities were already equivalent with respect (only) to the former source. These checks can be performed efficiently in practice: the former is syntactic, and the latter involves a single semantic test using an ontology reasoner.

Section (*iii. Conservativity problem*) also indicates another variant of the conservativity principle cited in [23], where the integrated ontology O_u must not introduce new subsumption relationships between concepts within the input ontologies. This variant of the conservativity principle follows the assumption of disjointness proposed in [22]. So if two atomic concepts A, B from one of the input ontologies are not involved in a subsumption relationship nor share a common subconcept (excluding \perp), they can be considered as disjoint. Hence, the problem of detecting and solving conservativity principle violations, is reduced to a mapping (incoherence) repair problem, if the input ontologies are extended with sufficient disjointness axioms. The detection of conservativity principle violations is done in the same

way as *LogMap* (examined below).

For detecting ontology change preservation violation, [28] considers that initial alignment cannot be coherent. Because, some correspondences propagate axioms from one ontology version to another; this violates the constraint of *conserving the changed meaning*. The goal is to identify these correspondences and provide means to choose among them which must be eliminated. The identification of these correspondences is simply obtained by identifying the signature of the propagated axiom. To choose among correspondences, the author introduces an order relation called *relevance relation* on the signature elements of the propagated axiom. The relevance relation (noted $<_{rel}$) compares the degrees of intentional persistence of these elements. The intentional persistence of an element signature s denoted (*intPersistence*(s)) is expressed as the ratio of the number of occurrences of this element in the set of persistent axioms (denoted *nboccurrence*(s, A^P_i) for a version i) on the total number of persistent axioms. Formally defined:

$$s_1 <_{rel} s_2 \text{ iff } intPersistence(s_1) < intPersistence(s_2) \text{ and } intPersistence(s) = nboccurrence(s, A^P_i) / |A^P_i|.$$

For the detection of consistency principle violations we will discuss some of the most famous methods ([27], [17], [10] and [12]) treating the unsatisfiability of alignment between ontologies.

Authors of *Lily* [27] define two types of inconsistencies: *i.* Mappings that form a circle and *ii.* Mappings that do not meet the *equivalentClass/disjointWith* axioms mentioned in the original ontology. Therefore the authors use an algorithm that combines the two ontologies to align (the alignment between them is a single graph (*is-a*)), and detects the paths which constitute a circle to inform the user of inconsistent mappings by considering them as wrong.

Alcamo's approach [17] can only ensure the coherence of alignments between ontology TBoxes, by applying preprocessing step to any reasoning activities by removing the ABox of O_1 and O_2 . An iterative algorithm on the entire signature (concepts and properties) of the alignment between two ontologies is proposed to detect unsatisfiable entities. This algorithm detects entities representing unsatisfiable logical consequences of the signature of alignment A between O_1 and O_2 , and checks if they are logical consequences of the signature of O_1 and O_2 . Meilicke [17] identifies the notion of MIPS (Minimal Incoherence Preserving Sub-alignment) and MUPS (Minimal Unsatisfiability Preserving Sub-alignment), to detect inconsistency and unsatisfiability in a sub-alignment (note that MIPS (A, O_1, O_2) \subseteq MUPS (A, O_1, O_2)), and proposes a variant algorithm (expand-and-shrink-algorithm) for debugging incoherent alignments.

ASMOV [10] introduces the notion of mapping validation, a graph built from the alignment and information of the ontologies. Two different constructs constitute this graph: nodes and edges. The nodes contain pairs of entities, whereas the edges contain pairs of properties. The validation process is done in three phases:

³ <http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html>

concept validation, property validation and concept-property validation. In the first two phases, the considered edges (three types: *is-a*, *same-as* and *disjoint-from*) are created using the predefined properties of the ontology. The validation of the graph is reduced to an investigation of edge violations; a node may not be valid if one or more of the edges are violated. If an edge violation exists, only the linked nodes are investigated.

The detection of consistency principle violations was also studied in LogMap [12]. The core of LogMap is an iterative process that alternates *mapping repair* and *mapping discovery* steps. In each iteration, LogMap maintains two structures.

1. A working set of *active mappings*, which are mappings discovered in the preceding iteration. Mappings found in earlier iterations are established, and cannot be eliminated in the repair step. In the first iteration, the active mappings coincide with the set of anchors.
2. For each anchor, LogMap maintains two contexts (one per input ontology), which can be expanded in different iterations. Each context consists of a set of classes and has a distinguished subset of active classes, which is specific to the current iteration. In the first iteration, the contexts for an anchor $C_1 \equiv C_2$ are $\{C_1\}$ and $\{C_2\}$ respectively, which are also the active classes.

Thus, active mappings are the only possible elements of a repair plan, whereas contexts constitute the basis for mapping discovery.

Violations detection alone is not enough. For this, the phase of repairing violations is also of major importance because it ensures us an acceptable quality of alignment.

C. Conservativity violation repair

Conservativity violation repair is a process aiming to correct violations, output of the previous detection phase. The goal of this part is to uncover repair strategies used by the systems under study.

The conservativity principle proposed in [13] suggests that the obtained pairs of mappings which lead to violations are in conflict and (at least) one of them in each pairs is likely to be incorrect. Actually, the locality principle⁴ is proposed to compute a confidence value⁵ for each conflicting mapping, which can then be exploited for (partially) automating the disambiguation process.

In [23], the detection of conservativity principle violations is done in the same way as LogMap. It uses the mapping (incoherence) repair algorithm presented in [12] and [23] for the extended Horn propositional theories P_1^d and P_2^d and the input mappings M . The mapping repair process exploits the Dowling-Gallier algorithm for propositional Horn satisfiability [5] and checks, for every

⁴ If two entities e_1 and e_2 from ontologies O_1 and O_2 are correctly mapped, then the entities semantically related to e_1 in O_1 are likely to be mapped to those semantically related to e_2 in O_2 [13].

⁵ Since UMLS-Meta does not assign a confidence value to each mapping.

propositional variable $A \in P_1^d \cup P_2^d$, the satisfiability of the propositional theory $P_A = P_1^d \cup P_2^d \cup M \cup \{true \rightarrow A\}$. Satisfiability of P_A is checked in worst-case linear time in the size of P_A , and the number of Dowling-Gallier calls is also linear in the number of propositional variables in $P_1^d \cup P_2^d$. In the case of unsatisfiability, the algorithm also allows to record conflicting mappings involved in the unsatisfiability, which will be considered for the subsequent repair process. The unsatisfiability will be fixed by removing some of the identified mappings. In the case of multiple options, the mapping confidence will be used as a differentiating factor⁶.

The signature element that has the less intentional persistent with respect to the relevance relation allows to choose the correspondence to be eliminated from the initial alignment [28]. When two of the signature elements have the same degree of intentional persistence, the choice is left to the user.

Like program debugging, Lily [27] treats all suspicious mappings as two categories: errors and warnings. Apparently, errors are the confirmed wrong mappings, but warnings are the ones which may be wrong, right or imprecise. There are two proposed solutions for the two types of inconsistencies detected by Lily:

1. For *i*. Mappings that form a circle: authors use an algorithm that combines the two ontologies to be aligned and the alignment between them in a single graph (*is-a*), and goes through the paths which constitute a circle to inform the user of inconsistent mappings by considering them as wrong. The choice to delete one of the arcs forming the circle is left to the user.
2. For *ii*. Mappings that do not meet the *equivalentClass/disjointWith* axioms mentioned in the original ontology: Lily proposes two potential solutions: (1) Importing a complex concept and representing the mappings in the form: $m: e_1 \equiv e_2 \vee e'_2$, such as: $(e_1) \in \text{ontology } O_1$ and $(e_2, e'_2) \in \text{ontology } O_2$. (2) Giving the user the choice to delete one of the mappings in conflict.

Note that Lily considers only the mappings between concepts and only *equivalentClass/disjointWith* as axioms.

In the third phase of the mapping validation process of ASOMV [10], the concept validation graph is modified. All edges are dropped from the remaining valid nodes and are replaced by edges created from the valid nodes of the property validation graph. The new graph is then validated, but in this time the nodes are favored; thus, only the edges are invalidated. All invalid mappings that have been identified are added to the invalid mapping list. If at least one violation was identified, the iteration process resumes and the invalid source-target pairs are ignored.

Concerning the mapping repair in LogMap [12],

⁶ In the scenarios where the confidence of the mapping is missing (e.g., in reference or manually created mapping sets) or unreliable, this mapping repair technique computes fresh confidence values based on the locality principle [13].

authors use a Horn propositional logic representation of the extended hierarchy of each ontology together with all existing mappings (both active and established). LogMap splits each \equiv mapping into two Horn clauses (\rightarrow, \leftarrow). Thus, for the unsatisfiability checking, LogMap implements the well-known Dowling-Gallier algorithm [5] for propositional Horn satisfiability, and calls the Dowling-Gallier module once (in each repair step) for each class. LogMap takes as input a class C (represented as a propositional variable) and determines the satisfiability of the propositional theory P_C consisting of

- the rule ($true \rightarrow C$),
- the propositional representations P_1 and P_2 of the extended hierarchies of the input ontologies O_1 and O_2 , and
- the propositional representation P_M of the computed mappings.

IV. RELATED SURVEYS

Several surveys were performed over the last years about alignment maintenance ([4], [7] and [24]).

In this context, [4] provides a thorough survey on mapping maintenance affected by ontologies evolution, by presenting, comparing and discussing existing proposals in different categories (mapping revision, calculation, adaptation and representation). We discuss this survey within its own categorization:

- *Mapping revision.* This category is subdivided in two subcategories: i. Identification of invalid mappings and, ii. Repairing of mappings. Despite this promising naming and categorization, the study does not thorough in the approaches addressing the problem, e.g., the former subcategory takes into account only approaches identifying invalid mappings between less expressive knowledge representation models like relational database schemas, peer-to-peer systems or XML schemas. Furthermore, the last subcategory does not presents many details on how approaches address the repairing task, e.g., the survey invoke the LogMap [12] system but don't give any information about the processing.
- *Mapping calculation.* This part is less interesting for our work; however, it develops works adopting mapping calculation to cope alignment maintenance problem. This category is also splitted into two subcategories: i. Full recalculation, all mappings affected by ontologies evolution are considered as invalid ones, therefore they are deleted and recomputed from scratch. ii. Partial recalculation, for methods adopting partial calculating of invalid mappings.
- *Mapping adaptation.* Includes: i. Mapping composition, approaches which compose various mappings to generate new ones and especially original alignment between two ontologies with

alignment between ontology versions to generate the alignment between the new version and the other ontology. ii. Mapping rewriting in database schemas, works performing mapping adaptation by incrementally rewriting the elements in queries which represent mappings between database schemas. iii. Synchronization of models, scenarios requiring establishing mappings between heterogeneous models like database schemas and ontologies. iv. Change propagation, approaches highlighting the impact of knowledge systems (databases, thesauri, ontologies...) evolution to support the mapping adaptation. v. Mapping change strategies, approaches adapting mappings impacted by knowledge systems changes.

- *Mapping representation.* Proposals in this category focus on representing mappings to support maintenance and alignment with a particular emphasis on user interfaces and mapping description languages.

As strength point, this study succeeded to divide the alignment maintenance problem into relevant subcategories. This categorization lead to separate several issues discussed at the end of the paper:

- *Knowledge systems evolution.* Since information regarding the evolution of knowledge systems remains cornerstone for mapping maintenance, how to correctly and completely invest it?
- *Mapping interpretation.* The semantics of established mappings are poorly interpreted to propose changes in the maintenance process, how to deal with this lack?
- *Mapping adaptation.* How to design efficient adaptation strategies to guarantee that mappings remain valid after suffering ontology changes?
- *Knowledge systems model.* Issues studying interrelated knowledge systems based on heterogeneous models like ontologies and thesauri, or database schemas and taxonomies, whose expressiveness differs substantially.

In return, this analysis has not conducted a comparative study between the investigated systems. Indeed, no metrics proposed in the survey can allow the reader to actually evaluate the advantages and drawbacks of each system. Moreover, the huge expanding of research issues does not really help the search for a significantly advancement, e.g., issues: how to design efficient adaptation strategies to guarantee that mappings remain valid after suffering ontology changes? It cannot be considered as a significant contribution since it just describe the problem and do not trace a new path for research.

[7] presented a comprehensive survey on the notions of alignment Disambiguation and alignment Debugging. An alignment is ambiguous, when some entities are matched with several other entities (assuming that the relation is equivalence), e.g., a $?:?$ alignment is expected but a $*:*$

alignment has been returned. A simple method for dealing with this problem is to always choose the correspondence with the higher confidence (greedy algorithm). An alternative solution is to suppose that the correct match among two classes is prone to have other correct matches among its more general and more specific entities [26]. Alignment debugging aims at restoring consistency and coherence of the produced alignment. Consistency is characterized by the aligned ontologies having no models. Coherence is characterized by no model of the aligned ontologies allowing a particular class to have instances. In this category the authors present the most famous systems (LogMap [12], ASMOV [10], ALCOMO [17]) with other less known systems:

- ContentMap [11] can be considered as a constraint-based debugging tool with the constraints provided by users. It aims at helping users to understand and evaluate the consequences of the integration of two ontologies as well as to identify and handle possible errors.
- [15] used a naive Bayes [9] classifier for learning how to generate disjointness axioms in order to apply ontology repair techniques through inconsistency detection. Such a classifier is trained on various data sets and uses different similarity features (path distance, shared properties, similarity, instance sets) of pairs of classes, for deciding which ones are disjoint.
- [30] proposed restoring consistency only within spheres, which are local sets of ontologies and alignments.

Alignment Evolution has been also studied in [7]. According to the authors, managing alignments requires keeping them available in servers and making them evolve if necessary. Usually, alignment evolution corresponds to the creation of a new alignment, derived from an existing one. In this survey the different cases in which the alignment evolution is required are shown:

- Alignment evolution should be recorded within the alignment metadata (Annotations of alignments, or alignment metadata, record useful information for retrieving alignments or for explaining them) in addition to changes in the structure.
- An alignment may also evolve because it is no longer useful, being superceded by another one, or more generally, by the addition of further qualification to an alignment.
- Alignment evolution may also be triggered either by adding or by discarding correspondences manually produced, or by better methods, since new information is available.
- As soon as ontologies evolve, new alignments have to be produced following the evolution of the ontology. This can be achieved by transforming the changes made to ontologies into an alignment (from one ontology version to the next one), which

can be composed with the old alignment to obtain an updated alignment.

Another survey is presented in [24]. In this analysis, the author discusses detecting and correcting conservativity principle violations in ontology mappings by presenting the problem statement based on works of [12]. Several works dealing with the conservativity principle have been presented in this survey according to two categories:

1. *Approaches introduced the notion of Assumption of Disjointness:*
 - [22] originally introduced the assumption of disjointness to address the repair of ontologies underspecified in terms of negative constraints.
 - [25] applied the assumption of disjointness in the context of repairing ontology mappings, and limited the number of disjointness axioms to be inserted by using learning techniques.
 - In [8] the authors present an interactive system to guide the expert user in the manual enrichment of the ontologies with disjointness axioms.
2. *Ontology matching systems dealing with the conservativity principle:* in this part, authors presented three systems as example ASMOV [10], Lily [27] and YAM++ [19] which implemented different heuristics to avoid violations of the conservativity principle. In addition, another relevant approach [1] presents a set of sanity checks and best practices when computing ontology mappings.

To accomplish the comparison of different surveys performed on alignment maintenance, the following table (Table 5) summarizes the different problems studied in each survey.

Table 5. Comparison between different surveys according to the studied problems.

	Studied problem		
Approach	<i>Satisfiability Preservation Problem</i>	<i>Ontology change Preservation Problem</i>	<i>Conservativity Problem</i>
[7]	Alignment Debugging	Alignment Evolution	Not studied
[24]	Approaches introducing the notion of Assumption of Disjointness	Not studied	Ontology matching systems dealing with the Conservativity Principle
[4]	Mapping revision	Mapping calculation Mapping adaptation	Not studied

V. STATISTICS

The current section includes some statistics about approaches and surveys studied in this paper. The first figure (Fig 1) presents the number of articles (most of papers are cited in [20]) produced for each approach.

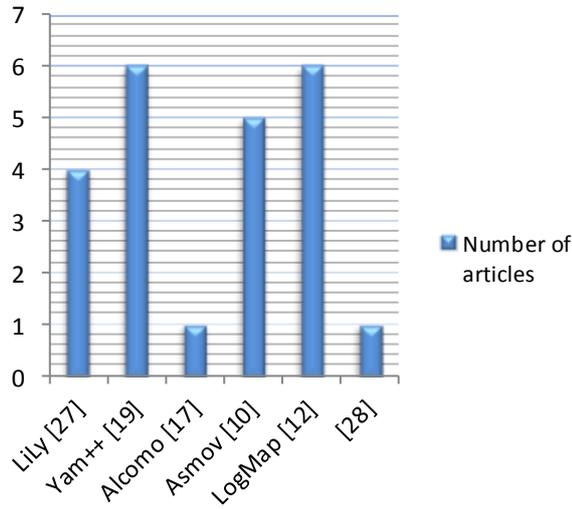


Fig.1. Number of articles produced for each approach

Fig 1 shows that the approaches [19] and [12] were the subject of the highest number of scientific productions. The smallest number of papers was devoted to the approaches [10] and [28].

Fig 2 is a bar graph including the number of proposed approaches for each sub-problem of the conservativity principle. Whereas Fig 3 shows the amount of open research questions for the conservativity principle suggested by each analysis studied in our survey.

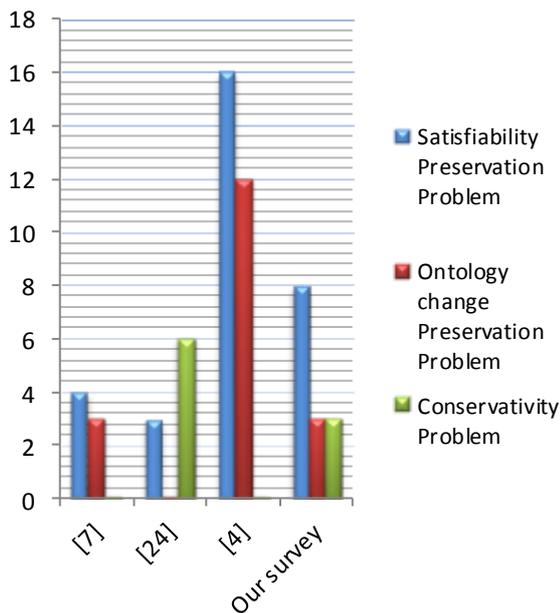


Fig.2. Number of proposed approaches for each conservativity sub-problem

Fig 2 shows that in [4] we find the largest number of the proposed approaches to address the problem of "Satisfiability Preservation" and "Ontology change Preservation". In our survey, we focus on the problem of "Conservativity" which is not presented in any of the other surveys.

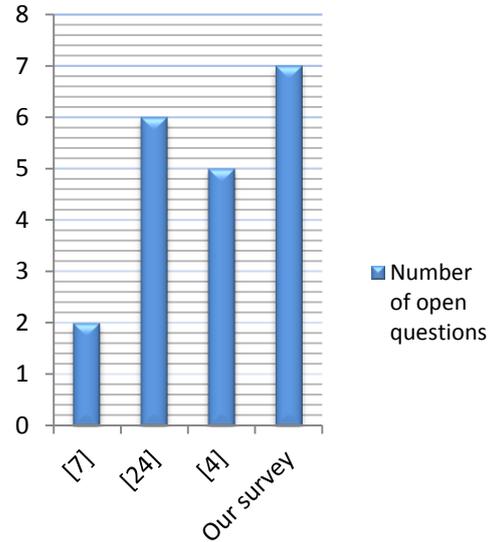


Fig.3. Number of open research questions suggested by each studied analysis

It is clear in Fig 3 that our survey provides the highest number of open problems in the field of "Conservativity Principle Violations".

VI. CONCLUSION AND RESEARCH CHALLENGES

This study allows pointing out open key issues, which existing approaches, addressing the conservativity principle, have neglected. Our main observation relies on the fact that literature did not deal with the conservativity principle problem in an effective and complete manner. The two main approaches identifying the conservativity of mappings ([13] and [23]) between ontologies have been a subject of many neglected logical consequences. First, they take only the source ontology and the alignment in consideration, and discard the target ontology in the process of detecting conservativity principle violations. Second, they also deal with conservativity principle violations only at the concept hierarchy level within the input ontologies, and therefore, drop out the others possible types of violations.

We still consider as research challenges the following questions:

- *Violation treatments.* Which appropriate methods could be applied to face the conservativity principle violations of alignment between ontologies? What are the possible ways to reduce the conservativity problem to a consistency problem which will allow reusing the available infrastructure and techniques for the mapping

repair?

- *The impact of violations.* Which consequences may have an alignment violating the conservativity principle in different application scenarios? Which impact could violations have on the aligned ontologies? What is the degree of the violations propagation? And which metrics can we use to measure this impact?
- *The cost of repairing mappings.* What is the cost needed to address conservativity principle violations?
- *Algorithm performance.* Which is the trade-off between completeness and runtime for these algorithms?

There are probably other research questions about the conservativity problem, but we consider that these issues are of great importance to ensure the quality of alignments between ontologies.

REFERENCES

- [1] Beisswanger, E., Hahn, U., et al.: Towards valid and reusable reference alignments-ten basic quality checks for ontology alignments and their application to three different reference data sets. *J. Biomed. Semant.* 3(suppl. 1), S4 (2012).
- [2] Borgida, A., Serafini, L.: Distributed description logics: Assimilating information from peer sources. *Journal on Data Semantics*, (2003).
- [3] DENNAI, A., BENSLIMANE, S.M.: A New Measure of the Calculation of Semantic Distance between Ontology Concepts. *International Journal of Information Technology and Computer Science (IJITCS)*. pp. 48-56, (2015).
- [4] Dos Reis J.C., Pruski C., Reynaud-Dela f're C.: State-of-the-art on mapping maintenance and challenges towards a fully automatic approach. *Expert Systems with Applications* 42 (2015), Elsevier, pp. 1465–1478, (2015).
- [5] Dowling W.F., Gallier, J.H.: Linear-Time Algorithms for Testing the Satisfiability of Propositional Horn Formulae. *J. Log. Prog.* 1(3), 267–284 (1984).
- [6] Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer Verlag, (2007).
- [7] Euzenat, J., Shvaiko, P.: *Ontology Matching*. Springer Heidelberg, (2013).
- [8] Ferré S., Rudolph, S.: *Advocatus Diaboli - Exploratory Enrichment of Ontologies with Negative Constraints*. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) *EKAW 2012. LNCS (LNAI)*, vol. 7603, pp. 42–56. Springer, Heidelberg (2012).
- [9] Good, I.J.: *The Estimation of Probabilities: an Essay on Modern Bayesian Methods*. MIT Press, Cambridge, (1965).
- [10] Jean-Mary, Y.R., Shironoshita, E.P., Kabuka, M.R.: *Ontology Matching With Semantic Verification*. *J. Web Sem.* 7(3), 235–251 (2009).
- [11] Jiménez Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: *Ontology integration using mappings: towards getting the right logical consequences*. In: *Proc. 6th European Semantic Web Conference (ESWC)*, Hersounisous, Greece. *Lecture Notes in Computer Science*, vol. 5554, pp. 173–188, (2009).
- [12] Jiménez-Ruiz, E., Cuenca Grau, B.: *LogMap: Logic-based and Scalable Ontology Matching*. In: *Int'l Sem. Web Conf. (ISWC)*. pp. 273–288 (2011).
- [13] Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga, R.: *Logic-based Assessment of the Compatibility of UMLS Ontology Sources*. *J. Biomed. Semant.* 2(Suppl 1), S2 (2011).
- [14] Kalfoglou, Y., Schorlemmer, M.: *Ontology mapping: the state of the art*. *The Knowledge Engineering Review* 18(1), 1–31, (2003).
- [15] Meilicke, C., Völker, J., Stuckenschmidt, H.: *Learning Disjointness for Debugging Mappings between Lightweight Ontologies*. In: *Int'l Conf. on Knowl. Eng. (EKAW)*. pp. 93–108 (2008).
- [16] Meilicke, C., Stuckenschmidt, H.: *An Efficient Method for Computing Alignment Diagnoses*. *Proceedings of the Third International Conference on Web Reasoning and Rule Systems (RR-09)*, Chantilly, Virginia, USA, (2009).
- [17] Meilicke, C.: *Alignments Incoherency in Ontology Matching*. Ph.D. thesis, University of Mannheim (2011).
- [18] Melnik, S., Garcia-Molina, H., Rahm, E.: *Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching*. In: *IEEE Int'l Conf. on Data Eng.* (2002).
- [19] Ngo, D., Bellahsene, Z.: *YAM++: A Multi-strategy Based Approach for Ontology Matching Task*. In: ten Teije, A., Völker, J., Handschuh, S., Stuckenschmidt, H., d'Acquin, M., Nikolov, A., Aussenac-Gilles, N., Hernandez, N. (eds.) *EKAW 2012. LNCS (LNAI)*, vol. 7603, pp. 421–425. Springer, Heidelberg, (2012).
- [20] Otero-Cerdeira L., Rodríguez-Martínez F.J., Gómez-Rodríguez A.: *Ontology matching: A literature review*. *Expert Systems with Applications* 42 (2015), Elsevier, pp. 949–971 (2015).
- [21] Santos, E., Faria, D., Pesquita, C., Couto, F.: *Ontology Alignment Repair Through Modularization and Confidence-based Heuristics*. arXiv: 1307.5322 preprint, (2013).
- [22] Schlobach, S.: *Debugging and Semantic Clarification by Pinpointing*. In: *Eur. Sem. Web Conf. (ESWC)*, pp. 226–240. Springer, (2005).
- [23] Solimando, A., Jiménez-Ruiz E., and Guerrini G.: *A Multi-strategy Approach for Detecting and Correcting Conservativity Principle Violations in Ontology Alignments*, In *Proceedings of the 11th International Workshop on OWL: Experiences and Directions (OWLED 2014)*, pp. 13-24, (2014).
- [24] Solimando, A.: *Detecting and Correcting Conservativity Principle Violations in Ontology Mappings*. In: P. Mika et al. (Eds.) *ISWC 2014, Part II, LNCS 8797*, pp. 545–552, (2014).
- [25] Stuckenschmidt, H., Serafini, L., Wache, H.: *Reasoning about Ontology Mappings*, In *ECAI2006 Workshop on Context Representation and Reasoning*, Ria del Garda, Italy, (2006).
- [26] Tordai, A.: *On combining alignment techniques*. PhD thesis, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands. pp. 65-193, (2012).
- [27] Wang, P., Xu, B.: *Debugging Ontology Mappings: A Static Approach*. *Computing and Informatics*. 27(1), pp 21–36, (2012).
- [28] Zahaf, A.: *Alignment between versions of the same ontology*. *Proc. 4th International Conference on Web and Information Technologies*, Sidi Bel Abbes, Algeria, (2012).
- [29] Zimmermann, A., Le Duc, C.: *Reasoning with a network of aligned ontologies*. *Proceeding of the 2nd International Conference on Web Reasoning and Rule systems*

(RR2008), (2008).

- [30] Zurawski, M., Smail, A., Robertson, D.: Bounded ontological consistency for scalable dynamic knowledge infrastructures. In: Proc. 3rd Asian Semantic Web Conference (ASWC), Bangkok, Thailand. Lecture Notes in Computer Science, vol. 5367, pp. 212–226, (2008).

Authors' Profiles



Yahia Atig is a lecturer at the Institute of Sciences and Technologies, Dept. Mathematics and Computer Science, of University Centre of Naama, Algeria. He received his Engineer and Magister degree in Computer Science in 2009 and 2013 respectively. He is currently pursuing his Ph.D. 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, Linked data, ontology engineering, knowledge management and information systems.



Ahmed Zahaf received his Engineer degree in computer science from Oran University, Algeria, in 1994, and M. Sc. in computer science from Sidi Bel Abbes University, Algeria, in 2005. 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 Assistant Professor since January 2011. Currently, he is an Assistant Professor at the University Center of Naama. His research interests include semantic web services, web reverse-engineering, ontology engineering, knowledge management and information systems.

How to cite this paper: Yahia Atig, Ahmed Zahaf, Djelloul Bouchiha, "Conservativity Principle Violations for Ontology Alignment: Survey and Trends", International Journal of Information Technology and Computer Science (IJITCS), Vol.8, No.7, pp.61-71, 2016. DOI: 10.5815/ijitcs.2016.07.09