The Knowledge Engineering Review (2022), **37**, e10, pp. 1–27
doi:10.1017/S0269888922000066

The Knowledge Engineering Review

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

OWL ontology evolution: understanding and unifying the complex changes

Viviane Torres da Silva[1], Jéssica Soares dos Santos[2], Raphael Thiago[1], Elton Soares[1] and Leonardo Guerreiro Azevedo[1]

[1]IBM Research Brazil, Rio de Janeiro, Brazil; E-mails: vivianet@br.ibm.com, raphaelt@br.ibm.com, eltons@ibm.com, lga@br.ibm.com;
[2]Instituto de Computação, Universidade Federal Fluminense, Niterói, RJ, Brazil; E-mail: s.jessicasoares@gmail.com

Received: 25 August 2021; **Revised:** 11 October 2022; **Accepted:** 17 October 2022

Abstract

Knowledge-based systems and their ontologies evolve due to different reasons. Ontology evolution is the adaptation of an ontology and the propagation of these changes to dependent artifacts such as queries and other ontologies. Besides identifying basic/simple changes, it is imperative to identify complex changes between two versions of the same ontology to make this adaptation possible. There are many definitions of complex changes applied to ontologies in the literature. However, their specifications across works vary both in formalization and textual description. Some works also use different terminologies to refer to a change, while others use the same vocabulary to refer to distinct changes. Therefore, there is a lack of a unified list of complex changes. The main goals of this paper are: (i) present the primary documents that identify complex changes; (ii) provide critical analyses about the set of the complex changes proposed in the literature and the documents mentioning them; (iii) provide a unified list of complex changes mapping different sets of complex changes proposed by several authors; (iv) present a classification for those complex changes; and (v) describe some open directions of the area. The mappings between the complex changes provide a mechanism to relate and compare different proposals. The unified list is thus a reference for the complex changes published in the literature. It may assist the development of tools to identify changes between two versions of the same ontology and enable the adaptation of artifacts that depend on the evolved ontology.

1. Introduction

Knowledge-based systems are subject to continual changes. Theire evolution can be influenced by many factors (Stojanovic, 2004; Floris *et al.*, 2008): (i) changes in the environment or a better understanding of the domain (adaptive maintenance); (ii) changes in the application requirements (evolutive maintenance); (iii) corrections on the misbehavior of the system (corrective maintenance); and (iv) improvements to avoid future problems (preventive maintenance). As part of system evolution, the evolution of the ontologies applied to such systems is also required.

When an ontology is modified and, thus, another version is created, many other tasks have to be performed (Stojanovic *et al.*, 2002; Floris *et al.*, 2008), such as: (i) concept instances need to be changed to preserve consistency; (ii) applications have to be refactored, for example, data structures created based on the ontology should be refactored, and queries have to be inspected and modified to accommodate the changes of the ontology; and, (iii) dependent ontologies must be revised and eventually synchronized with the modified one; and (iv) related documentation must be updated to document the evolution.

One of the main challenges of knowledge-based systems is to keep dependent artifacts consistent with the evolving ontology. Manually finding and fixing all side effects of ontology changes is cumbersome,

Cite this article: V. Torres da Silva, J. S. dos Santos, R. Thiago, E. Soares and L. Guerreiro Azevedo. OWL ontology evolution: understanding and unifying the complex changes. *The Knowledge Engineering Review* 37(e10): 1–27. https://doi.org/10.1017/S0269888922000066

© The Author(s), 2022. Published by Cambridge University Press.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

error-prone, time-consuming, and costly. There are several proposals in the literature to overcome these problems, such as (i) identification and classification of ontology changes (Klein, 2004; Papavasileiou *et al.*, 2013; Rahnama & Barforoush, 2015), (ii) tools to track the changes while users are updating the ontology (Najla *et al.*, 2009), or (iii) tools to compare two versions of the same ontology and provide a list of detected changes (Stojanovic *et al.*, 2002; Stojanovic, 2004; Hartung *et al.*, 2013; Rahnama & Barforoush, 2015).

This paper concerns identifying and classifying changes that tackle the delta between two versions of the same ontology. There are two main types of changes: basic changes and complex changes. Basic changes (Klein & Noy, 2003; Stojanovic, 2004; Klein, 2004), also called low level changes (Hartung *et al.*, 2012), atomic changes (Khattak *et al.*, 2008; Dos Reis *et al.*, 2014) or elementary changes (Stojanovic *et al.*, 2002; Najla *et al.*, 2009), are atomic, fine-grained changes (Stojanovic *et al.*, 2002). These changes are typically *adds* or *deletes* applied to classes, properties, annotations and axioms. Complex changes (Klein, 2004; Dos Reis *et al.*, 2014), also called composite changes (Palma *et al.*, 2009; Khattak *et al.*, 2013) or high-level changes (Papavasileiou *et al.*, 2013; Zablith *et al.*, 2015), are operations that are composed of multiple basic operations that incorporate some additional knowledge about the changes (Klein, 2004). Operations like *class rename*, *pull up class*[1] and *move subtree*[2] are complex changes that, besides being decomposed into several basic changes, also incorporate additional knowledge. For instance, *move subtree* is a complex change since it represents the combination of basic changes related to the moving of each class in the subtree and indicates that several hierarchically related classes have been moved together.

There are in the literature several approaches mentioning complex changes, but there is no standard list of them (Flouris *et al.* 2008; Zablith *et al.*, 2015). Each approach presents its own different set of complex changes. Therefore, this work proposes a unified classification for complex changes based on the several works we have found in the literature. To create a unified list of complex changes, we have investigated the literature and mapped them toward a unification of the terminologies used to refer to the complex changes. In addition, we have also proposed to classify the complex changes according to their purpose.

The unification of those changes is fundamental since some approaches use a different set of terminologies for the same set of changes (e.g., *extract domain* Klein, 2004 and *domain enlarged* Stuckenschmidt & Van Harmelen, 2005) and, others use the same terminology for changes with a different meaning (e.g., *move property* in Stojanovic *et al.*, 2002 has a different meaning in Kondylakis, 2010). In addition, some approaches only provide a list of changes without formalism or even a textual explanation of the changes' semantics. From a set of 34 documents (including books, papers, and theses) resulting from a deep literature review (detailed in Section 2), only 23% of the documents formalize the changes they mention. Therefore, it is challenging to devise the true meaning of a change and, thus, to compare changes extracted from different works.

In total, we have found 112 complex changes in 34 documents. Many complex changes are mentioned in only one document, and only a few are mentioned in several documents. For instance, around 43% of changes applied to classes are mentioned in only one document, and only the two most mentioned complex changes applied to classes, *split* and *merge*, are mentioned in more than 82% of the documents. Due to the number of mentioned complex changes spread over the documents, it is difficult to assess how relevant a given complex change is and how representative a document that describes the changes is for the literature.

Identifying complex changes that appear between two versions of the same ontology is fundamental for adapting related artifacts. These include transforming instances and queries created based upon the classes, relationships, and axioms described in the ontology. For example, without being able to detect when a *class rename* occurs, a tool would recognize that 'a class was deleted' and 'a new class was created,' and we would lose all instances of the 'old class'. Without being able to detect when a *pull up*

[1]Pull up class: Move a class to a higher position in the subsumption hierarchy (Papavasileiou *et al.*, 2013).
[2]Move subtree: Move a concept (and subtree) from parent A to parent B (Dos Reis *et al.*, 2014).

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

class occurs, we would need to review all queries involving such class, even those mentioning properties of superclasses that remain superclasses of the moved class.

The main contributions of this paper are:

- *Unified list of complex changes.* It is not our goal to provide an exhaustive list of all possible complex changes. The number of complex changes is potentially infinite (Klein & Noy, 2003; Plessers, 2006; Flouris *et al.*, 2008). We aim to provide a unified list of those complex changes mentioned in the literature that tackles the delta between two versions of the same ontology. We have analyzed several documents that mention complex changes, and we propose a mapping between all mentioned changes to provide a unified list of complex changes. Such a unified list is a *dictionary and a thesaurus* of the complex changes found in the literature. It is a dictionary because it provides the available descriptions and formalization for most complex changes (following their availability in the literature). And it is a thesaurus because we have mapped the complex changes and found their synonyms, that is, complex changes with different terminology but with the same semantics. In addition, we have also described hypernyms and hyponyms relations between the complex changes.
- *Complex changes classification.* The approaches in the literature classify the complex changes according to the entity they apply. Also, we provide two other classifications: one classifies the complex changes according to their intentions, and the other identifies hierarchical relations between them. The hierarchical classification guides the development of heuristics to identify the complex changes since one heuristic may be defined as an extension of another. The intention classification helps in understanding that different complex changes share the same abstract semantics.
- *Analysis of the set of complex changes.* Together with the list of complex changes, we provide a critical analysis of the set of complex changes proposed in the literature and the documents mentioning them. We found out (i) the documents that have mentioned more complex changes; (ii) the most cited and, thus, relevant complex change of each kind (class, property, axiom, or annotation); (iii) the least cited complex change of each kind; (iv) the documents that mention almost the same set of changes written or not by the same set of authors; and (v) the number of documents that formalize the complex changes, which only provide textual descriptions, and those that only mention the complex change.
- *Most relevant documents that mention complex changes.* The set of most relevant documents is the minimum set of materials used as a reference for the collection of complex changes included in the unified list. The paper briefly presents those documents and comments on how they were chosen.
- *List of open directions.* After investigating several approaches that propose sets of complex changes, we were able to list critical open directions of the area.

This paper is particularly valuable to different personas involved in the ontology evolution process, such as:

- *Researches being introduced to the area.* Researchers being introduced to the area can easily find in this paper the set of primary documents used as a reference for the complex changes. The reading of those documents and comparing the different complex changes are facilitated. The created mapping states similarities and differences between complex changes, besides description and formalization;
- *Users selecting a tool to check for complex changes.* The users can more easily compare the set of complex changes supported by such tools due to the mappings between the complex changes and the classifications provided in this paper;
- *Ontology engineers willing to define new complex changes.* They can find in this paper a unified list of the complex changes already proposed in the literature together with their classifications, descriptions, and formalization (see Appendices). Such information helps in understanding the state-of-the-art before proposing a new complex change;

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

- *Ontology engineers willing to create a change language.* Ontology engineers that want to create a change language, that is, a language to represent the changes, will find in this paper the list of all complex changes proposed in the literature together with their synonyms, hyponyms, and hypernyms (as stated in the mappings). The synonyms indicate that different terms and labels can be used to refer to the same change, while the hyponyms and hypernyms may be used to define the specializations between the complex changes;
- *Developers of tools to check for complex changes.* They are also users of this paper since the classification and the mappings may guide the definition of heuristics able to check for the changes. The classification states that different complex changes share similar intentions, indicating that similar heuristics could be developed. The mappings describe hyponyms and hypernyms, meaning that one heuristics could be defined as a specialization of another;
- *Developers of tools able to (semi-)automatically adapt artifacts.* Such artifacts are adapted by propagating the ontology changes to the application level. Thus, application developers who may implement such adaptations need to clearly understand the changes made to the original version of the ontology. Once having the list of changes (provided by, for instance, a checker tool), developers may use as reference the description and formalization of each change provided in this paper. Such information helps interpret what happened and find the best way to adapt the artifacts.

The remainder of this paper is organized as follows. In Section 2, we describe the steps followed in the literature review we conducted. Section 3 presents the unified list of complex changes, explains how the mappings are represented, and illustrates one example of mapping. All other mappings are described in Appendices 1, 2, 3, and 4. The Appendices are provided as a Supplementary Material and are available at `https://researcher.watson.ibm.com/researcher/view_group_subpage.php?id=10992`. In Section 4, the classifications created to organize and group the complex changes are described. Section 5 presents a critical analysis of the mappings created between different complex changes, the documents, and the number and kind of complex changes mentioned in each document. In Section 6, the minimal set of documents necessary to find out all complex changes mentioned in the literature is described. Section 7 presents the threats to validity of this work and Section 8 discusses about some open directions. Finally, Section 9 presents the conclusion.

2. Literature review

The steps of the literature review applied in this paper were inspired by Kitchenham *et al.* (2009), Paré and Kitsiou (2017), Templier and Paré (2015), and Paré *et al.* (2015).

Formulating the problem Our literature review started by formulating the research objectives and questions. The goals of the review were to: (i) find documents mentioning complex ontology changes; (ii) check if the set of changes mentioned in the documents are the same; (iii) check if the authors use the same terminology to refer to the same change; and, (iv) check if a textual description or formalization supported each identified change. We are interested in complex changes applied to OWL or RDFS ontologies. Both OWL[3] and RDFS[4] are supported by the W3C consortium, and they are two of the most popular languages to create ontologies (Kalibatiene & Vasilecas, 2011). Since different ontology languages support distinct features and the changes are applied to those features, we need to restrict the scope of these languages. For instance, unlike OWL, some languages do not support 'Disjoint' and 'Equivalence'. Therefore, complex changes such as 'Add Disjoint Set' or 'Modify Equivalence to SuperClass' do not apply to those languages. Without this restriction, we would have some complex changes conditional on distinct languages.

[3] https://www.w3.org/OWL/.
[4] https://www.w3.org/TR/rdf-schema/.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Following the goals, we have defined the research questions below:

1. What are the complex changes proposed in the literature?
2. Are the complex changes supported by textual description or formalism?
3. Do the documents use the same terminologies to name the complex changes with the same semantics?
4. If not, is it possible to map the terminologies and produce a unified list of complex changes?

Searching the literature The literature review search focused on the four main digital libraries that include conference and journal papers related to the topic of this work. They are *ACM*[5], IEEE[6], *Springer*[7], and *Elsevier*[8]. In addition, we have also used *DBLP*[9] and *Google Scholar*[10] because we wanted to check (i) if there were any other publishers that should be considered in the search; and, (ii) if there were other kinds of documents besides papers (such as books and theses) mentioning complex changes that could not be found when searching in the four digital libraries. All the documents returned in *DBLP* search were briefly analyzed by considering their titles and publishers. After such analysis, only 39 papers were considered for further investigation since the others were not related to the topic of this paper. The analysis of the documents coming from the use of *Google Scholar* considered the first 100 unique documents returned. This limit was set considering that search results are ranked by their similarity to the given search terms. We concluded that: (i) more than half of the documents returned when searching in *Google Scholar* were also returned when searching in the digital libraries, and no other digital library (or venue) had a representative presence (i.e., at least 5% of the documents returned) on the set of returned documents; and (ii) the search in *Google Scholar* was important to identify theses that mention relevant sets of complex changes. From the set of documents that only appears in *Google Scholar*, almost 15% of the documents were theses, and almost 50% of these theses identify and describe complex changes.

The search string we used represents our interest in searching for documents mentioning complex changes or related changes applied to ontology.

```
((“change” OR “evolution”) AND “ontology” AND (“complex change” OR
“composite change” OR “high-level changes”))
```

Screening for inclusion Using this search string, we have found 229 papers in *ACM*, 160 papers in *Springer*, 35 papers in *IEEE*, 31 papers in *Elsevier* and 775 papers in *DBPL*. After removing duplicates and including the 100 documents found in *Google Scholar* we arrived at 553 in total. We analyzed these 553 documents to check if they genuinely mention examples of complex changes. Our *criterion* to consider a document as one mentioning complex changes was: it must, at least, name a complex change operation applied to an OWL or RDFS ontology. Hence, we have not excluded documents that have not been formalized and have not provided a textual description of the complex change.

We *defined complex change* (`Definition` 2.1) to make this meaning clear to be able to include or not a document in the list of documents mentioning such set of changes. By using such *criterion* and *definition*, we selected 34 documents (between books, papers, and theses) mentioning complex change operations.

Definition 2.1 *Complex change is an operation composed of multiple basic change operations that may incorporate some additional knowledge about the change (Stuckenschmidt & Van Harmelen, 2005) applied to an OWL or RDFS ontology.*

Assessing the quality of primary studies For each selected document, we checked if it does only mention complex changes (8 documents), if it includes textual descriptions for complex changes

[5] https://dl.acm.org/dl.cfm.
[6] https://ieeexplore.ieee.org/Xplore/home.jsp.
[7] https://link.springer.com/.
[8] https://www.sciencedirect.com/.
[9] https://dblp.org/.
[10] https://scholar.google.com.br/.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

(20 documents), and if it also provides a formalization of complex changes (7 documents). Textual descriptions and formalizations give us a high degree of confidence in a given mapping. Besides, this confidence is more elevated when formalizations are present. Therefore, our confidence in mappings involving changes from documents that only mentioned them is lower than when documents provide a formalization or textual description. In Section 7, we present the threats to validity related to this investigation.

Extracting data The complex changes unified list was created while reading each document. The first complex changes to be included in the unified list of changes were the ones mentioned in the first document we faced. When a new document was analyzed, we tried to match each complex change mentioned in the new document with one on the current set of changes. If no match was found, the set of changes was updated to include the new change. Sometimes, it was necessary to review some of the changes already in the set to ensure that the mappings already created were still valid. It was the case, for example, when a new change (e.g., *move class to a different hierarchy*) introduced a slight refinement to another already in the set (e.g., *move class*). We double-checked the related matching to guarantee that they were adequate.

The unified list of complex changes contains 58 changes applied to classes, 46 applied to properties, 2 applied to axioms, and 6 applied to annotations. We have excluded all changes applied to instances, metaclasses, or metaproperties, like the ones detailed in Papavasileiou *et al.* (2013) and Kondylakis (2010), since they were not the focus of this paper.

After finishing the unified list of complex changes, we started classifying complex changes applied to classes and properties. The hierarchical relationship between some of them influenced the creation of the *hierarchical classification*. The definition of sets of changes having similar abstract intentions or purpose influenced the creation of the *intention classification*.

Analyzing data To produce the unified list of complex changes, we tried to map each complex change mentioned in a new document with one of the complex changes already included in the unified list. The mappings produced and their justification are presented in Appendices 1, 2, 3, and 4.

3. Unified list of complex changes

The unified list of complex changes was created based on the mappings between the changes published in the literature. Our goal was to map the complex changes with similar semantics, that is, find out the complex changes that are synonyms. The mappings were defined by comparing the formalization, textual description, and terminology used to refer to each complex change. After finding complex changes with similar semantics, we grouped them and selected one complex change having at least a textual description to be the reference for such a group.

Tables[11] 1, 2, and 3 represent the result of the mappings we have created among the complex changes we found out in the documents. Such tables can be seen as a *thesaurus* for complex changes. In the first column we have a complex change included in the unified list and in the second column the set of synonyms found in the documents mentioned in the third column. The forth column presents the number of documents where the complex change was mentioned. Complex changes marked with * have no accompanying textual description or formalization. The changes applied to axioms (*axiom split* and

[11]The numbers in these tables refer to the following references: [1](Stojanovic, 2004); [2](Stojanovic *et al.*, 2002); [3](Klein & Alexander, 2004); [4](Rahnama & Barforoush, 2015); [5](Papavasileiou *et al.*, 2013); [6](Hartung *et al.*, 2013); [7](Rogozan & Paquette, 2005); [8](Najla *et al.*, 2009); [9](Xie *et al.*, 2011); [10](Dos Reis *et al.*, 2014); [11](Khattak *et al.*, 2008); [12](Tang & Yang, 2007); [13](Kirsten *et al.*, 2011); [14](Javed *et al.*, 2013); [15](Maedche *et al.*, 2002); [16](Gröner *et al.*, 2010); [17](Stuckenschmidt & Van Harmelen, 2005); [18](Palma *et al.*, 2009); [19](Liu *et al.*, 2014); [20](Khattak *et al.*, 2013); [21](Dinh *et al.*, 2014); [22](Dos Reis *et al.*, 2013); [23](Javed *et al.*, 2012); [24](Hartung *et al.*, 2010); [25](Mahfoudh *et al.*, 2015); [26](Rogozan & Paquette *et al.*, 2009); [27](Kondylakis *et al.*, 2010); [28](Djedidi & Aufaure, 2010); [29](Khelladi *et al.*, 2015) [30](Galani *et al.*, 2016) [31](Galani *et al.*, 2015) [32](Noy *et al.*, 2004) [33](Khattak *et al.*, 2013) [34](Herrmannsdoerfer *et al.*, 2001).

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 1. *Complex changes applied to Annotations, their synonyms and related documents*

Complex changes	Synonyms	Documents	#of documents
Retire class		[3]	1
Retire property		[3]	1
Change comment		[5][27]	2
Change label	Replace name	[3][5][27]	3
Revoke obsolete	revokeObsolete(c), rvkObsolete(c)	[24][10][21][22]	4
To obsolete	toObsolete(c)	[24][10][21][22]	4

axiom joint) are mentioned only in document (Rahnama & Barforoush, 2015). Thus, neither mapping nor table was necessary.

To illustrate the mappings we defined, we created the graphical notation presented in Figure 1. The five elements of the graphical notation are:

- The complex change representing the group is illustrated as an oval with the terminology included in the unified list as its label.
- The complex change used as the reference is illustrated with a bold oval having the terminology used to refer to the complex change and the reference to the document mentioning it. There is only one per change group.
- Other complex changes included in the group are illustrated by an oval having the terminology and the reference to the document.
- The complex changes having textual description and/or formalization are linked by a solid line to the complex change representing the group.
- The complex changes that do not have a textual description are linked by a dashed line to the complex change representing the group. It indicates that the mapping was based only on the terminology since no description was provided.

Figure 2 depicts an example of the proposed graphical representation considering the complex change *rename class*. The *rename class* complex change included in the unified list presented in Table 2 was defined based on the complex change that receives the same terminology and was presented in Kondylakis (2010) (as indicated by the bold oval in Figure 2). Three other documents mention the same complex change, but two use different terminology. In Papavasileiou *et al.* (2013), the authors use *class renaming* and in Rogozan and Paquette (2009) they use *Modify_Class_Name*. In Rogozan and Paquette (2009), the authors do not provide a textual description for such complex change. Thus, we indicate by a dashed line that the mapping between *rename class* and *Modify_Class_Name* was based solely on the terminology used to refer to the complex change, that is, based on the string *Modify_Class_Name*. Due to space limit, all mappings are detailed in Appendices 1, 2, 3, and 4.

4. Classification of complex changes

The literature's approaches classify the complex changes according to the *kind of entity* they apply. Following this classification, there are four sets of complex changes: class (grouping complex changes applied to classes), property (for those applied to properties), axiom (for those applied to axioms), and annotation (for those applied to annotations). Tables 1, 2, and 3 (Section 3) present the list of complex changes following such classification.

Due to the numerous complex changes available, we felt the need to provide to the readers other possible classifications with the aim to increase the understanding and usability of the complex changes. So, in addition to the previous classification based on *the kind of entity*, we further classified the class and property complex changes in two dimensions according to their *intention* and their *hierarchical*

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 2. *Complex changes applied to Classes, their synonyms and related documents*

Complex change	Synonyms	Documents	#documents
Rename class	ModifyClassName, class renaming	[3][5][26[27][30]	9
	Rename_Class(a)	[31][33][34][34]	
Substitute concept c1 by c2	substitute, subst	[6][10][21][22][24]	5
Extract related concept	group	[2][12][15][19]	4
Extract class		[16][29][34]	3
Extract subclass	Extract Subconcepts	[15][16][29][34]	4
Extract superclass	Group classes (x,(x1,x2)), Extract SuperClass(C)	[1][2][9][12]	8
	Extract Superconcept, GroupConcepts (c1, c2, newC)	[14][16][23][34]	
Fold class		[34]	1
Fold superClass		[34]	1
Unfold superClass		[34]	1
Unfold class		[34]	1
Shalow concept copy	Copy, CopyConcept	[1][2][12][19]	4
Deep concept copy		[1][2][12]	3
Inline class		[16][29][34]	3
Inline superclass		[34]	1
Inline subclass		[34]	1
Add subtree	addSubGraph, sub-Hierarchy	[3][6][13][18][24][28]	6
Delete subtree	delSubGraph	[3][6][24][32]	4
Delete subclasses*		[3]	1
Add leaf	addLeafC, add a subclass A of C, Add subclass	[6][10][17][21][22][25][33]	7
Delete leaf	Delete subclass	[6][10][21][22][25][33]	6
Group classes	Group, Group concepts Group Classes, Concept Join	[4][5][8][19][27]	5
Ungroup classes		[5][27]	2
Add specialization class	addInterior, addIn, AddInterior-Concept	[1][3][10][14]	8
	Create new abstraction, Add Concept Specialization	[19][21][23]	
	add specialisation concept, add generalisation addInnerC, add specialisation class	[31]	
Add generalization class	Add Concept Generalisation, Add_Superclass(a, b)	[1][14][19][23]	4
	add generalisation concept, add Specialisation		
Delete an abstraction	Delete Class and Move Siblings Up, delIn, delInnerC	[3][10][11]	6
	delInnerC, Removing parent class, Delete_Superclass(a, b)	[21][22][31]	

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 2. *Continued.*

Complex change	Synonyms	Documents	#documents
Move class	Change Superclasses, Move Concept	[1][3][5][14]	8
	Move Change	[23][25][26][33]	
Pull up class	ModifySuperclass-ToSuperclass	[1][3][4][5]	12
	Generalization, Pull up concept,	[8][14][17][19]	
	Reclassify Class Higher Move up,	[23][25][27][28]	
	Pull Concept Up, Concept Move up		
Pull down class	Pull Class down, Specialization	[1][3][4][5]	12
	ModifySuperclass-ToSubclass	[8][14][17][19]	
	Lower concept, Pull Concept Down,	[23][25][27][28]	
	Move Down, Concept Move down		
	Reclassify Class Lower, Pull down concept		
Move subtree	movec, Concept Group, Move Subtree,	[3][5][6][10]	9
	Concept Group Move, Move, Movec	[21][22][24][28][32]	
Move class dif. hierarchy	move concept, move class	[5][14][27]	3
Move sibling classes	move sibling concepts, move siblings	[1][3]	2
	Move Siblings to new subclass		
Pull up sibling classes*		[3]	1
Pull down sibling classes		[3]	1
Split a class	splitting a class, Extract Subconcepts	[1][2][4][5][6][7][9]	18
	oc:split, split, Split_Class(a, B), Split Concept	[10][11][12][13][31]	
	Concept split, Extract subconcepts	[18][21][22][24][26][34]	
Split hierarchy	leafSplit, Extract Hierarchy, Split of concept	[6][8][16]	3
Split into existing	Split_Class_Into_Existing(a, B)	[5][27][31]	3
S|plit into multiple siblings |	Split class, Split concept	[3][13][22]	3
Merge classes	merge concepts, merge, MergeClasses(C)	[1][2][4][5][6][7][8][34]	24
	merging two or more classes, oc:Merge	[9][10][11][12][13][33]	
	Fusion of concepts, Concept Merge	[14][15][20][21][22]	
	Complex union classes, Merge_Classes(A,b)	[23][24][26][27][31]	

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 2. *Continued.*

Complex change	Synonyms	Documents	#documents
Merge class into existing	Merge Classes into existing one, Incorporate Class Merge_Classes_Into_Existing (A, b)	[3][5][27] [31]	3
Collapse hierarchy	leafMerge	[6][16]	2
Merge siblings	MergeMultiple- Siblings	[3][19]	2
Modify equivalent class*		[26]	1
Modify equivalence To Subclass*		[3]	1
Modify equivalence To superclass*		[3]	1
Move disjoint class*		[26]	1
Modify disjointness To Subclass*		[3]	1
Modify disjointness To superclass*		[3]	1
Modify type to subclass*		[3]	1
Modify type to superclass*		[3]	1
Modify subclass to superclass*		[3]	1
Modify superclass to subclass*		[3]	1
Change to primitive*		[3]	1
Change to defined*		[3]	1
Add disjoint set*		[3]	1
Remove disjoint set*		[3]	1
Modify union*		[26]	1
Modify complement*		[26]	1
Modify intersection*		[26]	1

relations. After investigating all complex changes, we figure out that some have similar intentions or are hierarchically related. Such classifications contribute to the development of heuristics to check for complex changes and adapt the artifacts.

Tables 4 and 5 classify all complex changes applied to classes and properties following the *entity kind*, *intention,* and *hierarchy* classification.

4.1. Intention classification

The *intention classification* emphasizes that different complex changes share similar semantics. All group members share the intention of the complex changes included in the same group. We found out the following different main intentions:

- *Entity Aggregation*: The complex changes of this group are related to the aggregation of entities or their separation. The entities are combined into one unique entity, or one entity is separated into two or more entities. For instance, *split class* divides one class into two or more

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 3. *Complex changes applied to Properties, their synonyms and related documents*

Complex change	Synonym	Documents	total #of documents
Rename property	Modify Property-Name, Rename_Property(a, b)	[5][11][31]	5
	Renaming a property	[26][27]	
Shallow property copy		[2][12]	2
Deep property copy		[2][12]	2
Merge properties	Merging two or more properties, oc:Merge	[4][5][7]	6
	Relation Merge, Fusion of relationships	[8][11]	
	Merge_Properties(A, b), Merge Feature	[31][34]	
Merge properties into existing	Merge Properties Into Existing(A,b)	[5][27]	3
	Merge_Properties_Into_Existing (A, b)	[31]	
Split property	Splitting a property, Relation Split, oc:Split	[4][5][7]	6
	Split of relationships, Merge_Properties_Into_Existing (A, b)	[8][11][31]	
Split properties into existing	Split_Property_Into_Existing (a,B)	[5][27][31]	3
Move property in the class hierarchy		[2][12][16][29]	4
Pull up property in the class hierarchy	Pull property, Pull up feature	[2][12][14][29][34] [16][23]	7
Pull down property	Push-down property, Push propertyy	[2][12][14]	7
in the class hierarchy, push down feature		[16][23][29][34]	
Pull up property in the property hierarchy		[5][27]	2
Pull down property in the property hierarchy	Push-down property	[5][27]	2
Move properties to a different property hierarchy		[27]	1
Change range	Property modification details,	[5][20][26][27]	4
Modify object property range*		[26]	1
Modify datatype property value*		[26]	1
Restrict range	Restrict the range of a relation R, Delete_Range(a, b)	[3][17][28][31]	4
Extend range*	Add_Range(a, b)	[3][28][31]	3
Modify range to subclass		[3][5][27]	3

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 3. *Continued.*

Complex change	Synonym	Documents	total #of documents
Modify range to superclass		[3][5][27]	3
Change to object property		[3][27]	2
Change to datatype property		[3][27]	2
Change domain		[5][20][26][27]	4
Modify dataType property domain*		[26]	1
Restrict domain*	Delete_Domain(a, b)	[3][31]	2
Extend domain	Add_Domain(a, b)	[3][17][31]	3
Modify domain to subclass		[3][5][27][28]	4
Modify domain to superclass		[3][5][27][28]	4
Add superproperty		[31]	1
Delete superproperty		[31]	1
Change cardinality*		[16]	1
Restrict cardinality*		[3]	1
Extend cardinality*		[3]	1
Increase lowerbound*		[3]	1
Decrease lowerbound*		[3]	1
Increase upperbound*		[3]	1
Decrease upperbound*		[3]	1
Group property		[5][27]	2
Ungroup property		[5][27]	2
Modify equivalente property*		[26]	1
Modify equivalence to subproperty*		[3]	1
Modify equivalence to superproperty*		[3]	1
Modify inverse property*		[26]	1
Modify inverse to subproperty*		[3]	1
Modify inverse to superproperty*		[3]	1
Unidirectional to bidirectional		[16]	1
Bidirectional to unidirectional		[16]	1
Modify super property		[5][26][27]	3

classes while *merge class* groups two or more classes in one. These kinds of changes tackle two complementary problems on the original ontology. On the one hand, if there are in the original ontology two or more entities sharing similar characteristics, the *merge class/property* should be used. On the other hand, if there is in the original ontology one entity having several unrelated characteristics, such entity should be split into two or more entities by using *split class/property*.

- *Entity Mirror*: The complex changes of this group behave similarly to mirrors. Some of them create a copy of the original entity (such as *shallow and deep class/property copy*), and others, in the case of classes, substitute a class with another (such as *substitute concept c1 by c2*).

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

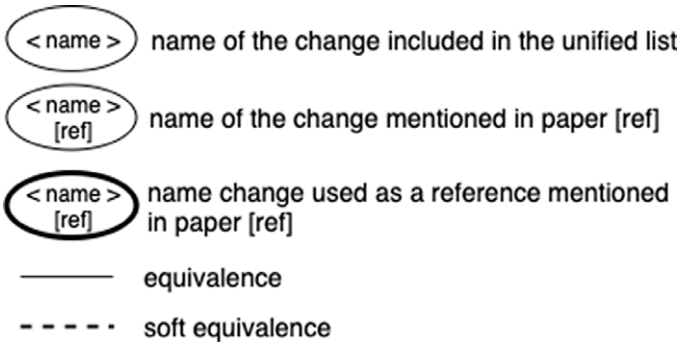


Figure 1. *Graphical notation used in the mapping figures.*

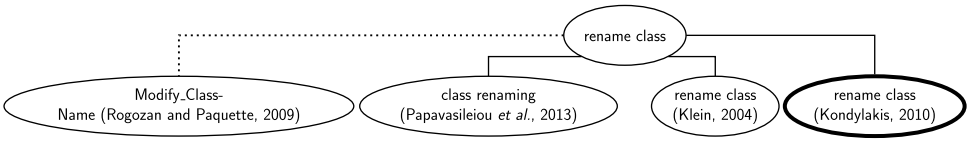


Figure 2. *Rename class mapping.*

- *Hierarchy Change*: The complex changes included in this group intend to change the hierarchy of entities by adding, deleting, or moving entities in the hierarchy. The adding or moving of an entity to a given hierarchy occurs because the entity shares similar characteristics with its new parents. The removal of an entity from a hierarchy occurs because the characteristics of such entity are no more shared with its parents or because the entity was deleted in the new ontology version.

    1. *Adding Entities*: The complex changes of this group add one or more entities to the hierarchy of entities. Such entities are sometimes not new since they might already exist in the ontology but not in the target hierarchy. For instance, *add subtree* adds a subtree of classes to the target class hierarchy.
    2. *Deleting Entities*: These complex changes exclude one or more entities from the hierarchy of entities. Such entity(ies) sometimes is(are) not deleted from the ontology but only excluded from the hierarchy. For instance, *ungroup properties* excludes the hierarchy relationship shared by all properties of the group.
    3. *Moving Entities*: The complex changes of this group delete one or more entities from a place in the hierarchy and add such entity(ies) in another place of the same hierarchy or a different hierarchy. For instance, *pull up class* moves a class one or more hops up in the same hierarchy of classes.

- *Characteristic Change*: The complex changes of this group modify the characteristics of the entities. In the case of classes, it is possible to change the name or the set of properties by moving some of the properties to another class. In the case of properties, it is possible to change the name, the range, domain, or cardinality of the property.
- *Description Change*: The complex changes in this set modify the description of the class by making it equivalent to another class or a subclass of another class.

Some complex changes included in the unified list were mentioned in the original documents without formalization or textual description. Therefore, it was challenging to infer the semantics of those changes based solely on the terminology used to refer to them. We could infer the semantics and, thus,

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 4. *Classification of complex changes applied to classes*

		Complex change	
Entity kind	Intention	Hierarchy Level 1	Hierarchy Level 2
Class	Entity aggregation	Split a class	Split into existing
			Split into multiple sibling
			Split hierarchy
		Merge classes	Merge into existing
			Collapse hierarchy
			Merge siblings
	Entity mirror	Substitute concept c1 by c2	
		Shallow concept copy	
		Deep concept copy	
	Hierarchical change:	Add subtree	
		Add leaf	
	Adding entities	Group classes	
		Add specialization class	
		Add generalization class	
	Hierarchical change:	Delete subtree	
		Delete subclasses	
	Deleting entities	Delete leaf	
		Ungroup classes	
		Delete an abstraction	
	Hierarchical change:	Move class	Pull up class
			Pull down class
			Move class to different hierarchy
	Moving entities	Move subtree	
		Move sibling classes	Pull up sibling classes
			Pull down sibling classes
	Characteristic change	Rename class	
		Extract related concept	
		Extract class	Extract subclass
			Extract superclass
		Fold class	Fold superclass
		Unfold class	Unfold superclass
		Inline class	Inline subclass
			Inline superclass
	Description change	ChangeToPrimitive	
		ChangeToDefined	
	Not specified	ModifyEquivalentClass	ModifyEquivalenceToSubclass
			ModifyEquivalenceToSuperclass
		MoveDisjointClass	ModifyDisjointnessToSubclass
			ModifyDisjointnessToSuperclass
		ModifyTypeToSubclass	
		ModifyTypeToSuperclass	
		ModifySuperclassToSubclass	
		ModifySuperclassToSuperclass	
		AddDisjointSet	
		RemoveDisjointSet	
		Modify union	
		Modify complement	
		Modify intersection	

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Table 5. *Classification of complex changes applied to properties*

		Complex change	
Entity kind	Intention	Hierarchy Level 1	Hierarchy Level 2
Property	Entity aggregation	Split property	Split properties into existing
		Merge property	Merge properties into existing
	Entity mirror	Shallow property copy	
		Deep property copy	
		Unidirectional to bidirectional reference	
		Bidirectional to unidirectional reference	
	Hierarchical change: Adding entities	Group property	
	Hierarchical change: Deleting entities	Ungroup property	
	Hierarchical change: Moving entities	Move properties in the Class hierarchy	Pull up in the class hierarchy
			Pull down in the class hierarchy
		Pull up in the property hierarchy	
		Pull down in the property hierarchy	
		Move properties to a different property hierarchy	
		Modify super property	
	Characteristic change	Rename property	
		Change range	Modify object property range
			Modify datatype property value
			RestrictRange
			ExtendRange
			ModifyRangeToSubclass
			ModifyRangeToSuperclass
			Change to object property
			Change to datatype property
		Change domain	Modify data type property domain
			RestrictDomain
			ExtendDomain
			ModifyDomainToSubclass
			ModifyDomainToSuperclass
		Change cardinality	RestrictCardinality
			ExtendCardinality
			IncreaseLowerbound
			DecreaseLowerbound
			IncreaseUpperbound
			DecreaseUpperbound
	Not specified	ModifyEquivalent Property	ModifyEquivalenceToSubproperty
			ModifyEquivalenceToSuperproperty
		ModifyInverseProperty	ModifyInverseToSubproperty
			ModifyInverseToSuperproperty

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

classified according to the previous classification for some of them. However, others were classified as *Not Specified*.

- *Not Specified*: The complex changes included in this group have no definition presented in the literature and, thus, do not participate in any mapping. Since it was challenging to infer the correct meaning, they were not classified according to their intention.

4.2. Hierarchical classification

The *hierarchical classification* represents the hierarchical relationship between one complex change and another. Some of the complex changes are defined in a more general way, and others are more specifically defined by extending the semantics of the generic one. For instance, *move class* is a complex change that generically defines that a class was moved from one place to another in a hierarchy. On the one hand, the complex change *pull up class* specializes *move class* by stating that the class has moved UP in the same hierarchy. On the other hand, the complex change *move class different hierarchy* specializes *move class* by stating that the class has moved to another hierarchy. In Tables 4 and 5, the complex changes included in the column called Hierarchy Level 2 extend the generic complex changes included in the column called Hierarchy Level 1.

5. Critical analyses

In this section, we present analyses about the mappings we created among the complex changes found in the 34 documents (Section 5.1) and analyses related to the changes mentioned in each document and their characteristics (Sections 5.2, 5.3, and 5.4).

The main contribution of this section is to provide evidence for the need to unify and formalize the set of complex changes available in the literature. In summary, we highlight:

- The number of complex changes is big. We found 112 complex changes: 58 applied to classes, 46 applied to properties, 6 applied to annotations and 2 applied to axioms.
- The set of complex changes is spread across different documents. There is no document describing the majority of the changes. Several subsets that intersect are mentioned in separate documents.
- Several documents use different terminologies to refer to the same change, while others use the same vocabulary to refer to distinct changes.
- There are several complex changes mentioned in only one document and others mentioned in several documents.
- The majority of documents only present a textual description of the identified changes. Several documents do not give any textual description or formalization for the complex changes. Several complex changes are mentioned in only one document and do not have any textual description or formalization.

5.1. Mapping challenges

The main challenges we faced when mapping complex changes from different documents were:

- *Same terminology with different meanings*: Some documents use the same terminology to refer to changes with different meanings. For instance, in Stojanovic *et al.* (2002) and Tang and Yang (2007), the authors use *move property* to indicate that a property was moved from one concept to another concept. But in Kondylakis *et al.* (2010), the same terminology, *move property*, is used to indicate that a property was moved to a different location in another class hierarchy. It means that the change described in Kondylakis *et al.* (2010) is, in fact, a specialization of the change described in Stojanovic *et al.* (2002) and Tang and Yang (2007).

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

- *Different terminologies with the same meaning*: We faced some examples of documents using different terminologies to refer to changes with the same meaning. For instance, *extract subconcept* is used in Tang and Yang (2007) to split a concept into several subconcepts and distribute properties among them. In several other documents, the authors use *split* to refer to a change with such a definition. Another example is the case of *extend domain* mentioned in Klein (2004) and *domain enlarged* mentioned in Stuckenschmidt and Van Harmelen (2005). Although Klein (2004) and Stuckenschmidt and Van Harmelen (2005) do not provide definitions for such changes, by taking into account the terminologies used to refer to the changes, we considered that they have the same meaning.
- *General terminology matching specific changes*: Some documents mention changes that are generalizations of changes that have been included in the unified list. It is the case of Khattak *et al.* (2013) that mentions *setting upper bound* that is a generalization of *increase upperbound* and *decrease upperbound* (Klein, 2004). By considering that (i) *setting upperbound* is composed of inclusion, deletion and modification of upperbound, (ii) inclusion and deletion are not complex changes, and (iii) *increase upperbound* together with *decrease upperbound* represent all possibilities related to *setting upper bound*, we did not include *setting upper bound* in the list but *increase upperbound* and *decrease upperbound*.
- *Similar terminologies used to define very similar changes in the same document*: In both Papavasileiou *et al.* (2013) and Kondylakis *et al.* (2010), there are similar terminologies that describe very similar or even the same change. It is the case of *pull up class* and *reclassify class higher* that have the same definition. Both are defined as: 'move class A to a higher position in the subsumption hierarchy'.
- *'Well-known' complex changes mentioned in document but without textual description or formalization*: As detailed in Subsection 5.3, *split* and *merge* of both classes and properties are mentioned in several documents. Although there are documents that describe and sometimes present a formalization for these changes, there are documents that do only mention them: Rogozan and Paquette (2005), Najla *et al.* (2009), Khattak *et al.* (2008), Kirsten *et al.* (2011), Maedche *et al.* (2002), Palma *et al.* (2009), and Khattak *et al.* (2013). In such cases, we are assuming that the meaning intended by the authors follows the meaning formalized or described in the documents, such as the changes described in Rahnama and Barforoush (2015), Xie *et al.* (2011), and Kondylakis *et al.* (2010).
- *No definition at all for the change*: Several changes are mentioned in only one document, and such a document does not provide any textual description or formalization for those changes. It is the case of *modify disjointness to superclass* mentioned only in Klein (2004).

5.2. Number of complex changes per document

All 34 documents that mention complex changes have at least one mention of complex change applied to classes. Almost half of the analyzed documents mentioned both changes applied to classes and to properties (Stojanovic *et al.*, 2002; Klein, 2004; Rogozan and Paquette, 2005; Stuckenschmidt and Van Harmelen, 2005; Tang and Yang, 2007; Khattak *et al.*, 2008; Najla *et al.*, 2009; Rogozan and Paquette, 2009; Djedidi and Aufaure, 2010; Gröner *et al.*, 2010; Kondylakis *et al.*, 2010; Javed *et al.*, 2012, 2013; Khattak *et al.*, 2013; Papavasileiou *et al.*, 2013; Rahnama and Barforoush, 2015; Mahfoudh *et al.*, 2015; Djamel *et al.*, 2015; Galani *et al.*, 2015; and Herrmannsdoerfer *et al.*, 2010). A small number of documents Klein (2004), Hartung *et al.* (2010, 2013), Dos Reis *et al.* (2013), Papavasileiou *et al.* (2013), Dos Reis *et al.* (2014), Dinh *et al.* (2014), and Kondylakis *et al.* (2010) mentioned changes applied to annotations. In addition, only Rahnama & Barforoush (2015) mentioned changes applied to classes, properties, and axioms. In addition, it was the only document mentioning changes applied to axioms. Figure 3 illustrates the number of documents that mention each kind of element change. It is clear that changes applied to classes are the most popular ones followed by changes applied to properties.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press



Figure 3. *Number of documents mentioning each kind of element change.*

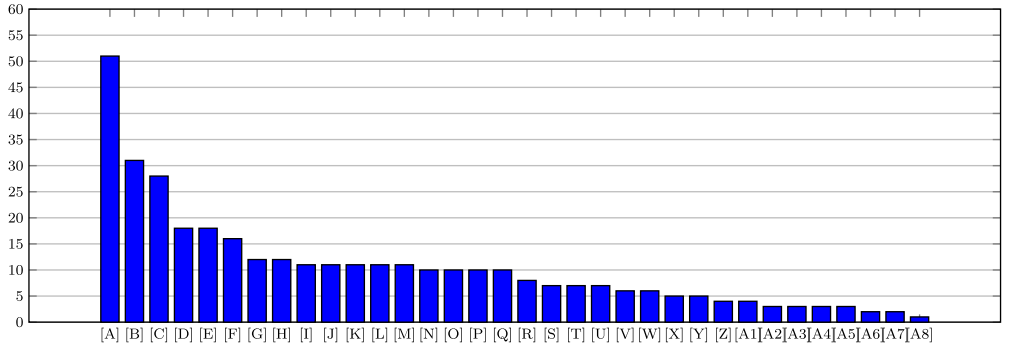


Figure 4. *Total number of changes mentioned per document.*

Although Rahnama & Barforoush (2015) mention changes applied to axioms when no one does, it only mentions two changes applied to properties and a few applied to classes. The fact that no other document mentions changes applied to axioms might indicate that axioms were not the object of their investigations, the ontologies analyzed have not described axioms, or changes applied to axioms do not frequently occur when comparing two versions of the same ontology.

Figure 4[12] depicts the total number of changes mentioned per document. As described before, we have identified 112 complex changes. From the set of 34 documents, 32% mention less than 6 changes, 50% of them mention more than 5 and less than 15 changes and only 17% documents mention more than 15 changes.

The document that mentions the greater number of changes was Klein (2004). It mentioned 51 changes, that is, around 44% of the changes identified in the 34 documents. Therefore, it is possible to conclude that no document is complete in mentioning all the complex changes proposed in the literature, which is one of the main goals of this paper.

Several documents written by the same authors mention the same changes, and documents written by different authors mention the same changes. We have compared the authors of the documents and the changes mentioned in each document. We noticed that:

[12]The references used in Figure 4 are defined as follows: [A](Klein, 2004), [B](Kondylakis *et al.*, 2010), [C](Papavasileiou *et al.*, 2013), [D](Rogozan & Paquette, 2009), [E][C7](Herrmannsdoerfer *et al.*, 2010) [F][C4](Falani *et al.*, 2015) [G](Gröner *et al.*, 2010), [H](Hartung *et al.*, 2013), [I](Stojanovic, 2004), [J](Stojanovic *et al.*, 2002), [K](Javed *et al.*, 2013), [L](Tang & Yang, 2007), [M](Rahnama & Barforoush, 2015), [N](Dos Reis *et al.*, 2014), [O](Dinh *et al.*, 2014), [P](Javed *et al.*, 2012), [Q](Hartung *et al.*, 2010), [R](Dos Reis *et al.*, 2013), [S](Najla *et al.*, 2009), [T](Liu *et al.*, 2014), [U](Djedidi & Aufaure, 2010), [V](Khattak *et al.*, 2008), [W][C3](Khelladi *et al.*, 2015) [X](Stuckenschmidt & Van Harmelen, 2005), [Y](Khattak *et al.*, 2013) [Z](Rogozan & Paquette, 2005), [A1](Khattak *et al.*, 2013), [A2](Xie *et al.*, 2011), [A3](Kirsten *et al.*, 2011), [A4](Maedche *et al.*, 2002), [A5](Palma *et al.*, 2009), and [A6](Mahfoudh *et al.*, 2015) [A7](Noy *et al.*, 2004), and [A8](Galani *et al.*, 2016).

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

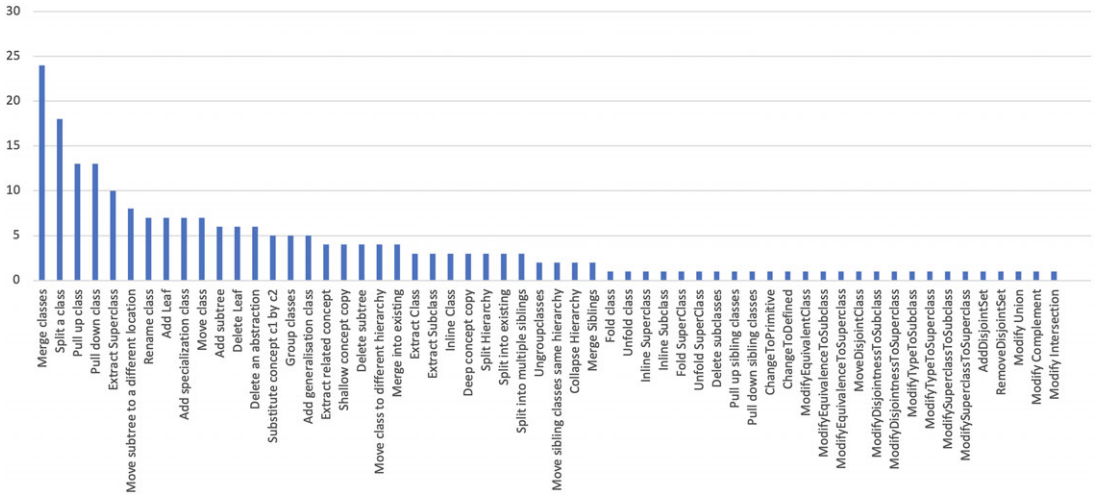


Figure 5. *Number of documents mentioning each change applied to classes.*

- Documents Hartung *et al.* (2010) and (2013) mention almost the same changes and have been written by the same authors.
- Documents Dos Reis *et al.* (2014), Dinh *et al.* (2014), and Dos Reis *et al.* (2013) mention the same set of changes and are almost from the same set of authors.
- Documents Dos Reis *et al.* (2014), Dinh *et al.*, (2014), Dos Reis *et al.* (2013), and Hartung *et al.* (2010) and (2013) mention almost the same set of changes since Dos Reis *et al.* (2014), Dinh *et al.* (2014), and Dos Reis *et al.* (2013) cite the changes mentioned in Hartung *et al.* (2013).
- Documents Stojanovic *et al.* (2002) and Tang and Yang (2007) mention precisely the same changes since Tang and Yang (2007) cite the changes mentioned in Stojanovic *et al.* (2002). They were not written by the same authors.
- Document Kondylakis *et al.* (2010) adopted the complex changes proposed initially in Papavasileiou *et al.* (2009a). The technical report Papavasileiou *et al.* (2009b) extended Papavasileiou *et al.* (2009a) with more complex changes and their formalizations. Papavasileiou *et al.* (2013) mention almost the same set of changes as in Papavasileiou *et al.* (2009b) but with some other formalizations.

5.3. Most and least mentioned complex changes

Changes applied to classes The most mentioned complex changes applied to classes are *merge of classes* and *split of a class*. As illustrated in Figure 5, *merge of classes* appeared in almost all documents (24 from a total of 34 documents). If we consider special cases of the *merge of classes* change (merge into existing, collapse hierarchy, merge siblings), the number of mentions increases to 27. The *split of a class* appeared in 18 documents. If we consider special cases of the *split of a class* change (split hierarchy, split into existing, split into multiple siblings), this number increases to 21. Besides being mentioned by the documents that list several complex changes, they are also mentioned by some of the documents that mention only a few examples of changes, that is, the documents that mention 5 or fewer complex changes. Therefore, we can conclude that these are considered representative examples of complex changes.

On the other hand, there are 25 class changes, that is, around 43% of the changes applied to classes, that are mentioned in only one document. In particular, 13 of these changes are mentioned only in the document Klein (2004). Since grouping basic changes form complex changes, this situation is expected. Different authors may propose their own set of complex changes.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

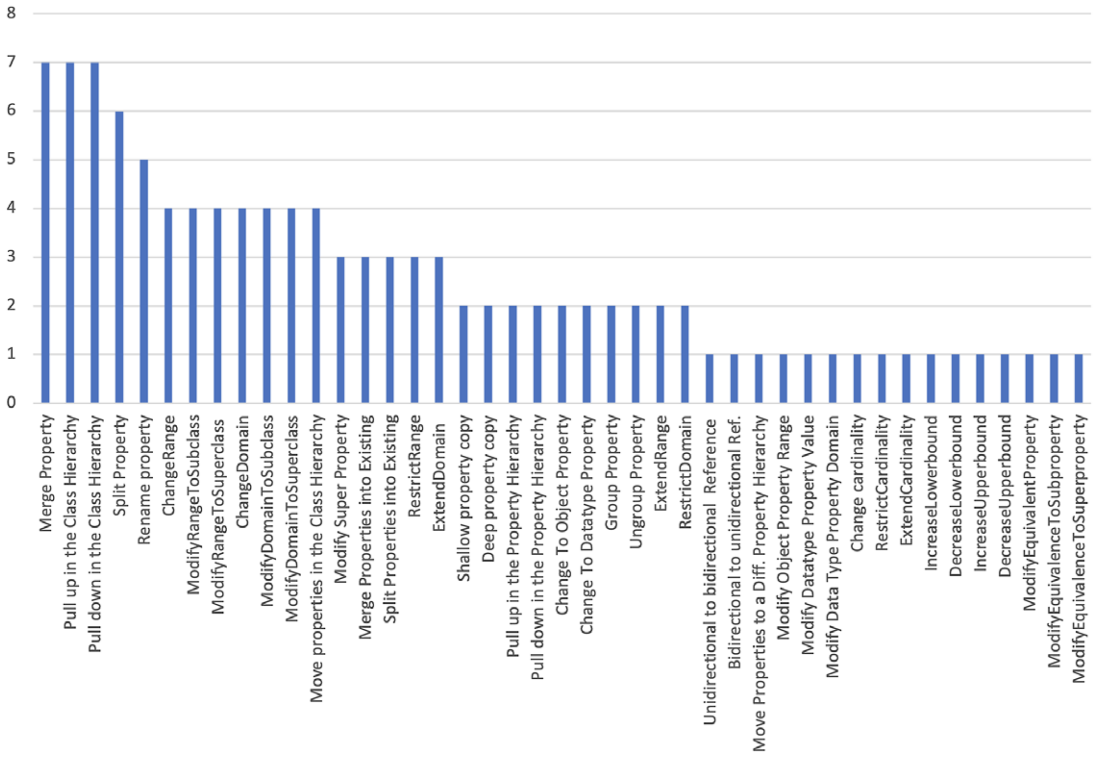


Figure 6. *Number of documents mentioning each change applied to properties.*

Also, almost half of complex changes applied to classes, that is, 44% of the changes, are mentioned in 5 or fewer documents. This may indicate that the approaches are considering very different complex changes in their use cases, and there might not be a consensus.

Changes applied to properties There are three changes applied to properties that were the most mentioned ones: *Merge Property*, *Pull up in the Class Hierarchy*, and *Pull down in the Class Hierarchy*, as illustrated in Figure 6. From the set of 19 documents mentioning changes applied to properties, these three changes have been mentioned by only 7 documents, that is, 36% of all documents mentioned the most cited changes. The majority of property changes are mentioned by 1 or 2 documents. This might indicate, for instance, that (i) there is no consensus in the literature about changes applied to properties, even in the case of the most mentioned changes, (ii) the use cases only consider a small set of changes, or (iii) it is difficult to detect such kind of changes.

Changes applied to annotations and axioms The most mentioned changes applied to annotations are the one indicating that a concept is obsolete and the one indicating that the obsolete characteristic has been revoked. They are mentioned in the same documents. The changes applied to *change comment* and *change label* are only mentioned by Papavasileiou *et al.* (2013) and Kondylakis *et al.* (2010). Figure 7 illustrates the number of changes applied to annotations. The only two changes applied to axioms are mentioned exclusively in one document Rahnama and Barforoush (2015).

After eliminating the duplicity of concepts mentioned in different documents written by the same authors, only the amount of 7 changes applied to classes have been updated. They are: *add specialization class, add subtree, add a subclass to several classes (add leaf), delete an abstraction, substitute concept c1 by c2, delete a subclass of several classes (delete leaf) and* delete subtree. The modification in the changes applied to classes has not influenced our analysis. Also, the number of changes applied to properties and axioms has not changed since those documents written by the same authors have not mentioned changes applied to properties or axioms. In the case of changes applied to annotations, the

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

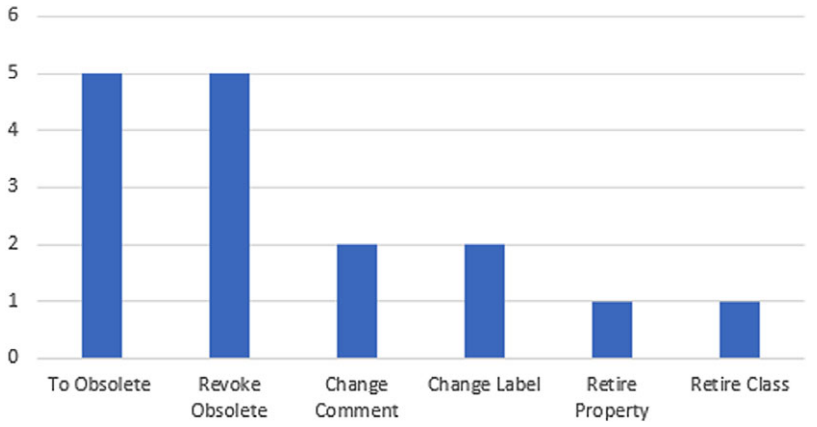


Figure 7. *Number of documents mentioning each change applied to annotations.*

amount of 2 changes have been modified (*toObsolete* and *revokeObsolete*) since they are mentioned in Dos Reis *et al.* (2013, 2014), Dinh *et al.* (2014), and Hartung *et al.* (2010) and (2013). In this case, such changes would not be considered the most mentioned changes applied to annotations.

5.4. Textual description and formalization of complex changes

From the set of 34 documents, only 8 documents Rahnama and Barforoush (2015), Xie *et al.* (2011), Liu *et al.* (2014), Hartung *et al.* (2010), Mahfoudh *et al.* (2015), Kondylakis *et al.* (2010), Khelladi *et al.* (2015), and Galani *et al.* (2015) present some formalization about the changes they mention and 9 documents Rogozan and Paquette (2005), Najla *et al.* (2009), Khattak *et al.* (2008, 2013), Kirsten *et al.* (2011), Maedche *et al.* (2002), Palma *et al.* (2009), Noy *et al.* (2004), and Khattak *et al.* (2013) do not present any textual description or formalization for the changes. In addition, even the 15 documents that present some textual description of the changes do not do so for all the mentioned changes. It is the case, for instance, of Klein (2004) that mentions 50 changes but does not provide descriptions for 38 of them. Therefore, several changes have no description at all. The changes that have no formalization or textual description have no mapping and are illustrated in the Appendices figures by a dashed line.

6. Main documents mentioning complex changes

The documents included in this section are the ones representing *the minimal set of documents necessary to find out all the complex changes mentioned in the literature*. They are not the ones with more citations, published more recently, or those that were the first mentioning complex changes. This minimal set of documents aims to provide a panorama of the state-of-the-art.

We started by selecting the documents mentioning complex changes that were not mentioned in any other document. Thus, the first documents we selected were Klein (2004), Rogozan and Paquette (2009), and Herrmannsdoerfer *et al.* (2010) (since they mention several complex changes applied to classes and properties not mentioned in other documents), Gröner *et al.* (2002) (since it mentions few changes applied to classes and properties not mentioned in other documents) and Rahnama and Barforoush (2015) (since it is the only document mentioning complex changes applied to axioms).

Since there were still complex changes included in the unified list that were not mentioned in those documents, we needed to include other documents in our minimal set of documents. We based our selection on the analyses presented in Section 5.2 where we identify the documents written by almost the same set of authors mentioning almost the same set of complex changes. Thus, we selected document Kondylakis (2010) and not Papavasileiou *et al.* (2013) since it mentions a few complex changes

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

not included in Papavasileiou *et al.* (2013) (and that were not mentioned in the previously selected documents). Next, we selected document Hartung *et al.* (2013) and not Hartung *et al.* (2010), Dos Reis *et al.* (2014), Dinh *et al.* (2014), and Dos Reis *et al.* (2013) since they all mention almost the same set of complex changes (not mentioned in the previous selected documents) and because Hartung *et al.* (2013) is cited by them. In the following, we selected document Stojanovic *et al.* (2002) and not Tang and Yang (2007) since Tang and Yang (2007) cites Stojanovic *et al.* (2002) and they mention the same set of complex changes (not mentioned in the previously selected documents). Finally, it was necessary to choose between Stojanovic (2004), Javed *et al.* (2013), Liu *et al.* (2014), and Javed *et al.* (2012) since they all cite the unique complex change not mentioned in the previously selected documents, *add generalization class*. We selected the paper Liu *et al.* (2014) since it was published more recently. In total, we selected 8 documents: Klein (2004), Rogozan and Paquette (2009), Herrmannsdoerfer *et al.* (2010), Gröner *et al.* (2002), Rahnama and Barforoush (2015), Kondylakis (2010), Hartung *et al.* (2013), Stojanovic *et al.* (2002) and Liu *et al.* (2014).

In Klein (2004), the authors propose a framework to deal with ontology evolution. As part of this work, the authors propose a definition for complex changes, describe a set of OWL simple and complex changes, and present transformations that are sets of change operations that specify how the old version of an ontology can be transformed in its new version.

From the set of 51 complex changes proposed in the document (as depicted in Tables 1, 2, and 3), (i) only 3 applied to classes (*incorporate class, adds disjoint set* and *remove disjoint set*); and (ii) 2 applied to annotations (*replace name, retire class*, and *retire property*) were described in the paper by a textual description. The other 46 complex changes do only have a terminology/name to identify them and no further explanation is provided. In addition, from the set of 51 complex changes, 29 are mentioned only in this document, and they all have no textual description.

The approach of Rogozan and Paquette (2009) is a framework composed of one module to track and manage the history of ontology changes, and another one that provides support to maintain the integrity of the ontology-based referencing of resources after the ontology evolution. To the set of 51 complex changes proposed in Klein (2004), this paper adds 16 complex changes to the list of complex changes (7 applied to classes and 9 applied to properties). Some of them are only mentioned in Rogozan and Paquette (2009): 5 applied to classes (*ModifyEquivalentClass, MoveDisjointClass, Modify Union, Modify Complement* and *Modify Intersection*) and 5 applied to properties (*Modify Object Property Range, Modify Datatype Property Value, Modify Data Type Property Domain, ModifyEquivalentProperty* and *ModifyInverseProperty*). No one has a textual description. To this set of complex changes, we added 6 complex changes applied to classes that do only appear in Herrmannsdoerfer *et al.* (2010), namely (*Fold Class*, *Unfold Class*, *Inline Superclass*, *Fold SuperClass*, *Unfold SuperClass* and *Inline Subclass*).

In Gröner *et al.* (2002), the authors tackle the problem of refactoring recognition using description logics (DL) reasoning to compare different versions of an OWL-DL ontology semantically. This paper adds to the set of complex changes found 12 changes: 6 applied to classes and 6 applied to properties. From the set of 12 changes, 2 applied to classes (*extract class* and *inline class*) and 3 applied to properties (*Unidirectional to bidirectional*, *Bidirectional to unidirectional*, and *change cardinality*) are only defined in this paper. No one has a textual description.

The document Rahnama and Barforoush (2015) proposes an ontology evolution process model that uses ontology change rules to represent the consistency of the ontology. This document was considered since it only provides complex changes applied to axioms (2 changes). It adds to the set of complex changes applied to classes already considered by the previous documents 1 change (*group classes*), to the set of changes applied to properties 2 changes (*merge property* and *split property*), and to the set of changes applied to axioms 2 changes (*Axiom Split* and *Axiom Join*). All other changes mentioned in the document were also mentioned in other documents.

In Kondylakis (2010), the authors propose an approach that gets as input different ontology versions and the original user query and rewrites the query over data integration systems that use different ontology versions. It can automatically identify the changes among the ontology versions. All complex

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

changes mentioned in the document have been formalized. The document mentions 2 complex changes applied to axioms, 10 applied to classes, and 17 applied to properties. The ones not mentioned in the previsions documents are: three applied to classes (*Ungroup classes, Move class to different hierarchy* and *Split into existing*); and seven applied to properties (*Merge Properties into Existing, Split Properties into Existing, Pull up in the Property Hierarchy, Pull down in the Property Hierarchy, Move Properties to a Diff. Property Hierarchy, Group Property* and *Ungroup Property*); and one applied to axioms (*Change Comment*). Note that *Move Properties to a Diff. Property Hierarchy* is only defined in this paper.

COnto-Diff, presented in Hartung *et al.* (2013), find out the diff evolution mapping between given versions of an ontology. It initially finds out basic changes and then adopts a rule-based approach to transform the basic change operations into a smaller set of more complex change operations. The document mentions 10 changes applied to classes (but only 3 on them (*Substitute concept c1 by c2, Add Leaf* and *Delete Leaf*) have not appeared in the documents mentioned above) and 2 applied to annotations.

The approach presented in Stojanovic *et al.* (2002) identifies an evolution process and provides the user with capabilities to control and customize it. They use an evolution strategy to encapsulate policy for evolution concerning user's requirements. The document indicates six complex changes applied to classes and five applied to properties. Three of ones applied classes (*Extract related concept, Shallow concept copy*, and *Deep concept copy*) and two applied to properties (*Shallow property copy* and *Deep property copy*) have not yet been considered in the set of complex changes by the previous documents.

In Liu *et al.* (2014), the authors propose a new calculus named SetPi to model ontology evolution. It extends the classical Pi calculus with the idea of 'set', creating new channels in the system. This document mentions 6 changes applied to classes, but only one was not mentioned in the already described documents, *Add generalization class*.

7. Threats to validity

In this section, we present threats to the validity of this work. They are related to the literature review and the mappings we defined.

Limited set of digital libraries We considered only four digital libraries: *ACM*, IEEE, *Springer* and *Elsevier*. Although this number may seem small, these four digital libraries are well known by the community and they publish papers from the majority of and most relevant conferences and journals in the area. Moreover, to check if we were missing any important digital library, we have used *DBLP* and *Google Scholar* to search for documents related to the topic of this work that could appear in other libraries. After a brief analysis of the titles and publishers of the documents returned in *DBLP*, we selected 39 papers for further investigation. In *Google Scholar* we considered the first 100 unique documents returned, and excluding the ones returned in the digital libraries searches, no other digital library accounted for more than 5% of the documents.

Search string too flexible By using the search string presented in Section 2 for searching in *ACM* (the first digital library we looked at for), the search returned 229 papers, but only 13 papers mentioned complex changes applied to ontologies. Since the search returned many papers that do not mention any complex change applied to ontology, we could have narrowed down the search string. However, we decided not to do so since we were afraid that using a narrower search string could imply missing any relevant document. The flexible search string could have been a problem when restricting the search's result in *Google Scholar* to the first 100 documents. However, 62% of the documents returned when searching *Google Scholar* mentioned complex changes applied to ontologies. Thus, we considered the search string adequate.

Low Mapping Confidence The majority of complex changes mentioned in the literature only have a textual description. Since textual descriptions are sometimes ambiguous, some of the mappings we have created based on them and the terminology used to name the changes might be incorrect. Also, other mappings were created based solely on the terminologies because some of the changes were not

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

formalized and did not have any textual description. In such circumstances, the confidence in the mappings was low. Thus, we illustrated the mappings in figures of the Appendices with low confidence, represented by a dashed line.

Relevance of complex changes From the total of 112 complex changes included in the unified list of complex changes, about 43% of the changes are mentioned in only one document. In those cases, it might indicate that these changes are not relevant or were not faced in the use cases considered by the other approaches. This paper's objective was to provide a unified list of all complex changes found in the literature applied to OWL and RDFS ontologies and not assess the relevance of each complex change included in the list.

Exclusion of Metaclasses and Metaproperties We have decided to exclude changes applied to metaclasses and metaproperties from the unified list of complex changes since they are only available in OWL 2[13] by the punning feature. Such changes were only mentioned in Kondylakis (2010) and in Papavasileiou *et al*. (2013), which repeats all the changes mentioned in Kondylakis (2010).

8. Open directions

In this section, we present some open directions in the context of using the unified list of complex changes that were not (entirely) addressed by the literature to the best of our knowledge.

- *Definition of complex changes.* There is a need to provide a textual description and a formalization of all complex changes. Some of the complex changes proposed in the literature and included in the unified list do not formalize, and others do not even have a textual description. These formalizations and textual descriptions are fundamental to understanding the semantics of the complex changes and, consequently, understanding the ontology evolution.
- *Motivation for the evolution and complex changes.* Knowing that the evolution of an ontology, triggered by a set of facts, implies changing the ontology, it would be beneficial to be able to associate sets of facts with sets of complex changes. It would help understand the triggers for the evolution of an ontology by only analyzing the delta between two versions. Therefore, there is a need to investigate the evolution of ontologies in different domains to find such associations.
- *Metrics and complex changes.* Some approaches describe a set of metrics to measure the quality of an ontology (Lozano-Tello & Gomez-Perez, 2004; Burton-Jones *et al*., 2005; Lantow, 2016). Nowadays, to figure out if the quality of an ontology improved or not after its evolution we need to measure both versions. If we were able to understand which complex changes increase, reduce or have no effect on each quality metric, one would only need to have the set of complex changes to figure out if the quality was improved or not. The set of complex changes would help identify the affected metrics and how they were affected and, thus, would help to check if the quality of the ontology has improved.
- *Domains and complex changes.* There may be the case that the same set of complex changes usually occurs when evolving different ontologies of a given domain. If this hypothesis is confirmed, we will be able to figure out the more suitable evolutions for such a domain. This information would help design ontologies and artifacts more adaptable to the given changes.

9. Conclusion

Investigating and proposing solutions to ontology evolution is evident when considering the numerous books, conferences, and journals dedicated to the topic. Although most documents mentioning complex changes were published more than five years ago, we still have limited technical support. Isolated tools can (semi-)automatically find complex changes between two versions of the same ontology (Tudorache *et al.*, 2013; Lara *et al.*, 2017). Some of them can provide additional information to the user besides

[13] http://www.w3.org/TR/owl2-new-features/.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

the list of changes (Kondylakis & Papadakis, 2018). Some tools can (semi-)automatically adapt related artifacts according to such changes (Davidovsky *et al.*, 2011; Kondylakis & Plexousakis, 2012; Zablith *et al.*, 2015).

Identifying complex changes between two versions of the same ontology is the first step in dealing with the evolution of the knowledge-based system. It is fundamental for adapting related artifacts (such as the adaptation of instances and queries created based upon the classes, relationships, and axioms described in the ontology). For example, without being able to detect when a *class rename* occurs, a tool would recognize that 'a class was deleted' and 'a new class was created', and we would lose all instances of the 'old class'. Without being able to detect when a *pull up class* occurs, we would need to review all queries involving such class, even those mentioning properties of superclasses that remain superclasses of the moved class.

In this paper, we presented the literature review we have conducted to determine the complex changes applied to OWL and RDFS ontologies that have been mentioned in the literature. The contributions of this paper are:

Unified list of complex changes The unified list of complex changes depicted in Tables 1, 2 and 3, plus the two changes applied to axioms (*axiom split* and *axiom joint*) may be used by the reader as a reference of complex changes published in the literature and the documents that mention each change. We strongly believe that the unification and classification of the complex changes will help develop different tools to deal with ontology evolution since they provide a unified vocabulary to refer to the changes.

Mappings among the complex changes As mentioned in Section 5.1, we noticed that different terminologies mentioned in various documents are used to describe the same change, and the same terminology might be used to refer to different changes. Therefore, the mappings represented in the tables mentioned above and detailed in Appendices 1, 2, 3, and 4 are themselves significant contributions since they provide a mechanism for matching and comparing the different approaches that propose complex changes. Using the mappings, the reader can find the similarities and differences between the complex changes mentioned in the approaches. Besides, those mappings also help compare different tools able to detect complex changes since they indicate the similarities between the changes they found.

Complex changes classification The classifications guide the definition of heuristics to find out complex changes since they help understand that different complex changes share similar intentions and that some may be specializations of others. The understanding that some of the changes share similar aims and others are in the same hierarchy of entities helps their reuse and extension of algorithms. The algorithms would be able to detect complex changes and explain the changes since more abstract ones can be omitted once a more specific change is found.

Analyses of the complex changes and documents Besides enumerating the complex changes we have found in the literature, we made several analyses related to the documents and the characteristics of the changes mentioned by them (Section 3). We have identified: (i) the documents that mention changes applied to each element kind (that may be used by readers that want to investigate only changes applied to classes, for instance), (ii) the documents that mention the same (or similar) sets of complex changes (used to understand the community and figure out the groups interested in the same topic), (iii) the most and least mentioned changes (used, for instance, when implementing tools able to check for the most relevant changes between two versions of the same ontology), (iv) the documents that present a textual description or formalization (used to separate the approaches that provide more explanation about the changes they consider), and (v) the changes that have no formalization or textual description.

Most relevant documents mentioned complex changes In this paper, we identify the set of 8 documents that can be used as a reference for the complex changes included in the unified list.

List of Open Directions After investigating several documents mentioning complex changes, we were able to understand the state-of-the-art, figure out its missing gaps, and propose some future directions in the area.

Conflicts of interest. The author(s) declare none.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Supplementary material. To view supplementary material for this article, please visit http://doi.org/10.1017/S0269888922000066.

References

Burton-Jones, A., Storey, V. C., Sugumaran, V. & Ahluwalia, P. 2005. A semiotic metrics suite for assessing the quality of ontologies. *Data and Knowledge Engineering*. North-Holland **55**(1), 84–102.

Davidovsky, M.,Ermolayev, V. & Tolok, V. 2011. Instance migration between ontologies having structural differences. *International Journal on Artificial Intelligence Tools* **20**(06), 1127–1156.

Dinh, D., Dos Reis, J. C., Pruski, C., Da Silveira, M. & Reynaud-Delaître, C. 2014. Identifying relevant concept attributes to support map-ping maintenance under ontology evolution. *Web Semantics: Science, Services and Agents on the World Wide Web* **29**, 53–66.

Dos Reis, J. C., Da Silveira, M., Dinh, D., Pruski, C. & Reynaud-Delaître, C. 2014. Requirements for implementing mapping adaptation systems. In *IEEE 23rd International WETICE Conference*, 405–410. IEEE.

Dos Reis, J. C., Dinh, D., Pruski, C., Da Silveira, M. & Reynaud-Delaîtrec. 2013. Mapping adaptation actions for the automatic reconciliation of dynamic ontologies. In *22nd ACM international conference on Information and Knowledge Management*, 599–608. ACM.

Djedidi, R. & Aufaure, M.-A. 2010. Onto-evoal an ontology evolution approach guided by pattern modeling and quality evaluation. In *International Symposium on Foundations of Information and Knowledge Systems*, 286–305. Springer.

Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D. & Antoniou, G. 2008. Ontology change: classification and survey. *The Knowledge Engineering Review* **23**(2), 117–152.

Galani, T., Papastefanatos, G. & Stavrakas, Y. 2016. A language for defining and detecting interrelated complex changes on RDF(S) knowledge bases. In *ICEIS*, 472–481.

Galani, T., Stavrakas, Y., Papstefanatos, G. & Flouris G. 2015. Supporting cOmplex changes in RDF(S) knowledge bases. In *DIACRON@ESWC*, 28–33.

Gröner, G., Parreiras, F. S. & Staab, S. 2010. Semantic recognition of ontology refactoring. In *International Semantic Web Conference*, 273–288. Springer.

Hartung, M., Groß, A. & Rahm, E. 2010. Rule-based generation of diff evolution mappings between ontology versions, ArXiv, vol.abs/1010.0122.

Hartung, M., Groß, A., & Rahm, E. 2013. COnto–diff: generation of complex evolution mappings for life science ontologies. *Journal of Biomedical Informatics* **46**(1), 15–32.

Herrmannsdoerfer, M., Vermolen, S. & Washsmuth, G. 2001. An extensive catalog of operators for the coupled evolution of metamodels and models. In *SLE*, LNCS **6563**, 163–182. Springer.

Javed, M., Abgaz, Y. M. & Pahl, C. 2012. Composite ontology change operators and their customizable evolution strategies. In *CEUR*.

Javed, M., Abgaz, Y. M. & Pahl, C. 2013. Ontology change management and identification of change patterns. *Journal on Data Semantics* **2**(2–3), 119–143.

Kalibatiene, D. & Vasilecas, O. 2011. Survey on ontology languages. In *Perspectives in Business Informatics Research.* Lecture Notes in Business Information Processing, **90**, 124–141. Springer.

Kirsten, T., Gross, A., Hartung, M. & Rahm, E. 2011. Gomma: acomponent-based infrastructure for managing and analyzing life science ontologies and their evolution. *Journal of Biomedical Semantics* **2**(1), 6.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J. & Linkman, S. 2009. Systematic literature reviews in software engineering - a systematic literature review. *Information and Software Technology* **51**(1), 7–15.

Khattak, A., Batool, R., Pervez, Z., Khan, A. & Lee, S. 2013. Ontology evolution and challenges. *Journal on Information Science and Engineering* **29**(5), 851–871.

Khattak, A. M., Latif, K., Khan, S. & Ahmed, N. 2008. Ontology recovery and visualization. In *4th International Conference on Next Generation Web Services Practices*, 90–96. IEEE.

Khattak, A. M., Latif, K. & Lee, S. 2013. Change management in evolving web ontologies. *Knowledge-Based Systems* **37**, 1–18.

Khelladi, D., Hebig, R., Bendraou, R., Robin, J. & Gervais, M. 2015. Detecting complex changes during metamodel evolution. In *CAiSE*, 263–278.

Klein, M. & Noy, N. 2003. A component-based framework for ontology evolution. *In Workshop on Ontologies and Distributed Systems at IJCAI*, **3**, 4–12.

Klein, M. C. A. 2004. *Change Management for Distributed Ontologies*. PhD dissertation, Vrije Universiteit Amsterdam.

Kondylakis, H. 2010. *Ontology Evolution in Data Integration*. PhD dissertation, University of Crete.

Kondylakis, H. & Papadakis, N. 2018. EvoRDF: evolving the exploration of ontology evolution. *The Knowledge Engineering Review* **33**, e.12.

Kondylakis, H. & Plexousakis, D. 2012. Ontology evolution: assisting query migration. In *Conceptual Modeling*, **7532**, 331–344.

Lantow, B. 2016. OntoMetrics: application of on-line ontology metric calculation. In *BIR Workshops*.

Lara, A., Henriques, P. R. & Gancarski, A. L. 2017. Visualization of ontology evolution using ontodiff graph. In *6th Symposium on Languages, Applications and Technologies (SLATE 2017)*. SchlossDagstuhl-Leibniz-Zentrum fuer Informatik.

Lambrix, P., Dragistic, Z., Ivanova, V. & Anslow, C. 2016. Visualization for ontology envolution, In *VOILA@ISWC*, 54–67.

Liu, L., Zhang, P., Fan, R., Zhang, R., & Yang, H. 2014. Modeling ontology evolution with Setpi. *Information Sciences* **255**, 155–169.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press

Lozano-Tello, A. & Gomez-Perez, A. 2004. ONTOMETRIC: a method to choose the appropriate ontology. *Journal of Database Management* **15**, 1–18.

Maedche, A., Motik, B., Stojanovic, L., Studer, R. & Volz, R. 2002. Managing multiple ontologies and ontology evolution in ontologging. In *International Conference on Intelligent Information Processing*, 51–63. Springer.

Mahfoudh, M., Forestier, G., Thiry, L. & Hassenforder, M. 2015. Algebraic graph transformations for formalizing ontology changes an devolving ontologies. *Knowledge-Based Systems* **73**, 212–226.

Najla, S., Wassim, J. & Faiez, G. 2009. Extension of protege to support evolution of ontology. In *First International Conference on Advances in Databases, Knowledge, and Data Applications*, 149–154. IEEE.

Noy, N. & Klein, M. 2004. Ontology evolution: not the same as schema evolution. *Knowledge and Information Systems* **6**(4), 428–440.

Noy, N., Kunnatur, S., Klein, M. & Musen, M. 2004. Tracking changes during ontology evolution. In *ISWC*, 259–273.

Palma, R., Haase, P., Corcho, O. & Gomez Perez, A. 2009. Changerepresentation for owl 2 ontologies. In *5th InternationalWorkshop on OWL: Experiences and Directions*. CEUR-WS.

Papavasileiou, V., Flouris, G., Fundulaki, I., Kotzinos, D. & Christophides, V. 2009a. On detecting high-level changes in RDF/S KBs. In *The Semantic Web—ISWC 2009*, **5823**, 473–488.

Papavasileiou, V., Flouris, G., Fundulaki, I., Kotzinos, D. & Christophides, V. 2009b. Formalizing high-level change detection for rdf/s kbs, FORTH-ICS (Technical Report TR-398).

Papavasileiou, V., Flouris, G., Fundulaki, I., Kotzinos, D. & Christophides, V. 2013. High-level change detection in RDF(S) KBs. *ACM Transactions on Database Systems* **38**(1), 1:1–1:42.

Paré, G. & Kitsiou, S. 2017. Methods for literature reviews. In *Handbook of eHealth Evaluation: An Evidence-based Approach*. University of Victoria. https://www.ncbi.nlm.nih.gov/books/NBK481583/.

Pare, G., Trudel, M.-C., Jaana, M. & Kitsiou, S. 2015. Synthesizing in-formation systems knowledge: a typology of literature reviews. *Information and Management* **52**(2), 183–199.

Pittet, P., Cruz, C. & Nicolle, C. 2013. Modeling changes for shoin (d)ontologies: an exhaustive structural model. In *2013 IEEE Seventh International Conference on Semantic Computing*, 104–109. IEEE.

Plessers, P. 2006. *An Approach to Web-based Ontology Evolution*. PhD dissertation, Faculteit van de Wetenschappen.

Rahnama, A. & Barforoush, A. A. 2015. A novel ontology evolution methodology. *Journal of Web Engineering* **14**(3–4), 301–324.

Rogozan, D. & Paquette, G. 2005. Managing ontology changes on the semantic web. In *The 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, 430–433. IEEE.

Rogozan, D. & Paquette, G. 2009. Ontology evolution and the referencing of resources in semantic web context. In *Semantic Web Technologies for e-Learning*.

Stojanovic, L. 2004. *Methods and Tools for Ontology Evolution*. PhD dissertation, Karlsruhe Institute of Technology, Germany.

Stojanovic, L., Maedche, A., Motik, B. & Stojanovic, N. 2002. User-driven ontology evolution management. In *International Conference on Knowledge Engineering and Knowledge Management*, 285–300. Springer.

Stuckenschmidt, H. & Van Harmelen, F. 2005. *Information Sharing on the Semantic Web*. Springer Science and Business Media.

Tang, X. & Yang, F. 2007. A study on dynamic ontology for information integration in e-governmental virtual organization. In *International Conference on Wireless Communications, Networking and Mobile Computing*, 3600–3604. IEEE.

Templier, M. & Pare, G. 2015. A framework for guiding and evaluating literature reviews. *Communications of the Association for Information Systems* **37**(1), 6.

Tudorache, T., Nyulas, C., Noy, N. F. & Musen, M. A. 2013. Webprotege: a collaborative ontology editor and knowledge acquisition tool for the web. *Semantic Web* **4**(1), 89–99.

Xie, C., Jiang, L. & Cai, H. 2011. Instance-driven ontology evolution mechanism towards enterprise data management. In *IEEE 8th International Conference on e-Business Engineering*, 24–30. IEEE.

Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H. & Motta, E. 2015. Ontology evolution: a process-centric survey. *The Knowledge Engineering Review* **30**(1), 45–75.

https://doi.org/10.1017/S0269888922000066 Published online by Cambridge University Press