Impact of Agricultural Ontologies Evolution on the Alignment Preservative Adaptation

Yahia Atig[1], Ahmed Zahaf[2], Nadri Khiati[2]

[1] University Center of Naâma, Algeria

[2] University of Saïda, Algeria

Abstract

Ontology matching techniques are a solution to surmount the problem of interoperability on the fly between ontologies. However, both alignments and ontologies are likely to be evolved throughout their life cycle, which frequently degrades their qualities. One of the main features of an alignment is its conservativity, so that it should never generate new knowledge compared to those generated by reasoning solely on ontologies. We focus in this paper on the issue of adapting the fresh alignment between evolved OWL-2 ontologies while respecting the conservativity principle. We also propose several patterns to deal with the problem of detection and repair of conservativity breaches during such evolution depending on the type of change in the related OWL-2 ontologies. We use famous ontologies from the field of agriculture to validate our experimentation. At the end we present a set of open research issues.

Keywords

Agricultural ontologies, matching adaptation, breaches identifying, breaches remedying, OWL-2 ontologies evolution, ontology matching, semantic web.

Atig, Y., Zahaf, A. and Khiati, N. (2024) "Impact of Agricultural Ontologies Evolution on the Alignment Preservative Adaptation", *AGRIS on-line Papers in Economics and Informatics*, Vol. 16, No. 3, pp. 17-32. ISSN 1804-1930. DOI 10.7160/aol.2024.160302.

Introduction

Several web applications seen in the last few years are essentially based on the Ontology Alignment task (Euzenat and Shvaiko, 2013). 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 perfect and faultless, since mappings can lead to many undesirable logical consequences in the aligned ontologies and therefore the domain covered by these ontologies. Alignment conservativity is one of three principles proposed in [Jiménez-Ruiz, 2011) to minimize the number of potentially unintended consequences. It intend to avoid introducing new semantic relationships between concepts from one of the input ontologies. Thereby, the alignment must allows interaction between ontologies, rather than providing a new description of the domain. Moreover, even if the alignment conservativity is well cared for during the calculation phase, or as a revision task just before its deployment, alignments such as ontologies are likely to be evolved throughout their life cycle (Stojanovic, 2004), and this evolution frequently degrades their qualities. As a result, alignments must be evolved and maintained in order to keep up with the change in ontology or to meet the demands of applications and users. In this work, we focus on the adaptation of the fresh alignment between evolved OWL-2 ontologies while respecting the conservativity principle, and make the following contributions:

- We formally define and illustrate the conservativity principle problem, highlighting the complexity of the problem. In addition, to present our problematic, we propose a concrete example of a non-conservative alignment evolution following the evolution of one of its input ontologies.
- We systematically review the literature on two main topics for our problematic which

are: the Conservativity Principle problem and Ontologies Alignment Evolution, offering a brief state-of-the-art by presenting and discussing the existing approaches.

- We propose a set of patterns, to adapt the fresh alignment (according to a conservative evolution) depending on the type of change applied to one of the input OWL-2 ontologies.
- Experimentation
- Discussion and open issues.

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 the Conservative Alignment Evolution which constitutes the background of our framework. Section 4 presents the (24*2) proposed patterns concerning the detection and repair process, following the source of the change. Section 5 discusses our findings. Section 6 is an examination of the conservativity principle problem studied in other related works. Finally, Section 7 examines challenges of different nature, representing open research issues and wraps up with concluding remarks and outlines future works.

Materials and methods

Preliminaries and notations. An ontology seen as logical theory is a pair (*S, A*), where *S* is a signature to designate a vocabulary and *A* is a set of axioms to specify the intended interpretation of this vocabulary in a domain of discourse. The signature of an ontology is the set $S = C \cup P \cup R \cup In$, where, C represents the set of vocabulary to designate concepts, *P* is the set of vocabulary to designate objects properties, *R* is the set of vocabulary to designate data properties and *In* is the set of 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 take into account all parts of *S*, such as: $S = C \cup P \cup R \cup In$ and we designate by ontological entity: a concept, a property or an individual.

Axioms act as constraints for interpretations of this vocabulary. An interpretation which satisfies all axioms of an ontology constitutes a model of that ontology.

Definition 1 (Ontology Model). An interpretation *I* is a model of an ontology *O* if and only if I satisfies every axiom δ in that ontology ($\forall \delta \in O, I \models \delta$).

Ontologies are expressed in logical languages such as RDF[1], RDFS[2] and OWL[3]. These languages provide a consequence relation between axioms of the language and ontologies. The W3C[4] proposes a finite set of OWL-2 axiom, subdivided into three subsets, to represent the different situations of expressivity that an ontology OWL2 can be found opposite, concretely :

- Class Expression Axioms: {SubClassOf, EquivalentClasses, DisjointClasses, DisjointUnion}.
- Object Property Axioms: {SubObjectPropertyOf, EquivalentObjectProperties, DisjointObjectProperties, InverseObjectProperties, ObjectPropertyDomain/Range, FunctionalObjectProperty, InverseFunctionalObjectProperty, Reflexive/IrreflexiveObjectProperty, Symmetric/AsymmetricObjectProperty, TransitiveObjectProperty}.
- Data Property Axioms: {SubDataPropertyOf, EquivalentDataProperties, DisjointDataProperties, DataPropertyDomain/Range, FunctionalDataProperty}.

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 (2013) we define a correspondence as follows and introduce an alignment as set of correspondences.

Definition 2 (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 $\{\sqsubseteq, \sqsupseteq, \equiv, \perp\}$.

The literature does not contain a standard for alignment semantics. Borgida and Serafini (2003) propose a distributed description logics semantics. The called reductionist semantics is

[1] https://www.w3.org/RDF/
[2] https://www.w3.org/wiki/RDFS
[3] https://www.w3.org/OWL/
[4] https://www.w3.org/TR/owl2-syntax/

a second approach which interprets correspondences of the alignment as axioms in some merged ontology Meilicke and Stuckenschmidt (2009) 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 3 (Natural Semantics). Given an alignment M between two ontologies O_1 and O_2 and trans: M→A, a function that transforms a correspondence to an axiom. The natural semantics of M is defined by the following aligned ontology:

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

Alignment consequence according to natural semantics is introduced as follows.

Definition 4 (Alignment Consequence). An axiom δ is an alignment consequence of an alignment M between two ontologies O_1 and O_2 if and only if δ is a logical consequence of the aligned ontology $O_1 \cup M\ O_2$.

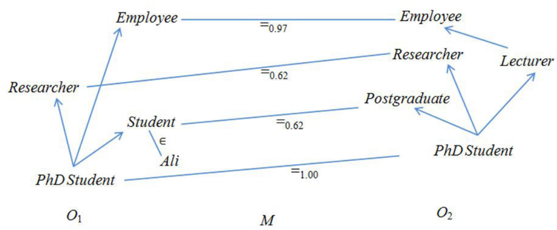
An axiom which is an alignment consequence either represents an ontological axiom or the image of a correspondence by the transformation function of the alignment.

Definition 5 (Ontology Signature Isomorphism). Given tow 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 $A_2 \vDash M(A_1)$ and $A_1 \vDash M^{-}(A_2)$, 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. We consider the conservativity principle as an alignment that allows interaction between ontologies, rather than providing a new description of the domain. However, two successive challenges are considered to overcome the problem of alignment conservativity violation Atig (2022). However, before detailing these challenges, let's start first by formally defining the conservative alignment.

Definition 6 (Conservative Alignment). An alignment M between two ontologies O_1 and O_2 is conservative if and only if for every ontological axiom δ that is not an image of any alignment correspondence, $(O_1 \cup_M O_2) \vDash \delta \rightarrow \exists i \in \{1, 2\} / O_i \vDash \delta$, i.e. any reasoning on the set $\{O_1 \cup M\ O_2\}$ that leads to logical consequences δ must not surpassed the set of entailments generated by reasoning on $\{O_1, O_2\}$ separately.

The concrete example in Figure 1 represents a scenario of non-conservative evolution, which is caused by a change affecting one of the input ontologies of an alignment. This scenario will allow us to reveal the problem of violating the conservativity principle in the case of evolving one related ontology.



Source: Authors

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

Example 1: Considering the alignment M of Figure 1. We use Description Logic like syntax to describe both ontologies. Also, we use the index number in ontologies notation as name space to designate entities. Table 1 shows the set of axioms within the input ontologies O_1 and O_2. Whereas, Table 2 presents the set of correspondences of the alignment M between these two ontologies.

Ontology O_1	**Ontology O_2**
α_1 PhD Student ⊆ Researcher	β_1 PhD Student ⊆ Lecturer
α_2 PhD Student ⊆ Student	β_2 Lecturer ⊆ Employee
α_3 Lecturer ⊆ Employee	β_3 PhD Student ⊆ Researcher
α_4 Student (Ali)	β_4 PhD Student ⊆ Postgraduate

Source: Authors

Table 1: Example of two educational domain ontologies

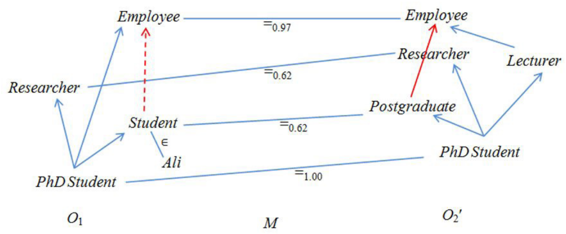
We use in Table 1, the α_i notification for ontology O_1 and the β_i notification for ontology O_2 to uniquely identify each axiom within these two ontologies.

Alignment *M*				
id	*e_1*	*e_2*	*n*	*ρ*
m_1	1 : *PhD Student*	2 : *PhD Student*	1.00	=
m_2	1 : *Researcher*	2 : *Researcher*	0.62	=
m_3	1 : *Employee*	2 : *Employee*	0.97	=
m_4	1 : *Student*	2 : *Postgraduate*	0.62	=

Source: Authors

Table 2: Correspondences for alignment *M* between ontologies O_1 and O_2.

Consider the set of correspondences m_i of the conservative alignment M of Table 2 between O_1 and O_2 generated by a generic ontology alignment system.



Source: Authors

Figure 2: Impact of evolving ontology O_2 into O_2'.

Assuming that one of the two input ontologies of the conservative alignment M has evolved. For example, and as illustrated in Figure 2, the ontology O_2 has evolved into O_2', and adds a new axiom $\beta_5 = \{2 : Postgraduate \subseteq 2 : Employee\}$, shown by the red solid arrow.

δ	Entailment:	follows from:	Violation?
δ_1	$1 : Student \subseteq 1 : Employee$	m_3, m_4, β_5	YES
δ_2	1: *Employee* (Ali)	δ_1, α_4	YES

Source: Authors

Table 3: Violations of the conservativity principle following the evolution of ontology O_2.

According to Table 3, after evolving the ontology O_2 towards O_2' (adding the axiom $\beta_5 = \{2 : Postgraduate \subseteq 2 : Employee\}$), the alignment M represented by the set of correspondences m_i violates the principle of conservativity according to the definition of conservative alignment (see definition 6), and introduces two new undesirable logical consequences (δ_1 shown by the red dashed arrow and δ_2) in the input ontology O_1. which represents an excess of the inferences generated by the reasoning on O_1 in isolation, and consequently an involuntary extension of the domain covered by this ontology. Therefore, the alignment M must be revised to restore its lost conservativity following the evolution of one of its input ontologies.

The alignment adaptation under ontology change problem aims to correct the alignment so that it fulfills its role in the interaction between ontologies, rather than generating new relationships within them, which provides a new description of the covered domains. However, this problem can be refined to include two sub-problems, namely: conservativity violations *detection* problem and *repair* problem.

Identifying Violations of Conservativity. The conservativity violations detection problem intends to designate the set of axioms causing violations of alignment conservativity upon evolving input ontologies. In fact, we are not interested here by the process of detecting ontological changes but by the impact of these changes on the alignment w.r.t conservativity principle. Despite, we must first identify the possible ontological changes to study this impact.

According to the W3C[5] definition, an OWL-2 ontology is a formal description of a domain of interest. The following three different syntactic categories composite the OWL-2 ontologies:

- Entities, represent the primitive terms that form the basic elements of an ontology. They are identified by IRIs, and composed by the set of *classes, properties*, and *individuals* which express the knowledge conveyed in a domain being described. For example, a class *O1:Child* can be used to represent the set of all childs. Similarly, the object property O_1*:hasBrother* can be used to represent the brotherhood relationship. Finally, the individual O_1*:Mohamed* can be used to represent a particular child called "Mohamed".
- Expressions, represent complex notions in the domain being described. For example, a *class expression* describes a set of individuals in terms of the restrictions on the individuals characteristics.
- Axioms are statements that are asserted to be true in the domain being described. For example, using a subclass axiom, one can state that the class O_1*:Boy* is a subclass of the class O_1*:Child*.

To analyze the possible changes that an OWL-2 ontology can undergo, Table 4 shows in the first column three sets of these changes: Class expression axiom set expresses the changes on ontology classes. Object property axioms and data property axioms sets express respectively the changes on the object properties and data-type properties of an OWL-2 ontology. The second column in Table 4 presents the change label. The third column shows the expression of each change jointly with a concrete example. While the last column is an interpretation of the change and the related example.

[5] https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/

Change Class	Change Type	Change Expression	Interpretation
Class Expression Axiom	SubClassOf	SubClassOf(O'_2:c'_2 O'_2:c_2)	Each c'_2 is an c_2.
		SubClassOf(*a:Baby a:Child*) SubClassOf(*a:Child a:Person*) ClassAssertion(*a:Baby a:Sara*) => SubClassOf(*a:Baby a:Person*)	Each baby is a child. Each child is a person. Sara is a baby. => Each baby is a person.
	EquivalentClasses	EquivalentClasses(O'_2:c_2) ObjectIntersectionOf(O'_2:c'_2 O'_2:c''_2)	The instances of c_2 are exactly those instances that are both an instance of c'_2 and an instance of c''_2.
		EquivalentClasses(*a:Boy ObjectIntersectionOf(a:Child a:Man)*) ClassAssertion(*a:Child a: Younes*) ClassAssertion(*a:Man a: Younes*) ClassAssertion(*a:Boy a: Mohamed*)	A boy is a male child. Younes is a child. Younes is a man. Mohamed is a boy.
	DisjointClasses	DisjointClasses(O'_2:c_2 O'_2:c'_2)	Nothing can be both an c_2 and an c'_2.
		DisjointClasses(*a:Boy a:Girl*) ClassAssertion(*a:Girl a:Sara*)	Nothing can be both a boy and a girl. Sara is a girl.
	DisjointUnion	DisjointUnion(O'_2:c_2 O'_2:c'_2 O'_2:c''_2)	Each c_2 is either an c'_2 or an c''_2, each c'_2 is an c_2, each c''_2 is an c_2, and nothing can be both an c_2 and an c''_2.
		DisjointUnion(*a:Child a:Boy a:Girl*) ClassAssertion(*a:Child a: Mohamed*) ClassAssertion(. ObjectComplementOf(*a:Girl a: Mohamed*)	Each child is either a boy or a girl, each boy is a child, each girl is a child, and nothing can be both a boy and a girl. Mohamed is a child. Mohamed is not a girl.
Object Property Axioms	SubObjectPropertyOf	SubObjectPropertyOf(O'_2:p'_2 O'_2:p_2)	Having the property p'_2 implies having the property p_2.
		SubObjectPropertyOf(*a:hasDog a:hasPet*) ObjectPropertyAssertion(*a:hasDog a:Yahia a:Brian*)	Having a dog implies having a pet. Brian is a dog of Yahia.
	EquivalentObjectProperties	EquivalentObjectProperties(O'_2:p_2 O'_2:p'_2)	Having the property p_2 is the same as having the property p'_2.
		EquivalentObjectProperties(*a:hasBrother a:hasMaleSibling*) ObjectPropertyAssertion(*a:hasBrother a:Younes a:Mohamed*) ObjectPropertyAssertion(*a:hasMaleSibling a:Mohamed a:Younes*)	Having a brother is the same as having a male sibling. Mohamed is a brother of Younes. Younes is a male sibling of Mohamed.
	DisjointObjectProperties	DisjointObjectProperties(O'_2:p_2 O'_2:p'_2)	Nothing can be both an p_2 and an p'_2.
		DisjointObjectProperties(*a:hasFather a:hasMother*) ObjectPropertyAssertion(*a:hasFather a: Mohamed a:Yahia*) ObjectPropertyAssertion(*a:hasMother a: Mohamed a:Fatima*)	Fatherhood is disjoint with motherhood. Yahia is Mohamed's father. Fatima is the mother of Mohamed.
	InverseObjectProperties	InverseObjectProperties(O'_2:p_2 O'_2:p'_2)	Having the property p_2 is the opposite of having the property p'_2.
		InverseObjectProperties(*a:hasFather a:fatherOf*) ObjectPropertyAssertion(*a:hasFather a: Mohamed a:Yahia*) ObjectPropertyAssertion(*a:fatherOf a:Yahia a:Younes*)	Having a father is the opposite of being a father of someone. Yahia is Mohamed's father. Yahia is Younes's father.
	ObjectPropertyDomain	ObjectPropertyDomain(O'_2:p_2 O'_2:c_2)	Each individual that has an outgoing p_2 connection must be an instance of c_2.
		ObjectPropertyDomain(*a:hasDog a:Person*) ObjectPropertyAssertion(*a:hasDog a:Yahia a:Brian*)	Only people can own dogs. Brian is a dog of Yahia.
	ObjectPropertyRange	ObjectPropertyRange(O'_2:p_2 O'_2:c_2)	Each individual that has an incoming p_2 connection must be an instance of O'_2:c_2.
		ObjectPropertyRange(*a:hasDog a:Dog*) ObjectPropertyAssertion(*a:hasDog a:Yahia a:Brian*)	The range of the *a:hasDog* property is the class *a:Dog*. Brian is a dog of Yahia.
	FunctionalObjectProperty	FunctionalObjectProperty(O'_2:p_2)	Each individual that has the property p_2 can point to at most one distinct individual.
		FunctionalObjectProperty(*a:hasFather*)	Each object can have at most one father.
		ObjectPropertyAssertion(*a:hasFather a:Mohamed a:Yahia*)	Yahia is Mohamed's father.
	InverseFunctionalObjectProperty	InverseFunctionalObjectProperty(O'_2:p_2)	Each individual that has the property p_2 can point to at most one distinct individual.
		InverseFunctionalObjectProperty*(a:fatherOf)*	Each object can have at most one father.
		ObjectPropertyAssertion(*a:fatherOf a:Yahia a:Mohamed*)	Yahia is Mohamed's father.
	ReflexiveObjectProperty	ReflexiveObjectProperty(O'_2:p_2)	Each individual that has the property p_2 must be connected to itself.
		ReflexiveObjectProperty(*a:knows*) ClassAssertion(*a:Person a:Yahia*) =>ObjectPropertyAssertion(*a:knows a: Yahia a: Yahia*)	Everybody knows themselves. Yahia is a person. Yahia knows himself.

Source: Author

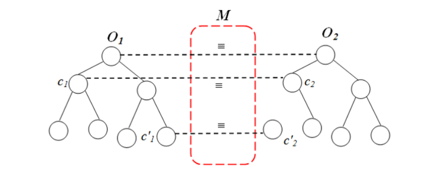
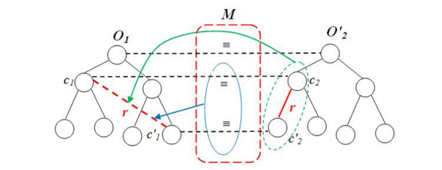
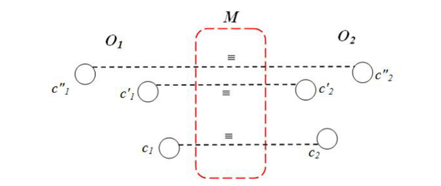
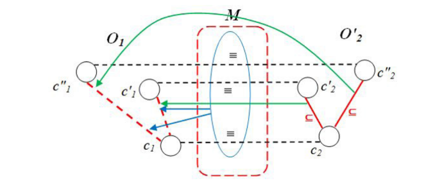
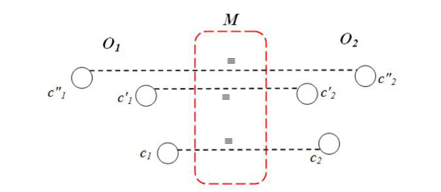
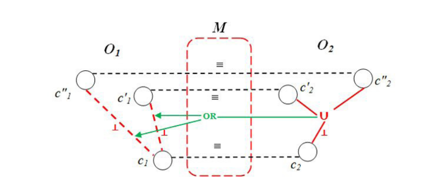
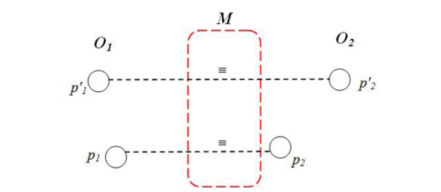
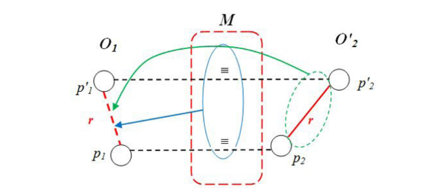
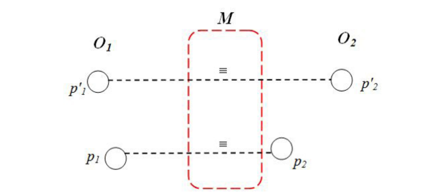
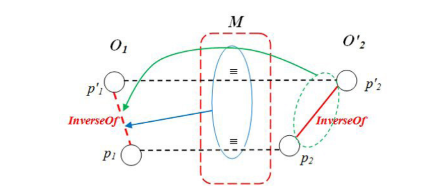
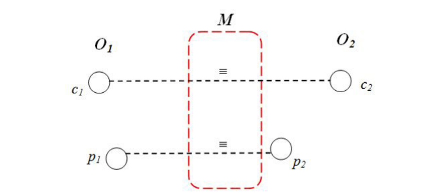
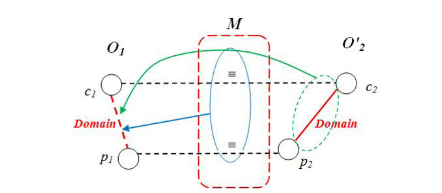
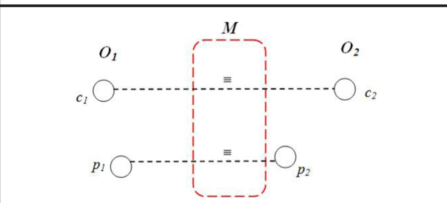
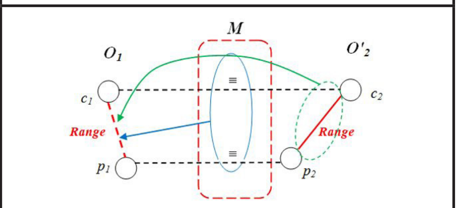
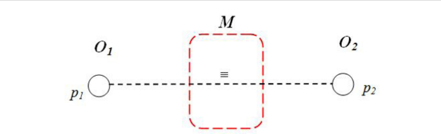
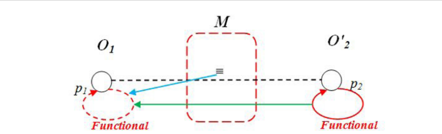
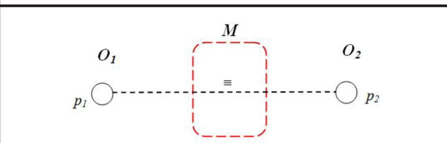
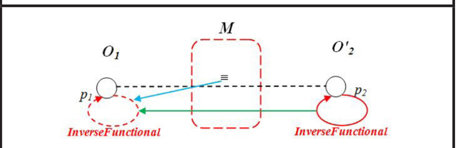
Table 4: OWL-2 Ontological changes (To be continued).

Change Class	Change Type	Change Expression	Interpretation
Object Property Axioms (Continuation)	IrreflexiveObjectProperty	IrreflexiveObjectProperty(O'_2:p_2)	Each individual that has the property p_2 cannot be connected to itself.
		IrreflexiveObjectProperty(*a:marriedTo*)	Nobody can be married to themselves.
	SymmetricObjectProperty	SymmetricObjectProperty(O'_2:p_2)	If an individual x is connected by p_2 to an individual y, then y is also connected by p_2 to x.
		SymmetricObjectProperty(*a:friend*) ObjectPropertyAssertion(*a:friend a:Yahia a:Brian*)	If x is a friend of y, then y is a friend of x. Brian is a friend of Yahia.
	AsymmetricObjectProperty	AsymmetricObjectProperty(O'_2:p_2)	If an individual x is connected by p_2 to an individual y, then y cannot be connected by p_2 to x.
		AsymmetricObjectProperty(*a:parentOf*) ObjectPropertyAssertion(*a:parentOf a: Yahia a: Mohamed*)	If x is a parent of y, then y is not a parent of x. Yahia is a parent of Mohamed.
	TransitiveObjectProperty	TransitiveObjectProperty(O'_2:p_2)	If an individual x is connected by p_2 to an individual y that is connected by p_2 to an individual z, then x is also connected by p_2 to z.
		TransitiveObjectProperty(*a:ancestorOf*) ObjectPropertyAssertion(*a:ancestorOf a:Djilali a:Yahia*) ObjectPropertyAssertion(*a:ancestorOf a:Yahia a:Sara*) =>ObjectPropertyAssertion(*a:ancestorOf a:Djilali a:Sara*)	If x is an ancestor of y and y is an ancestor of z, then x is an ancestor of z. Djilali is an ancestor of Yahia. Yahia is an ancestor of Sara. Djilali is an ancestor of Sara.
Data Property Axioms	SubDataPropertyOf	SubDataPropertyOf(O'_2:p'_2 O'_2:p_2)	If an individual x is connected by p'_2 to a literal y, then x is connected by p_2 to y as well.
		SubDataPropertyOf(*a:hasLastName a:hasName*) DataPropertyAssertion(*a:hasLastName a:Yahia "Atig"*) => DataPropertyAssertion(*a:hasName a:Yahia "Atig"*)	A last name of someone is his/her name as well. Yahia's last name is "Atig". Yahia's name is "Atig".
	EquivalentDataProperties	EquivalentDataProperties(O'_2:p_2 O'_2:p'_2)	In any expression in O'_2 containing such an axiom, p can be replaced with p' without affecting the meaning of O'_2.
		EquivalentDataProperties(*a:hasName a:Nom*) DataPropertyAssertion(*a:hasName a:Sara "Sara Atig"*) DataPropertyAssertion(*a:estNommé a:Sara "Atig Sara"*) => DataPropertyAssertion(*a:estNommé a:Sara "Sara Atig"*)	*a:hasName and a:estNommé* (in French) are synonyms. Sara's name is " Sara Atig". Sara's name is " Atig Sara".
	DisjointDataProperties	DisjointDataProperties(O'_2:p_2 O'_2:p'_2)	No individual x can be connected to a literal y by both p_2 and p'_2.
		DisjointDataProperties(*a:hasName a:hasAddress*) DataPropertyAssertion(*a:hasName a:Yahia "Yahia Atig"*) DataPropertyAssertion(*a:hasAddress a:Yahia "Saida, Algeria"*)	Someone's name must be different from his address. Yahia's name is "Yahia Atig". Yahia's address is "Saida, Algeria".
	DataPropertyDomain	DataPropertyDomain(O'_2:p_2 O'_2:c_2)	If an individual x is connected by p_2 with some literal, then x is an instance of c_2.
		DataPropertyDomain(*a:hasName a:Person*) DataPropertyAssertion(*a:hasName a:Yahia "Yahia Atig"*)	Only people can have names. Yahia's name is "Yahia Atig".
	DataPropertyRange	DataPropertyRange(O'_2:p_2 *DR*)	If some individual is connected by p_2 with a literal x, then x is in DR.
		DataPropertyRange(*a:hasName xsd:string*) DataPropertyAssertion(*a:hasName a:Yahia "Yahia Atig"*)	The range of the *a:hasName* property is *xsd:string.* Yahia's name is "Atig".
	FunctionalDataProperty	FunctionalDataProperty(O'_2:p_2)	For each individual x, there can be at most one distinct literal y such that x is connected by p_2 with y.
		FunctionalDataProperty(*a:hasAge*) DataPropertyAssertion(*a:hasAge a:Younes "17"^^xsd:integer*)	Each object can have at most one age. Younes is seventeen years old.

Source: Author

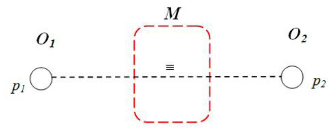
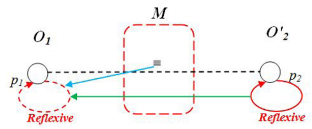
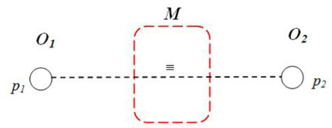
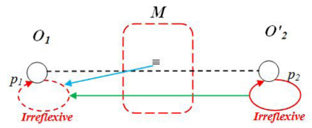
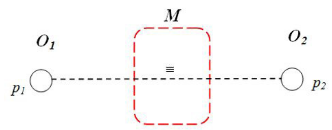
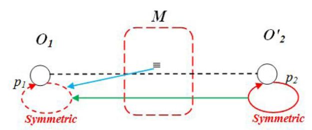
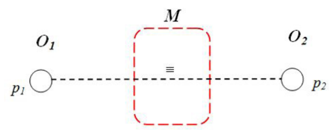
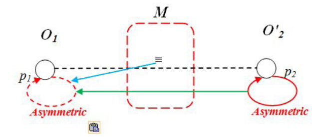
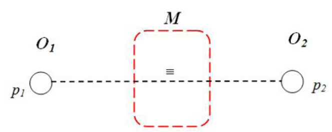
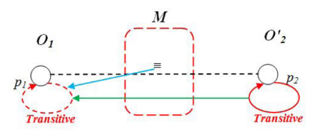
Table 4: OWL-2 Ontological changes (Continuation).

Based on the OWL-2 ontological changes and the ontology signature isomorphism (see definition 5), we propose in what follows a set of patterns to detect the violations of alignment conservativity following each change. Table 5 expresses in the first column the ontological change label. The second column shows the aligned ontology $O_1 \cup_M O_2$ before change application. While the third column shows the impact of the ontological change on the aligned ontology and the related conservativity violation detection pattern. For example, consider the first change type "SubClassOf". Upon evolving O_2 into O'_2 through the addition of the axiom $\{2{:}\ c'_2 \subseteq 2{:}\ c_2\}$, the related detection pattern requires that the image $M(2{:}\ c'_2 \subseteq 2{:}\ c_2) = \{1{:}\ c'_1 \subseteq 1{:}\ c_1\}$ be a logical consequence after reasoning on the ontology O_1 in isolation. Otherwise, this axiom is considered as a conservativity violation.

Change Type	Aligned Ontology	Conservativity Violation Detection Pattern
SubClassOf		
EquivalentClass (specific case)		
DisjointClasses		
SubObjectPropertyOf **EquivalentObjectProperties** **DisjointObjectProperties** **SubDataPropertyOf** **EquivalentDataProperties** **DisjointDataProperties**		
InverseObjectProperties		
ObjectPropertyDomain **DataPropertyDomain**		
ObjectPropertyRange **DataPropertyRange**		
FunctionalObjectProperty **FunctionalDataProperty**		
InverseFunctionalObjectProperty		

Source: Author

Table 5: Conservativity violation detection patterns (To be continued).

Change Type	Aligned Ontology	Conservativity Violation Detection Pattern
ReflexiveObjectProperty		
IrreflexiveObjectProperty		
SymmetricObjectProperty		
AsymmetricObjectProperty		
TransitiveObjectProperty		

Source: Author

Table 5: Conservativity violation detection patterns (Continuation).

Note that, equivalence relation in the alignment correspondences is the only relation considered in this work. At first glance, this may seems like a weakness for our approach. By cons, it is always possible to find a subset of the alignment with only equivalent relations in real semantic web applications. In addition, there are alignments which only accept equivalence relations within their correspondences such as UMLS[6] (https://www.nlm.nih.gov/research/umls/index.html). An equivalence relation expresses that linked entities represent the same thing in the domain of discourse. Kalfoglo and Schorlemmer (2003) consider such alignment as an isomorphism of ontological signature between two ontology vocabularies.

Remedying violations of conservativity. The process of repairing conservativity violations is the task of correcting the alignment to ensure that it fulfills the interoperability between ontologies rather than providing a new description of the domain. In the light of belief base revision theory (Hansson, 1999) and based on the work of Zahaf and Malki (2016) in the context of alignment consistency, we reformulate the alignment repair operation as a contraction operator (Hansson, 1994). Contraction is the act of selecting what to believe. It is the operation of removing a specified belief from the set of initial beliefs. Given an alignment M between two ontologies which generates a set of unwanted knowledge $\Omega = \{\delta \,/\, O_1 \cup_M O_2 \vDash \delta \text{ but } \nexists O_i,\, O_i \vDash \delta\}$, then the repair of the alignment M can be represented by the operator R defined as follows:

$R: (O_1, M, O_2) \rightarrow (O_1, M', O_2)$ which satisfies the following postulates:

Success: The postulate of success indicates that the retracted belief should not be believed after the contraction, unless it is a tautology. The contraction should be successful, that is, $M - \delta$ should not entails δ. The success postulate is formulated as follows:

$$\text{If} \nvDash \delta \text{ then } \{ O_1 \cup_M O_2 \} - \delta \nvDash \delta$$

Inclusion: The inclusion postulate implies that the contracted conservative alignment M' is included in the original alignment M, and is formulated as follows:

$$M - \delta \subseteq M$$

[6] The UMLS, or Unified Medical Language System, is a set of files and software that brings together many health and biomedical vocabularies and standards to enable interoperability between computer systems.

Uniformity: The postulate of uniformity requires that if each subset entailing a logical consequence δ also entails another logical consequence β, then the contraction by δ and β should be the same. The uniformity postulate is formulated as follows:

If it is valid for all then $M - \delta = M - \beta$ $M' \vDash \delta$ if $M' \vDash \beta$, then $M - \delta = M - \beta$.

Relevance: the postulate of relevance means that only the correspondences which are responsible for entailing the contracted logical consequence should be discarded. The relevance postulate is formulated as follows:

if $m \in M$ and $m \notin M - \delta$, then there is a subset M' of M such that, $M - \delta \subseteq M' \subseteq M$ and $M' \nvDash \delta$ but $M' \cup \{m\} \vDash \delta$

Core-retainment: the core- retainment postulate is a light version of the relevance postulate: Instead of requiring M' to be interposed between M and $M - \delta$, we are satisfied to require that it be only a sub-set of M. This version is formulated as follows:

If $m \in M$ and $m \notin M - \delta$, then there is a subset M' of M such that, $M' \nvDash \delta$ but $M' \cup \{m\} \vDash \delta$.

Zahaf and Malki (2016) define an alignment α-kernel as the minimal subset of the alignment that imply an ontological axiom *α*. We use this notion to define the Minimal Conservative Upon Revision (MCUR) as the minimal subset of an alignment that violates conservativity principle upon adding an axiom to one of the connected ontologies or adding a correspondence to the alignment. Formally,

Definition 7 (MCUR). Given an alignment M between two ontologies O_1 and O_2, a subset $M' \subseteq M$ is a Minimal Conservative Upon Revision if it satisfies the following conditions:

- M' is conservative
- for all δ is an undesirable logical consequence, M′ and M′ ∪ {c} are MUCRs (δ-kernels) respectively upon adding an axiom *α* to one of the connected ontologies O_1 or O_2, and a correspondence *n* to *A*.

We refer to α-MCURs the set of Minimal Conservative Upon Revision by adding an axiom *α* to one of the connected ontologies.

In Example 2 , the entailed axiom {1 : Student ⊆ 1 : Employee} δ is an undesirable logical consequence and the subset $(m_3\ , m_4)$ *is a unique α-MUCR upon adding the new axiom* β_5 *= {2 : Postgraduate ⊆ 2 : Employee} to the ontology* O_2.

To repair the alignment, we should prevent it to generate any undesirable knowledge. Hence the conservativity resolution problem can be reformulated to the alignment contraction problem (Zahaf and Malki, 2016). We define the alignment incision function operator as the operator that discard from *M* at least one element from each MCUR.

Definition 8 (Alignment Incision Function). An incision function *σ* for *A* is a function that for all MCUR:

1. σ(MCUR) ⊆ ⨆(MCUR) and
2. if ∅ ≠ X ∈ ∑ MCUR, then X ∩ σ(MCUR) ≠ ∅

As it is proved by Zahaf and Malki (2016), the alignment incision function is characterized by the success, the inclusion, the uniformity, and the core-retainment postulates. Concerning the problem of conservativity resolution, the success postulate corresponds to the conservativity principle which means the undesirable knowledge is successfully removed. The inclusion ensures no new correspondences should be added to alignment when realizing the resolution. The postulate of core-retainment expresses the principle of minimal change which means only correspondences that are somehow responsible for entailing the undesirable knowledge should be discarded. As to the uniformity that expresses determinism in resolution. There is no reason to solve the problem differently for logically related unwanted knowledge. The following representation theorem summarizes these postulates for every alignment incision function operator.

Theorem 1 (Alignment Incision Function Representation). The operator − is an alignment incision function for an alignment *M* if and only if it satisfies the following postulates:

[Success] if $\nvDash \delta$ then $\{O_1 \cup_M O_2\} - \delta \nvDash \delta$
[Inclusion] $M - \delta \subseteq M$

[Core−retainment] if $c \in M$ et $c \notin M - \delta$, then there is a subset M′ of M such that, $M' \nvDash \delta$ but $M' \cup \{c\} \vDash \delta$.

Uniformity] if it holds for all $M' \subseteq M$ that $M' \vDash \delta$ if and only if $M' \vDash \beta$, then $M - \delta = M - \beta$.

To resolve the problem of conservativity violations, we adapt a set of algorithms proposed in Zahaf (2017) for the resolution of the alignment inconsistency problem. Table 6 presents an algorithm to find an MCUR. It compute a minimal subset of correspondences that is responsible of alignment conservativity violations. The proposed algorithm consists in testing if *M* still implies the undesired axiom δ after removing each of its correspondences. If this is not the case the removed correspondence is reintroduced in *M*. The final result is an MCUR which is a minimal set M′ ⊆ M that do imply δ. Algorithm 1 can compute an MCUR in polynomial time in the size of the aligned ontology.

```
Algorithm 1: MCUR
MCUR (M,o1,o2,δ)
Input : o1,o2 // two ontologies
        M // M is an alignment between o1 and o2
        δ// δ is an undesired axiom
Output : M // an MCUR
for c ∈ M
    do
        if M \{c}⊨ δ
            then M ← M \{c}
return M
```

Source: Authors

Table 6: MCUR algorithm.

Example 4: Following example 2, Let be Postgraduate ⊆ Researcher the new axiom α added to O_2*, and therefore, the axiom Student ⊆ Employee will be an undesirable logical consequence δ in ontology* O_1*. Algorithm 1 return M = {1:Researcher* $=_{0.62}$ *2:Researcher; 1: Student* $=_{0.62}$ *2: Postgraduate} which is an MCUR.*

α-MCUR and Incision Functions. In order to compute the α-MCUR and the corresponding incision functions, we also adapt the alignment kernel algorithm and it incision functions proposed in by Zahaf (2017) to deal with the alignment consistency problem, itself inspired from the Hitting set algorithm proposed by Reiter (1987) to diagnose systems. Given a collection of sets F, a Hitting set is a set that intersects each set of the collection. Hitting set algorithm builds a Tree for a collection of sets F such that, its root is labeled by √ if F is empty. Otherwise, it is labeled by an arbitrary set of F. If *n* is a node of the tree, define H(n) to be the set of edge labels on the path from the root to the node *n*. If *n* is labeled by √, it has no successor nodes in the tree. If *n* is labeled by a set ∑ of F, then for each σ∈∑, *n* has a successor node n_σ joined to *n* by an edge labeled by *σ*. The label for n_σ is a set S∈F such that S∩H(n)=∅, if such a set S exists. Otherwise, nσ is labeled by √.

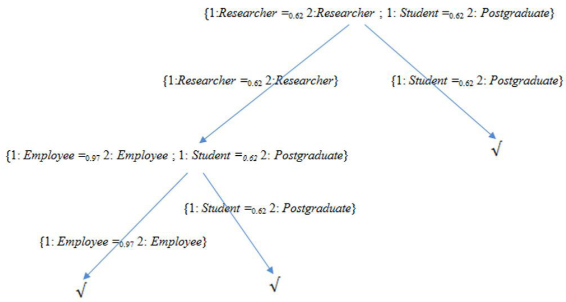
The α-MCUR is the collection of all MCUR upon adding the new axiom *α*. As defined above (see definition 8), the incision function intersects each MCUR of the collection α-MCUR. Consequently, we can consider the incision function as a Hitting set of the α-MCUR. The nodes of the tree are labeled by MCURs and edges are labeled by the elements of these MCURs. However, the α-MCUR is not given explicitly and we should compute it using the output of algorithm 1 which is the first computed MCUR. At each node, an MCUR of the set $M \setminus$H(n) is computed if such an MCUR exists. Otherwise, H(n) is an incision function. Unfortunately, the Hitting set algorithm has an exponential time (Rymon, 1991). Table 7 outlines this algorithm. The example 5 as well as Figure 3 illustrate the progress of the algorithm.

```
Algorithm 2: α-MCUR and Incision functions
α-MCURandIncisionFct (M,o1,o2,δ)
Input : o1,o2 // two ontologies
        M // M is an alignment between o1 and o2
        δ// δis an undesired axiom
Output : α-MCUR // an collection of α-MCURs
        Incision // set of incision functions
Incision ← ∅
Stack ← Empty
C ← MCUR (M,o1,o2,δ)
α-MCUR ← {C}
for c ∈ C
    do insert {c} in the top of the stack
While Stack not Empty
    do Hn ← last element of the stack
    remove last element of the stack
    If M\{Hn} ⊨ δ
        Then C ← MCUR (M\{Hn},o1,o2,δ)
        α-MCUR ← α-MCUR ∪ {C}
        for c ∈ C
            do insert Hn∪{c} in the top of the stack
    Else Incision ← Incision ∪{Hn}
End.
```

Source: Authors

Table 7: α-MCUR and Incision functions algorithm.

Example 5. Following the example 4, we apply Algorithm 2 to compute incision functions. We obtain Incision = {{1: Student $=_{0.62}$ *2:Postgraduate},{1:Researcher* $=_{0.62}$ *2:Researcher; 1: Student* $=_{0.62}$ *2: Postgraduate}, {1:Researcher* $=_{0.62}$ *2:Researcher ; 1: Employee* $=_{0.97}$ *2: Employee}}*



Source: Authors

Figure 3: Hitting set tree of α-MCUR incision functions.

Related works. We consider the Consistency Preservation and the Ontology Change Preservation problems as two instances of the conservativity problem (Atig et al., 2022). Therefore, we examine also literature concerning these two problems.

Violations detection of conservativity principle was subject of study by Jiménez-Ruiz et al. (2011) and Solimando, Jiménez-Ruiz and Guerrini (2016). According to iménez-Ruiz et al. (2011), the violations appeared when the set of inferences after reasoning on an ontology is different if used in conjunction with the alignment. Solimando, Jiménez-Ruiz and Guerrini (2016) defines two types of conservativity violations: subsumption and equivalence. To avoid subsumption violations, the aligned ontology must not introduce new subsumption relationships between concepts within the input ontologies, while equivalence violation is defined as two subsumption violations in both directions. Both approaches rely on confidence values to eliminate the correspondence with smallest value in each conflict set. They propose the locality principle[7] to compute confidence values for conflict correspondences if these values are missing in the alignments to be repaired. For detecting ontology change preservation violation, the study by Zahaf (2012) is based on the signature of the propagated axiom in the versions of the same ontology. Resulting correspondences are ordered using the comparison of their intentional persistence degrees to penalize the weakest.

Consistency principle was the object of study in several works. For instance, Lily (Tang et al., 2018) invokes user decision to repair two types of inconsistencies: mappings that form a circle and mappings that do not meet the equivalentClass/disjointWith axioms mentioned in the original ontology. YAM++ (Bellahsene et al., 2017) is based on ALCOMO (Meilicke, 2011) to detect alignment consistency violations via disjoint-subsumption patterns. The reparation process defines two type of diagnosis: Global optimal diagnosis which removes the slightest amount of confidence, and Local optimal diagnosis, which an incremental check of the correspondences set. ASMOV (Jean-Mary, Shironoshita, and Kabuka, 2009) introduces the notion of mapping validation in a graph constitutes nodes (pairs of entities) and edges (pairs of properties). The iterative validation process is done in three phases: concept validation, property validation and concept-property validation. 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. The core of LogMap (Jiménez-Ruiz, 2019) is an iterative process that alternates mapping discovery (using unsatisfiability detection) and mapping repair (with minimum confidence) steps to deal with consistency violations. Most of repair process in the studied approaches are based on computing and discarding diagnosis from alignment. A diagnosis is the set of correspondences that have the lowest confidences values in each conflict set. We follow the same approach in dealing with conservativity principle violations.

Results and discussion

To demonstrate the methods proposed in this research, we have created a Java application based on OWL API (Horridge and Bechhofer, 2009) and Align API (Euzenat, 2004) to manipulate OWL ontologies and alignments between them. This application also integrates pellet (Meilicke, 2011) as the main reasoning engine on OWL ontologies.

Numerous ways exist to assess our suggested approach. For instance, we can gauge its performances based on the minimal change principle, ensuring the least possible information loss during the adaptation process. Another way is to compare these performances against other alignment adaptation methods. But, it seems a priority to investigate whether our proposal deals well with the identified alignment conservative adaptation problem. For this purpose, the goal of this experimentation is to show the limits of alignment evolution w.r.t the conservativity principle upon ontology change.

Data Set. In the current evaluation approach, we need at least two ontologies together with an alignment between them, in addition to the ontological changes either in the form of explicit change logs or in the form of a version of the modified ontology. In this section, we present the used ontologies together with the ontological changes and the alignments.

a. **Agronomy Ontology (AgrO).** Is an ontology for representing agronomic practices, techniques, variables and related entities. According to its founder, the Consultative Group for International Agricultural Research[8] (CGIAR), a global research partnership for a food-secure future,

[7] 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 (Jiménez-Ruiz and and Cuenca Grau 2011).

[8] https://www.cgiar.org/

the Agronomy Ontology[9] provides terms from the agronomy domain that are semantically organized and can facilitate the collection, storage and use of agronomic data, enabling easy interpretation and reuse of the data by humans and machines alike. AgrO is of significant size containing in its latest version (uploaded on 11-02-2022 in the AgroPortal repository[10]) 4163 classes, 209 properties and 552 individuals. In this experiment, we use two versions that are sufficiently different (1^{st} and 5^{th}) to generate the set of ontological changes between versions. The first version is the 06-06-2016 version (AgrO-1) while the second one is the 03-04-2022 version (AgrO-5).

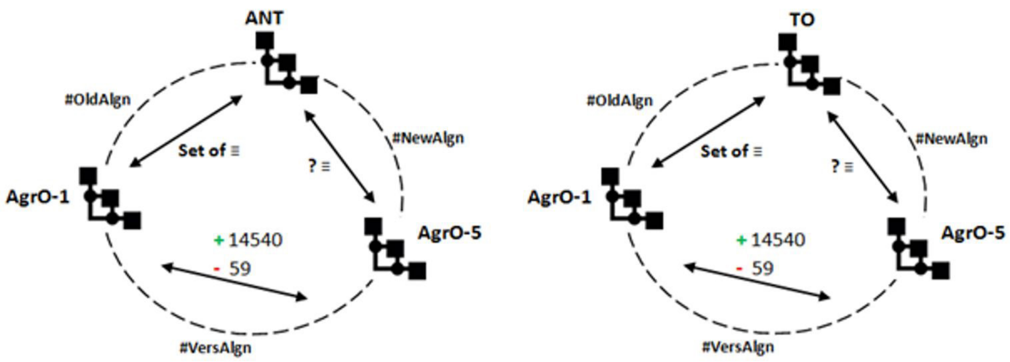
b. **Plant Trait Ontology (TO).** For its owner the Wheat Data Interoperability Working Group[11] (WG), the Plant Trait Ontology is an ontology for describing phenotypic traits in plants. Each trait is a distinguishable feature, characteristic, quality or phenotypic feature of a developing or mature plant. Although the first version of TO was launched in 2016, its most recent version dates from 2022. This version has 5,262 classes in addition to 159 properties and no individuals. This ontology is considered here to be a second ontology allowing to construct an original alignment.

c. **Agricultural and Nutrition Technology Ontology (ANT).** Is an ontology created by the International Food Policy Research Institute[12] (IFPRI), which provides research-based policy solutions to sustainably reduce poverty and end hunger and malnutrition in developing countries. ANT is a small ontology with 127 classes, 4 properties and only 4 individuals. This makes it easy to use as a dataset in this experiment. ANT is released on 07-28-2013 and the last known version is uploaded in the AgroPortal repository on 10-16-2017. Same as TO, ANT is used here to be a second ontology for constructing an original alignment.

To generate the ontological change, we have used the method developed in study by 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. For the used dataset, Figure 4 shows the number of added (+) and deleted matchings (-) between the pair of ontology versions. It also presents the alignment considered as original (#OldAlgn) between the AgrO-1 and the second ontology TO/ANT, also the alignment considered as new (#NewAlgn) between the TO/ANT and the new ontology version AgrO-5.

d. **Ontological changes.** The difference between the two versions of AgrO ontologies is calculated in terms of the number of added and deleted axioms. In total, 14540 axioms have been added and 59 axioms have been deleted. Table 8 shows this difference in detail. We have for each type of axiom (see definition 1) the count calculated for the two versions.

[9] https://bigdata.cgiar.org/resources/agronomy-ontology/
[10] https://agroportal.lirmm.fr/ontologies/AGRO/?p=summary
[11] https://ist.blogs.inrae.fr/wdi/
[12] https://www.ifpri.org/



Source: Authors

Figure 4: Dataset.

Axiom type		Agro V1	Agro V5
Class Axioms	SubClassOf	611	3967
	EquivalentClasses	56	386
	DisjointClasses	11	36
	DisjointUnion	55	376
Object Property Axioms	SubObjectPropertyOf	29	177
	EquivalentObjectProperties	0	22
	InverseObjectProperties	14	44
	DisjointObjectProperties	0	0
	FunctionalObjectProperty	2	3
	InverseFunctionalObjectProperty	0	0
	TransitiveObjectProperty	2	18
	SymmetricObjectProperty	0	2
	AsymmetricObjectProperty	0	0
	ReflexiveObjectProperty	0	1
	IrreflexiveObjectProperty	0	1
	ObjectPropertyDomain	19	55
	ObjectPropertyRange	21	57
	SubPropertyChainOf	6	69
Data Property Axioms	SubDataPropertyOf	0	2
	EquivalentDataProperties	0	0
	DisjointDataProperties	0	0
	FunctionalDataProperty	1	1
	DataPropertyDomain	2	3
	DataPropertyRange	2	2
Individual Axioms	ClassAssertion	179	360
	ObjectPropertyAssertion	27	43
	DataPropertyAssertion	2	0
	NegativeObjectPropertyAssertion	0	0
	NegativeDataPropertyAssertion	0	0
	SameIndividual	0	0
	DifferentIndividuals	2	2
Annotation Axioms	AnnotationAssertion	1058	9106
	AnnotationPropertyDomain	0	0
	AnnotationPropertyRange	0	0

Source: Authors

Table 8: Number of axioms in the AgrO versions.

e. Alignments. Concerning the alignments to be repaired, the old alignments are considered to be the one between TO and AgrO-1 and also between ANT and AgrO-1, while the set of matchings between TO/ANT and the version AgrO-5 make the evolved alignments after change. Figure 4 schematizes this situation. To calculate the original alignments, we are based on the results available on the AgroPortal platform[13]. For almost each of the ontologies available in this portal, a set of alignments with the other ontologies of the portal are available in the "Mappings" section.

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 measure the effectiveness of our method in the alignment adaptation context, we use the number of conservativity violations by changed axioms. In addition, we calculate the elapsed time, as well as the rate of violations reparation compared to the original alignments. The violations reparation rate of an alignment M is defined by %Rep=($\Delta/\mathcal{M}$)*100%. where Δ is a diagnosis of initial alignment *M*.

[13] https://agroportal.lirmm.fr/mappings

Experimentation. The experimentation process was conducted in two steps. In the first one, we exploit the change log between the original ontology AgrO-1 and the new version AgrO-5 to detect the set of conservativity violations for the original alignments upon input ontology evolution. Then, in the second step, we use our method to adapt these alignments following the ontological changes applied to the new AgrO-5 version.

a. Detection Process. To detect conservativity violations upon ontology evolution, we use the change log between AgrO-1 and the new version AgrO-5. This log contains 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. Then, we count the number of conservativity violations caused by the related ontological changes. Table 9 shows the detailed results for both tests in this experiment. The first line designates the test named by its related ontologies, while the second and third lines show respectively the number of correspondences and conservativity violations in the old alignments.

b. Reparation Process. This step aims to show the impact of our proposed method to avoid alignment conservativity violations upon ontology change. The fourth line in Table 9 presents the number of correspondences in every diagnosis. The fifth line shows the size of new alignments considered as conservative upon ontology change.

	AgrO-1↔TO↔ AgrO-5	AgrO-1↔ANT↔ AgrO-5
#OldAlgn	596	10
#Viol	2	1
#Diagnosis	2	1
#NewAlgn	594	9
#Time ns	10.5	0.59
%Rep	0.33	10

Source: Authors

Table 9: Results in the context of alignment adaptation problem.

Comment 1. Despite the relatively high number of matchings in the #OldAlgn (596) in the first test (AgrO-1↔TO↔ AgrO-5), the number of violations detected is very small (2). This is due to the quality of the matchings in this test. 98% of the correspondences are of type (SAME_URI) and are not considered by the detection process. 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. 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.

Traditional assessment methods are mainly designed to rely on benchmarks to compare precision and recall of results. However, in the alignment adaptation context, we haven't these benchmarks. 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 test, at what degree our proposed method reuses the original alignment while respecting the conservativity principle upon ontology change. The two last lines in Table 9 show the results of these measures. The sixth line shows the elapsed time measured in nanosecond to repair old alignments while the seventh line shows the repair rate compared to old alignments size.

Comment 2. Although the repair time is very encouraging for potential automatic use (with normal computational performance), the repair rate is crucial in this experiment. We observe a very tiny rate in the first test (0.33%), while a medium rate is noted in the second test (10%). The latter can be justified by the reduced amount of correspondences in the original alignment (10). 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 the strategy of adapting alignments following the evolution of their input ontologies makes it possible to exploit to the maximum the efforts provided the first time in constructing alignments.

Conclusion

The current work is a continuation of previous works on the alignment adaptation problem in the case of evolution of one of its input ontologies (Atig et al. 2013; Atig, Zahaf, and Bouchiha, 2016; Atig et al., 2022 and Atig, 2022). After identifying the problem via an example, we have confined a set of the possible ontological changes that an OWL-2 ontology can undergo. Then we propose an adequate pattern for each change in order to detect the conservativity violations. Inspired by diagnostics theory, we have adapted two algorithms from the literature designed to the alignment consistency problem to the alignment conservativity problem. This adaptation has served as a satisfactory repair process to deal with conservativity violations. Indeed, the experimentation has shown the applicability of our approach to help improve the quality of alignments between ontologies. However, other research questions are still pending. For instance, in the near future we need to study the repairs impact on the alignment interoperability, as well as improving the techniques to repair correspondences instead of eliminating them. Another promising issue is the study of this problem for other languages outside OWL-2. As general conclusion, we note that alignment adaptation problem has not received the necessary importance given the lack of works provided until this day.

Corresponding author:
Yahia Atig
Institute of Science and Technology, Department of Mathematics and Computer Science
University Center of Nâama. Algeria
Phone: (+213)48379368, E-mail: y_atig@esi.dz

References

[1] Atig, Y. (2022) "Découverte et Réparation des Alignements d'Ontologies dans le Web de Données Liées", Ph.D. thesis. University of Sidi Bel Abbès. Algeria. [Online]. Available: http://rdoc.univ-sba.dz/bitstream/123456789/3788/1/DS_Info_ATIG_Yahia.pdf [Accessed: May 12, 2024].

[2] Atig, Y., Amine, A., Zahaf, A. and Kumar, A. (2013) "Alignment Evolution between Ontologies Using a Change Log", *International Journal Of Data Mining And Emerging Technologies*, Vol. 3, No. 2, pp. 81-87. E. ISSN 2249-3220. DOI 10.5958/j.2249-3220.3.2.011.

[3] Atig, Y., Zahaf, A. and Bouchiha, D. (2016) "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. E-ISSN 2074-9015, ISSN 2074-9007. DOI 10.5815/ijitcs.2016.07.09.

[4] Atig, Y., Zahaf, A., Bouchiha, D. and Malki, M. (2022) "Alignment Conservativity under the Ontology Change", *Journal of Information Technology Research*, Vol. 15, pp. 1-19. E-ISSN 1938-7865, ISSN 1938-7857. DOI 10.4018/JITR.299923.

[5] Bellahsene, B., Emonet, V., Ngo, D. and Todorov, K. (2017) "YAM++ Online: A Web Platform for Ontology and Thesaurus Matching and Mapping Validation", *ESWC: European Semantic Web Conference*, Portroz, Slovenia, pp.137-142. DOI 10.1007/978-3-319-70407-4_26. hal-01987659.

[6] Borgida, A. and Serafini, L. (2003) "Distributed description logics: Assimilating information from peer sources", *Journal of Data Semantics*, Vol. 1, pp. 153–184. E-ISSN 1861-2040. DOI:10.1007/978-3-540-39733-5_7.

[7] Euzenat, J. (2004) "An API for ontology alignment", *Proceedings of the 3rd conference on international semantic web conference (ISWC)*, Hiroshima, Japan, pp. 698-712. ISBN 978-3-540-23798-3. DOI 10.1007/978-3-540-30475-3_48.

[8] Euzenat, J. and Shvaiko, P. (2007) "*Ontology Matching*", Springer Verlag. 333 p. ISBN 978-3-540-49611-3. DOI 10.1007/978-3-540-49612-0.

[9] Euzenat, J. and Shvaiko, P. (2013) "*Ontology matching*", Springer. Heidelberg. 511 p. E-ISBN 978-3-642-38721-0. ISBN 978-3-642-38720-3. DOI 10.1007/978-3-642-38721-0.

[10] Hansson, S. O. (1994) "Kernel contraction", *The Journal of Symbolic Logic*, Vol. 59, No. 3, pp. 845-859. E-ISSN 1943-5886. ISSN 0022-4812. DOI 10.2307/2275912.

[11] Hansson, S. O. (1999) "A Textbook of Belief Dynamics. Theory Change and Database Updating, Dordrecht: Kluwer. ISBN 978-0-792-35327-0. DOI 10.1007/978-94-007-0814-3.

[12] Horridge, M. and Bechhofer, S. (2009) "The OWL API: A Java API for Working with OWL 2 Ontologies", *Proceedings of the 5th International Workshop on OWL: Experiences and Directions,* Chantilly, Virginia. DOI:10.3233/SW-2011-0025.

[13] Jean-Mary, Y. R., Shironoshita, E. P. and Kabuka, M. R. (2009) "Ontology Matching With Semantic Verification", *Journal of Web Semantics*, Vol. 7, No. 3, pp. 235-251. ISSN 1570-8268. DOI 10.1016/j.websem.2009.04.001.

[14] Jiménez-Ruiz, E. and Cuenca Grau, B. (2011) "LogMap: Logic-based and Scalable Ontology Matching", In: Aroyo, L. et al. *The Semantic Web – ISWC 2011. ISWC 2011*, Lecture Notes in Computer Science, Vol. 7031. Springer, Berlin, Heidelberg, pp. 273-288. ISBN 978-3-642-25072-9. DOI 10.1007/978-3-642-25073-6_18.

[15] Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I. and Berlanga, R. (2011) "Logic-based Assessment of the Compatibility of UMLS Ontology Sources", *Journal of Biomedical Semantics*, Vol. 2, Suppl. 1, S2. E. ISSN 2041-1480. DOI 10.1186/2041-1480-2-S1-S2.

[16] Jiménez-Ruiz, E. (2019) "LogMap Family Participation in the OAEI 2019", *Proceedings of the 14th International Workshop on Ontology Matching*, CEUR-WS.org. [Online]. Available: https://hal.science/hal-02984947 [Accessed: April 20, 2024].

[17] Kalfoglou, Y. and Schorlemmer, M. (2003) "Ontology mapping: the state of the art", *The Knowledge Engineering Review,* Vol. 18, No. 1, pp. 1-31. E-ISSN 1469-8005, ISSN 0269-8889. DOI 10.1017/S0269888903000651.

[18] Meilicke, C. and Stuckenschmidt, H. (2009) "An efficient method for computing alignment diagnoses", In: Polleres, A., Swift, T. (eds) *Web Reasoning and Rule Systems. RR 2009.* Lecture Notes in Computer Science, Vol 5837, pp. 182-196. Springer Berlin Heidelberg. ISBN 978-3-319-11112-4. DOI 10.1007/978-3-642-05082-4_13.

[19] Meilicke, C. (2011) "Alignments Incoherency in Ontology Matching", Ph.D. thesis, University of Mannheim. [Online]. Available: https://madoc.bib.uni-mannheim.de/29351 [Accessed: Febr. 5, 2024].

[20] Reiter, R. (1987) "A theory of diagnosis from first principles", *Artificial Intelligence*, Vol. 32, No. 1, pp. 57-95. ISSN 0004-3702. DOI 10.1016/0004-3702(87)90062-2.

[21] Rymon, R. (1991) "A Final Determination of the Complexity of Current Formulations of Model-Based Diagnosis (Or Maybe Not Final?), Technical Report No. MS-CIS-91-13, University of Pennsylvania. [Online]. Available: https://core.ac.uk/download/76360613.pdf [Accessed: Febr. 5, 2024].

[22] Solimando, A., Jiménez-Ruiz E. and Guerrini G. (2016) "Minimizing Conservativity Violations in Ontology Alignments: Algorithms and Evaluation", *Knowledge and Information Systems*, Vol. 51, pp. 775-819. ISSN 0219-3116. DOI 10.1007/s10115-016-0983-3.

[23] Stojanovic, L. (2004) "Methods and tools for ontology evolution", PhD thesis, University of Karlsruhe. [Online]. Available: https://www.researchgate.net/profile/Ljiljana-Stojanovic-2/publication/35658911_Methods_and_tools_for_ontology_evolution/links/556708f508aec226830090b2/Methods-and-tools-for-ontology-evolution.pdf [Accessed: Febr. 5, 2024].

[24] Tang, Y., Wang, P., Pan, Z. and Liu, H. (2018) "Lily results for OAEI 2018", In: *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018*, Monterey, CA, USA. pp. 179-186. [Online]. Available: https://ceur-ws.org/Vol-2788/oaei20_paper9.pdf [Accessed: Febr. 15, 2024].

[25] Zahaf, A. (2012) "Alignment between versions of the same ontology", *ICWIT*, pp. 318-323. [Online]. Available: https://ceur-ws.org/Vol-867/Paper36.pdf [Accessed: Febr. 15, 2024].

[26] Zahaf, A. and Malki, M. (2016) "Kernel Contraction and Consolidation of Alignment under Ontology Change", *Journal of Information Technology and Computer Science (IJITCS)*, Vol. 8, No. 8, pp. 31-42. E-ISSN 2074-9015, ISSN 2074-9007. DOI 10.5815/ijitcs.2016.08.04.

[27] Zahaf, A. (2017) "Alignment Evolution under Ontology Change: A formal Framework and Tools", Ph.D. thesis, University of Sidi Bel Abbès. Algeria.