Pitfalls in Networked and Versioned Ontologies

Omar Qawasmeh[1], Maxime Lefrançois[2], Antoine Zimmermann[2], Pierre Maret[1]

[1] Univ. Lyon, CNRS, Lab. Hubert Curien, UMR 5516, F-42023 Saint-Étienne, France
{omar.alqawasmeh,pierre.maret}@univ-st-etienne.fr
[2] Univ. Lyon, MINES Saint-Étienne, CNRS, Lab. Hubert Curien, UMR 5516, F-42023 Saint-Étienne, France
{maxime.lefrancois,antoine.zimmermann}@emse.fr

Abstract. The listing and automatic detection of ontology pitfalls are crucial in ontology engineering. Existing work mainly focused on detecting pitfalls in stand-alone ontologies. Here, we introduce a new categorization of ontology pitfalls: stand-alone ontology pitfalls, pitfalls in versioned ontologies and, pitfalls in ontology networks. We investigate pitfalls in a situation of ontology co-evolution and we provide a systematic categorization of the different cases that could occur during the co-evolution process over two ontology portals: the Linked Open Vocabulary and BioPortal. We also identify 9 candidate pitfalls that may affect versioned ontologies or ontology networks. We evaluate the importance and potential impact of the candidate pitfalls by means of a web-based survey we conducted in the semantic web community. Participants agreed that listing and investigating ontology pitfalls can effectively enhance the quality of ontologies and affect positively the use of ontologies. Moreover, the participants substantially agreed with the new categorization we proposed. We conclude by providing a set of recommendations to avoid or solve the different pitfalls we identified.

Keywords: Ontology Networks · Ontology Versions · Ontology Pitfalls.

1 Introduction

Ontologies provide a common infrastructure for a specific domain, which leads to a better understanding, sharing and analyzing of the knowledge [10]. However, domain description is subject to changes, thus arises the need to evolve ontologies (i.e. versioning) in order to have an up-to-date representation of the targeted domain [33]. Ontology evolution is the process of maintaining an ontology up to date with respect to the changes that might arise in the described domain, and/or in the requirements [41].

Following good practices during the development of ontologies help to increase their quality, which reflects in their usage [3, 6]. Reusability is considered as a good practice while designing an ontology [30]. On the one hand, reusability saves time for knowledge engineers while developing ontologies, but on the other hand it raises the problem of adapting one's ontology to the evolution of an imported ontology and thus complicates the maintenance process.

Moreover, reusability leads to the creation of connections between different ontologies. Authors in [28] categorized ontologies based on their connections into: 1. *stand-alone ontologies* that have no connection with other ontologies, or 2. *ontology networks:* sets of ontologies that are connected to each other via relationships, such as imports or uses links.

During the development or the usage of ontologies (stand-alone ontologies or ontology networks), there exist some pitfalls which the knowledge engineers can fall in while developing, evolving or maintaining an ontology. These pitfalls may cause abnormal behavior for the related artifacts (e.g. systems that are using the ontology, or other connected ontologies to it). Several researchers worked on observing (e.g. [8, 38]) or listing (e.g. [25]) the set of pitfalls that might affect stand-alone ontologies.

In a previous contribution [27], we investigated the set of pitfalls that are targeted to a specific case of ontology networks, that we name ontology co-evolution (i.e. the evolution of an ontology O that uses terms having the namespace of another ontology O'). We provided an exhaustive categorization of the different cases that could occur for this situation. We observed 74 cases of co-evolution involving 28 different ontologies in the Linked Open Vocabulary (LOV), and 14 cases of co-evolution involving 10 different ontologies in BioPortal. We concluded the paper by listing a set of good practices, bad practices and uncertain practices that could happen within ontologies that use some terms from other ontologies.

In this paper we extend our previous contribution [27] and the current pitfall analysis [8, 38, 25] by observing and listing the set of pitfalls that can affect versioned ontologies or ontology networks. We propose to distinguish between three types of pitfalls:

1. Stand-alone ontology pitfalls. These pitfalls are addressed by [8, 38, 25].
2. Versioned ontologies pitfalls (i.e. when an ontology O evolves from v_1 to v_2).
3. Ontology networks pitfalls (i.e. when an ontology O imports a different ontology O').

Moreover, we assess the importance and potential impact of these pitfalls over ontology networks and versioned ontologies.

The rest of the paper is organized as follows: Section 2 presents our motivating scenario. Section 3 presents an overview of our research methodology and the related definitions we propose. Section 4 lists our previous contributions at [27], where we presented a theoretical analysis of the need for changes that could stem from the evolution of an imported ontology, in addition to an exhaustive theoretical analysis of how an ontology may be adapted to such evolution. In Section 5 we introduce a catalogue of pitfalls that could hamper versioned ontologies and ontology networks. Section 6 presents the survey we distributed to the semantic web community in order to measure the importance and potential impact for the candidate pitfalls. Section 7 discusses and concludes the paper.

2 Motivating Scenario

Let Amal be a knowledge engineer who develops a child care domain ontology called *Childcare*. In the version v1.1 of *Childcare*, created in May 2017, Amal used a specific term `programmOfStudy` from another ontology called *Education* created in January 2017 (Figure 1). *Childcare* contains at least a link to a term of *Education*. This creates a two ontologies network. In September 2017 the creators of the *Education* ontology released version v1.2. Amal does not notice the evolution. Thus, she thinks that her ontology is still using v1.1 version of the *Education* ontology. Inside this simple ontology network, several issues might arise:

- The term `programmOfStudy` was removed from *Education*, however it is still used in *Childcare*. This has an impact over *Childcare*. As a consequence of this impact, Amal should adapt her ontology.
- New terms were introduced in *Education,v1.2* (e.g. `boarding school`). Amal should be made aware of these new terms in order to possibly make use of them in her ontology.

After noticing the evolution of the *Education* ontology, Amal created v1.2 *Childcare* ontology in November 2017. During this versioning, several issues might arise, such as:

- The v1.1 of *Childcare* ontology is not accessible any more by its IRI. This pitfall is caused by Amal, and she is the responsible of maintaining the *Childcare* ontology.
- Let us assume that the v1.2 of the *Education* ontology is inconsistent, importing this ontology by the *Childcare v1.2* will make it become inconsistent too. This versioned ontology pitfall is caused by the owners of the *Education* ontology, and it is their responsibility to maintain their ontology.

If Amal publishes a bigger network of ontologies, the connections between these ontologies are expanding, which makes it vulnerable to falling into some pitfalls. For example:

- If Amal presents an inconsistent ontology and she published it. Any other ontology or system (e.g. question answering) that use this ontology might become inconsistent too.

In the next section, we present an overview about ontology evolution, and such approaches that took care of same problem we are targeting in this paper.

3 Research Overview

This section presents the following: Section 3.1 presents an overview of ontology evolution. Section 3.2 presents our definition of an ontology network. Section 3.3 presents an overview of ontology pitfalls, and our new categorization for the set of pitfalls.

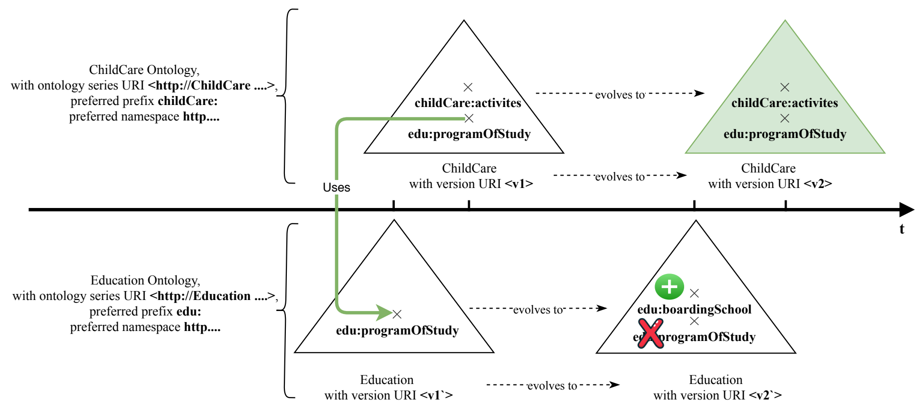

Fig. 1: An illustrative figure for the motivating example

3.1 An overview of ontology evolution

Ontology evolution is the process of maintaining an ontology up to date with respect to the changes that might arise in the described domain, and/or in the requirements [41]. Zalbith et al. [41] studied the different methodologies and approaches to evolve ontologies, and defined a comprehensive life-cycle of ontology evolution: 1. Detect the need for evolution, 2. Suggest changes to evolve the ontology, 3. Validate the suggested changes, 4. Assess and study the impact of the evolution on external artifacts that rely on the ontology (e.g. other ontologies, systems), 5. Keep track of the implementation of the changes.

We investigated [27] in the first and fourth phases and we introduced two definitions to detect the need of evolution, and to assess the impact of the evolution of two connected ontologies.

Different approaches target *Phase 1. Detecting the need for evolution.* In [32], the author proposed two techniques to detect the need for the evolution: 1. Detect the need of the evolution by studying the ontology instances using data mining techniques. 2. Detect the need of the evolution by observing the structural changes inside an ontology.

For instance, the following approaches are following the first technique: In [40], author propose a technique to detect the need for evolution, by comparing the concepts of the targeted ontology with external data sources (e.g. text documents, databases), and they suggest new concepts based on the external data sources. Castano et al. [5] detect from external data sources the need for ontology evolution. Their approach detects whether the ontology needs to be enriched in case of missing concepts to describe a new multimedia resource.

The following approaches are following the second technique: Authors in [35] and [21] agree that ontology evolution is caused mainly by three reasons: 1. Changes in the described domain. 2. Changes in the conceptualization (e.g. deletion and addition). 3. Changes in the explicit specification. In [23] a change detection algorithm is proposed which relies on a specific language they also proposed. One feature of their algorithm is to detect the need of evolution out of the changes that happen, such as renaming a class (i.e. delete and add).

As for *Phase 4. Assessing the impact of ontology evolution* of [41], different approaches were proposed to study the impact of ontology evolution in different techniques. For instance, Dragoni and Ghidini [7] studied how ontology evolution affects research systems. They observed three operations that could happen during the evolution process of an ontology: 1. rename a concept, 2. delete a concept, and 3. move a concept. They applied 75 queries over a search system for every version of the evolved ontology, they compared the effectiveness of the search system with a baseline.

Abgaz et al. [2] analyzed both structural and semantic impact over ontologies. They predefined a bag of rules to study the impact by detecting undesirable statements and wrong instances. They defined 10 change operations that cover the different change scenarios.

Groß et al. [9] studied how some statistical artifacts are affected by the evolution of the *Gene* ontology [3]. They used CODEX tool [13] to detect the changes (e.g. addition, merging, moving). They created a stability measure by choosing a fixed set of genes to compute the experimental result set at different point of time with freely chosen ontology and annotation versions.

Mihindukulasooriya et al. [20] investigated how *DBpedia* [19], *Schema.org* [11], *PROV-O* [18] and *FOAF* [4] ontologies evolved within time. They observed the changes between the different versions such as, addition and deletion of classes, properties, sub-classes and sub-properties. They show that the process of ontology evolution relies on the size of the ontology, and it becomes more challenging when the ontology size is large. They conclude by showing the need of creating tools that can help during the evolution process.

Abdel-Qader et al. [1] analyzed the impact of the evolution of terms in 18 different ontologies referenced in LOV. Their method consisted of two phases: 1. retrieve all the ontologies that have more than one version, and 2. investigate how terms are changed and adopted in the evolving ontologies. They applied their analysis on three large-scale knowledge graphs: DyLDO[4], BTC[5] and Wikidata.[6] They found that some of the term changes in the 18 ontologies are not mapped into the three knowledge graphs. Also they concluded that there is a need for a service to monitor the ontology changes, which will help to maintain the external artifacts (other ontologies, systems or data sets).

3.2 Ontology networks

This topic has not been widely studied yet. The term "ontology network" (a.k.a. networked ontologies) is informally defined by [29, 34, 12] as the set of ontologies that are connected to each other via a variety of relationships (e.g. owl:imports,

[3] `http://geneontology.org/`
[4] `http://km.aifb.kit.edu/projects/dyldo/data`
[5] `https://km.aifb.kit.edu/projects/btc-2012`
[6] `https://www.wikidata.org`

modularization, version). Authors in [26] studied 18,589 terms appearing in 196 ontologies, and they concluded that *Uses* and *Imports* are the main relationships between ontologies. Hence, in this paper we propose a formal definition of an ontology network as:

Definition 1. ***An ontology network*** *An ontology network is a directed graph* $G = (\mathcal{N}, \mathcal{E})$*, consisting of a set* $\mathcal{N}$ *of ontologies and a set* $\mathcal{E}$ *of relationships, which are ordered pairs of elements of* $\mathcal{N}$*. Furthermore, every ontology* $O \in \mathcal{N}$ *has an owner* $\mathrm{author}(O)$*, an IRI* $\mathrm{iri}(O) \in \mathrm{IRI}$*, an ontology series IRI* $\mathrm{series\_iri}(O) \in \mathrm{IRI}$*, a namespace* $\mathrm{ns}(O) \in \mathrm{IRI}$*, and a publication date* $\mathrm{date}(O) \in \mathbb{N}$*; Every ontology relationship* $e \in \mathcal{E}$ *is labeled by a non-empty set of types* $\mathrm{type}(e) \in \mathcal{T}$*.*

We consider only two types of relationships between the different ontologies (regardless of the owner). $\mathcal{T} = \{uses, imports\}$:

uses *uses* $\in \mathrm{types}\langle O, O' \rangle$ happens when an ontology O uses a term t (that is, an IRI denoting an individual, a class or a property) that has the namespace of a different ontology O'.

imports *imports* $\in \mathrm{types}\langle O, O' \rangle$ happens when an ontology O imports another ontology O', using the OWL importing mechanism.[7]

3.3 Ontology pitfalls

In the field of semantic web, several researchers used the term “pitfall” to refer to the set of mistakes/errors that can be made during the development or usage of ontologies. In this research, we propose to distinguish between three types of pitfalls:

1. *Stand-alone ontology pitfalls:* can happen within a single ontology O that is created by $author(O)$ (e.g. *Childcare V1.1* from Figure 1).
2. *Versioned ontologies pitfalls:* can happen when an $author(O)$ creates/publishes a new version of the ontology O (e.g. the evolution of *Education* ontology from Figure 1).
3. *Ontology network pitfalls* can happen within the set of ontologies that are connected to each other, such as when an ontology O is connected to a different ontology O' (e.g. both of the ontologies *Childcare* and *Education* from Figure 1). The responsible of resolving these pitfalls is either $author(O)$ or $author(O')$, depending on if the pitfall occur in O or O'.

Existing work observed and listed the set of pitfalls in different scenarios. Sabou and Fernandez [28] provide methodological guidelines for evaluating both stand-alone ontologies and ontology networks. Their methodology relies on selecting a targeted ontology component to evaluate based on a predefined goal.

[7] https://www.w3.org/TR/owl2-syntax/, sections 3.4

Poveda et al. [25] gathered 41 stand-alone ontology pitfalls from different sources in a catalogue [8] and categorized them based on the structural (i.e. syntax and formal semantics), functional (i.e. the usage of a given ontology) and usability (i.e. the communication context of an ontology) dimensions. In addition, they tag each pitfall with it's importance level (i.e. critical, important, or minor).

Gaudt and Dessimoz [8] analyzed annotation pitfalls that exist in the GO-basic ontology.[9] The authors summarized the set of pitfalls (e.g. Annotator Bias and Authorship Bias) and provided good practices to help solving them. They showed how these pitfalls might introduce problems when the data is used in other tasks.

As a conclusion, we see that current research take care of listing or observing pitfalls for stand-alone ontologies, and there is a lack of research papers that observe and list the set of pitfalls that might affect versioned ontologies and ontology networks. In our research we observe and list the set of pitfalls that are related to versioned ontologies and ontology networks.

4 Observing the Impact and Adaptation to the Evolution of an Imported Ontology

In this section we update our findings from [27], where we introduced two situations related to observe the impact and the adaptation to the evolution of an imported ontology. In Section 4.1 we target the evolution of an imported ontology (if ontology O uses some terms t from another ontology O', and then O' evolves). In Section 4.2 we target the adaptation to the evolution of the imported ontology.

4.1 Observing the Impact of the Evolution of an Imported Ontology

As mentioned in Section 3.1, there is a need to detect when to perform changes on ontologies (i.e. Phase 1 from the ontology evolution life-cycle). Two behaviors can be distinguished:

1. There was already a problem: an ontology O uses a term t that has the namespace of another ontology O', however it is not defined in O'.
2. A problem has occurred because of the evolution process: Let's assume that there is an ontology O that uses a term t that has the namespace of another ontology O'. O' evolved which causes the deletion of t. This evolution might cause problems for O. This raises the need to evolve O in order to reflect the changes.

[8] Last check January 2020, can be found here: http://oops.linkeddata.es/catalogue.jsp

[9] http://geneontology.org/docs/download-ontology/

To represent formally these two situations, in [27] we have defined a situation that can be used to detect the need of the evolution, mainly named "imported ontology evolution":

Definition 2. ***Imported ontology evolution***
Imported ontology evolution is a situation where: O is an ontology which has at least one version v_1. O' is a different ontology which has at least two versions v'_1 and v'_2. O uses terms [10] *that have the namespace of O'. $time(v)$ is the creation time for a version.*

A case of imported ontology evolution is noted $\langle v_1, v'_1, v'_2 \rangle$ and holds when the following conditions are satisfied:
$time(v'_1) < time(v'_2) \wedge time(v'_1) < time(v_1)$

To illustrate this definition, Figure 2 presents a real life example of one case of imported ontology evolution, where *Music* ontology has one version (v_1: mo_2010-11-28) that uses some terms that have the namespace of the *BIO* ontology (v'_1: bio_2010-04-20, v'_2: bio_2011-06-14). Table 1 lists the different cases that may occur with respect to Definition 2. t is a term that has the namespace of O'. Each circle represents the set of terms terms (t) that exist in the different versions of the two ontologies (i.e. v'_1, v'_2, and v_1). Four possible cases might happen:

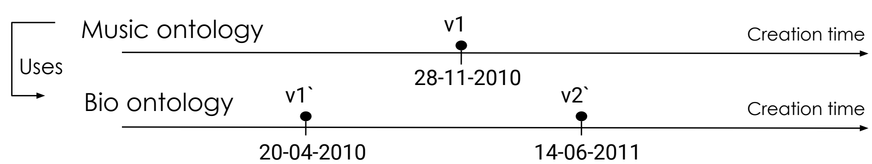

Fig. 2: A time line showing the creation times of the Music ontology and the BIO ontology, where the Music ontology uses terms that are defined by BIO [27]

Row 1. No changes over t

Case 1.a There is no change of t to detect, therefore there is no interest in studying this case. This case holds for all the terms t with the namespace of O', that are neither defined in O' nor used in O

Case 1.b This case holds when O uses a term t with the namespace of O', but that is not defined in O'. Some terms that have the namespace of O' are being used in v_1 without being defined before. This is a mistake, hence there is a need to evolve v_1 to reflect the latest changes.

[10] A RDF term is generally defined as: $IRI \cup Blanknodes \cup Literals$. In this research we take into consideration only the IRIs.

Row 2. t **is deleted in** v'_2
The owners of O' decided to stop using a term (e.g. `programmOfStudy`) in v'_2:

Case 2.a The term is not used in v_1. No problems to be reported, and v_1 was not affected by the evolution of O'.
Case 2.b During the evolution, the term t was deleted. However, it is still being used in v_1. This might introduce a problem of using terms that does not exist anymore. So v_1 should evolve to better reflect the changes of O'.

Row 3. t **exist in both** v'_1 **and** v'_2
There is no changes on t:

Case 3.a The term is not used in v_1. However, it can be recommended for use in the upcoming versions of v_1.
Case 3.b No changes over the terms during the evolution.This case is not problematic.

Row 4. t **is added to** v'_2
The owners of O' introduced a new term (e.g. `boardingSchool`) in v'_2:

Case 4.a The term t is not used in v_1. It can be interesting to use, thus this addition can be notified.
Case 4.b The term t is used in v_1, however it was defined later in v'_2.

So far, we have seen the definition and impacts of imported ontology evolution. As a consequence of this evolution, the impacted ontology should be evolved accordingly to better reflect the changes. This creates a situation which we call ontology co-evolution. It will be discussed in the next subsection.

4.2 Observing the Adaptation to the Evolution of an Imported Ontology

The term “ontology co-evolution” has been already used in three research papers. Authors in [16, 15] define the co-evolution as the integration between the database schemes and ontologies to design and evolve the targeted ontologies. Also [22] defines the co-evolution as the creation of ontologies by taking advantage of natural language techniques to process some raw text. These definitions are irrelevant to the problem we investigate. In [27] we have defined ontology co-evolution as:

Definition 3. ***Ontology Co-Evolution***
Ontology co-evolution is a situation where: O is an ontology which has at least two versions v_1 and v_2. O' is a different ontology which has at least two versions v'_1 and v'_2. O uses terms that have the namespace of O'. $time(v)$ is the creation time for a version.

Table 1: The set of cases that might happen during the evolution of O' considering a term t that has the namespace of O' [27]

	uses O O`	a: v_1	b: v_1 t
1	v_1` $v_2$`	Case 1.a: No changes occurred	Case 1.b: Term is used in v_1 without being defined in O`
2	v_1` $v_2$` t	Case 2.a: No impact occurred	Case 2.b: There is a need for evolution, because the term is no longer in O`
3	v_1` $v_2$` t	Case 3.a: No impact occurred. Suggest to add new terms	Case 3.b: No impact occurred
4	v_1` $v_2$` t	Case 4.a: Suggest to add new terms	Case 4.b: Term used before it is defined

In order to have a co-evolution case between O and O' with the ontologies $\langle v_1, v_1', v_2, v_2' \rangle$, the following condition must be satisfied:
$time(v_1) < time(v_2) \wedge time(v_1') < time(v_2') \wedge time(v_1') < time(v_1) \wedge time(v_2') < time(v_2)$

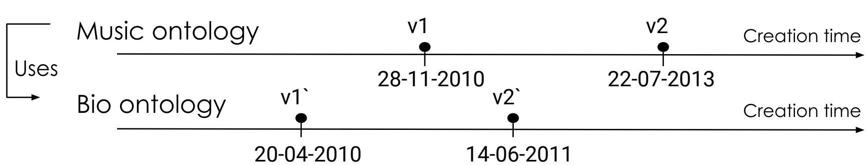

Fig. 3: A time line showing the creation times of the music ontology (mo) and the bio ontology (bio), where mo uses terms that are defined by bio [27]

To illustrate this definition, Figure 3 extends the previous example to show the case of ontology co-evolution: the *Music* ontology has two versions (v_1: mo_2010-11-28, v_2: mo_2013-07-22) that are respectively using the two versions of the *Bio* ontology (v_1': bio_2010-04-20, v_2': bio_2011-06-14).

During the evolution of O', terms may be introduced or deleted. We exhaustively identify the occurrences of adaptation to ontology evolution of O and O' (i.e. co-evolution). We observe the set of terms that have the namespace of O'.

Table 2 shows the different cases that may occur. t is a term that has the namespace of O'. Each circle represents the set of terms (t) that exist in the different versions of the two ontologies (i.e. v_1', v_2', v_1, ad v_2).

Table 2: The set of cases that might happen during the ontology co-evolution considering t that has the namespace of O' [27]

		a	b	c	d
	O uses O`	v_1 v_2	v_1 t v_2	v_1 t v_2	v_1 t v_2
1	v_1\` $v_2$\`	Case 1.a: No changes occurred	Case 1.b: Term is used in v_1 but doesn't exist in O\`	Case 1.c: Term is used in both v_1 and v_2 but doesn't exist O\`	Case 1.d: Term is used in v_2 but doesn't exist O\`
2	v_1\` $t$ $v_2$\`	Case 2.a: Term is deleted from v_2\` and not used in O	Case 2.b: Term is deleted in v_2\` but still used in v_1	Case 2.c: Term is deleted in v_2\` but still used in both v_1 and v_2	Case 2.d: Term is deleted in v_2\` and still used in v_2
3	v_1\` $t$ $v_2$\`	Case 3.a: Term exists in both v_1\` and $v_2$\` and not used in O	Case 3.b: Term exists in both v_1\` and $v_2$\` and used in v_1	Case 3.c: Term exists in both v_1\` and $v_2$\` and used in both v_1 and v_2	Case 3.d: Term exists in both v_1\` and $v_2$\` and used in v_2
4	v_1\` $t$ $v_2$\`	Case 4.a: Term introduced in v_2\` and not used in O	Case 4.b: Term exists in v_2\` and used in v_1	Case 4.c: Term exists in v_2\` and used in both v_1 and v_2	Case 4.d: Term exists in v_2\` and used in v_2

In our first example of Section 2, let us assume that Amal finally noticed the evolution of *Education* ontology and decided to evolve her ontology *Childcare* to V1.2 on November 2017. Based on our definition, the ontology *Childcare* is considered as O which has two versions v_1: *Childcare V1.1*, created in May 2017 and v_2: *Childcare V1.2*, created in November 2017. The ontology *Education* is considered as O' and has two versions v_1': *Education V1.1*, created in January 2017 and v_2': *Education V1.2*, created in September 2017. Amal is using the term `programmOfStudy` from O'. Following each line of Table 2, the following set of cases might occur during the life journey of Amal's ontology:

Row 1. No changes over the terms of v_1' or v_2'

Case 1.a There is no change of t to detect, therefore there is no interest in studying this case.

Case 1.b Amal made a typo by using the term `programOfStudy` (i.e. program is written with one "m" instead of two) in v_1, but then she realizes that this term does not exist in O'. She fixes this mistake by not using it in v_2 anymore.

Case 1.c Amal uses t in both v_1 and v_2. This case might be explained by the fact that t is defined in a previous version (e.g. v_0) of the ontology O' (i.e. $t(v_0') < t(v_1')$).

Case 1.d Amal introduces a mistake by using t in v_2.

Row 2. t is deleted in v'_2
The owners of O' decided to stop using the term `programmOfStudy` in v'_2:

Case 2.a Amal does not use t that was recently deleted. Hence v_1 and v_2 were not affected.
Case 2.b Amal realizes that t was deleted, so she stops using it in v_2.
Case 2.c Amal does not realize the deletion of t, and she keeps using it in v_2.
Case 2.d Amal starts to use t in her second version (v_2), which introduces a mistake.

Row 3. t exist in both v'_1 and v'_2
None of the cases (3.a, 3.b, 3.c, and 3.d) is problematic.

Row 4. t is added to v'_2
The owners of O' introduced a new term `boardingSchool` in v'_2:

Case 4.a Amal has not noticed the addition of t, even if it might be interesting for her to introduce it.
Case 4.b Amal was already using t in (v_1), but she decided to remove it from v_2.
Case 4.c Amal was already using t in v_1, and she continues using it in v_2.
Case 4.d Amal realizes the addition of t, and she starts using it in v_2.

Cases 4.b and 4.c are corner cases that are discussed further Section 4.3.

4.3 Identification of the Occurrences of Adaptation to Ontology Evolution

In this section, we update the results of an experiment[11], initially presented in [27], to detect ontology co-evolution using the cases that are defined in Section 4.2.

We retrieved and analyzed a set of ontologies from two ontology portals:

- The Linked Open Vocabulary (LOV) [37] which currently references 697 different ontologies.[12] Each ontology is described with different features, such as number of incoming links (i.e. how many ontologies are using ontology O), number of outgoing links (i.e. how many ontologies are used by ontology O), number of different versions, and datasets that are using ontology O.
- BioPortal [39] which currently references 827 different ontologies [13] that are related to the biomedical domain. Each ontology is described with different features, such as the number of different versions, along with general metrics (e.g. number of classes, properties and instances).

[11] The experiments with full results can be found at: https://github.com/OmarAlqawasmeh/coEvolutionTermsExtraction
[12] Last counted on January 2020
[13] Last counted on January 2020

After examining the different ontologies, we could retrieve from LOV a set of 28 ontologies with 74 co-evolution instances, as for BioPortal we could retrieve a set of 10 ontologies with 14 co-evolution instances. [14] Then, we extracted the set of terms for each version, and the namespaces for the used ontologies (O''s versions), and we used them to compute the number of occurrences of the different co-evolution cases. For interested readers, full details about our retrieval and processing methods are described in [27].

Table 3 shows the number of occurrences for each co-evolution case for LOV (first value in each cell) and BioPortal (second value). We group the results into three categories: 1. good practices, 2. pitfalls, and 3. uncertain cases.

Table 3: The number of occurrences for each co-evolution case for LOV (first value) and BioPortal (second value) with respect to namespace of (O') [27]

	uses O / O`	a	b	c	d
		v_1 v_2	v_1 t v_2	v_1 t v_2	v_1 t v_2
1	v_1` $v_2$`		✓ 0 130	✗ 3 929	✗ 3 115
2	v_1` $t$ $v_2$`	✓ 23 27	✓ 0 0	✗ 0 3	✗ 0 0
3	v_1` $t$ $v_2$`	? 16875 9135	? 10 0	✓ 270 2058	? 23 0
4	v_1` $t$ $v_2$`	? 2420 1560	✗ 0 115	✗ 0 908	? 0 0

✓ Good practices, ✗ Pitfalls, and ? Uncertain cases

Category 1. Good practices

Case 1.b (i.e. a term t is used in v_1, however it does not exist in v'_1 and v'_2): shows a good practice from the owners of O. They noticed that the term t is not defined in both v'_1 and v'_2, so they decided to delete it from v_2. This co-evolution case occurred 130 times in BioPortal but never in LOV. An example is the co-evolution process of the *Schema.org core and all extension vocabularies* (v_1: created in 2014-10-30 and v_2: created in 2017-05-19), with *Schema.org* ontology (v'_1: created in 2012-04-27 and v'_2: created in 2017-03-23). The terms `Bacteria`,

[14] The co-evolution cases of LOV and BioPortal are inside the *resources* folder at https://github.com/OmarAlqawasmeh/coEvolutionTermsExtraction

`FDAcategoryC` and `Diagnostic` are used in v_1, however they do not exist in v'_1 and v'_2.

Case 2.a (i.e. a term t is deleted in v'_2, and it is not used in any of O's versions): shows a good practice from the owners of O'. They noticed that the term t is not used in both v_1 and v_2 so they decided to delete it from v'_2. This case occurred 23 times in LOV and 27 times in BioPortal. This is a normal case, and no problem occurred during the co-evolution.

Case 2.b (i.e. a term t is deleted in v'_2, and then deleted in v_2): indicates that the set of ontologies stops using the terms after they have been deleted in O' which is a good practice. This case has no occurrence in LOV nor in BioPortal. We are not discussing it further.

Case 2.d (i.e. a term t is deleted in v'_2, however it is added in v_2): indicates that there were no mistake of using the set of deleted terms in the newest version of O. This case has no occurrences in LOV nor in BioPortal. We are not discussing it further.

Category 2. Pitfalls

Cases (1.c and 1.d) demonstrate the problem of using terms that do not exist in v'_1 and v'_2.

Case 1.c (i.e. a term t is used in both v_1 and v_2, however it does not exist in v'_1 and v'_2): This case occurred 3 times in LOV and 929 times in BioPortal. An example is the co-evolution process of the *Statistical Core Vocabulary* (v_1: created in 2011-08-05 and v_2: created in 2012-08-09), with *DCMI Metadata Terms* (v'_1: created in 2010-10-11 and v'_2: created in 2012-06-14). The terms `dc:status` and `dc:partOf` are used in v_1 and v_2, however they do not exist in v'_1 and v'_2.

Case 1.d (i.e. a term t is used in v_2, however it does not exist in v'_1 and v'_2): This case occurred 3 times in LOV and 115 times in BioPortal. An example is the co-evolution process of the *Europeana Data Model vocabulary* (v_1: created in 2012-01-23 and v_2: created in 2013-05-20), with *Dublin Core Metadata Element Set* (v'_1: created in 2010-10-11 and v'_2: created in 2012-06-14). The terms `dc:issued` and `dc:modified` are used in v_2 however they do not exist in v'_1 and v'_2.

A possible explanation is that these terms were used from a previous version of O'. Let's assume that this previous version is v'_0, then these cases can happen only if the publishing time of $t(v'_0)$ is before the publishing time of $t(v'_1)$. In these cases, the owners of O, should be notified of the changes, and they should be suggested to delete the terms that do not exist anymore.

Case 2.c (i.e. a term t is deleted in v'_2, however it is used in v_1 and still in v_2): shows that some terms are still used in both of O's versions after being deleted from O'. This case occurred 3 times in BioPortal. It shows a problem of using terms that do not exist anymore in O'. For example in the co-evolution process of the *Schema.org core and all extension vocabularies* (v_1: created in 2014-10-30 and v_2: created in 2017-05-19), with *Schema.org* ontology (v'_1: created in 2012-04-27 and v'_2: created in 2017-03-23). The terms `MedicalClinic`, `Optician` and `VeterinaryCare` are used in both v_1 and v_2, however they do not exist in the latest version of O' (these different terms were deleted from v_2 of *Schema.org*). In order to prevent such kind of problems the owners should be notified about these cases.

Case 4.b (i.e. a term t is added in v'_2, and it was already used in v_1): This case occurred 115 times in BioPortal, and it has no occurrence in LOV. An example is the co-evolution process of the *Schema.org core and all extension vocabularies* (v_1: published in 2014-10-30 and v_2: published in 2017-05-19), with *Schema.org* ontology (v'_1: created in 2012-04-27 and v'_2: created in 2017-03-23). The terms `SoundtrackAlbum`, `Hardcover` and `SingleRelease` are used in v_1, however they were introduced later in v'_2. The v_1 of *Schema.org core and all extension vocabularies* uses terms that were later defined by v'_2 of *Schema.org* ontology. The *Schema.org core and all extension vocabularies* is an extension of *Schema.org*, however it has its own namespace. Each reviewed extension for schema.org has its own chunk of schema.org namespace (e.g. if extension name is x, the namespace of this extension is x1.schema.org). [15] We retrieved all terms that have the namespace of *Schema.org*.[16] Other terms with different namespaces were discarded. [17] This reflects a bad practice in a way of using terms that have not been defined in the second version. These terms could be harbinger to add in the next versions.

Case 4.c (i.e. a term t is added in v'_2, and it was already used in both of O's versions): This case occurred 951 times in BioPortal, and it has no occurrence in LOV. An example is the co-evolution process of the *Semanticscience Integrated Ontology (SIO)* (v_1: created in 2015-06-24 and v_2: created in 2015-09-02), with *The Citation Typing Ontology (CITO)* (v'_1: created in 2010-03-26 and v'_2: created in 2015-07-03). The term `citesAsAuthority` is used in both v_1 and v_2, however it was introduced in v'_2. One explanation for this kind of pitfalls is that the knowledge engineers might have introduced a typos during the development process of the ontology. In these cases, the owners of O, should be notified that the term they use is not a term.

Category 3. Uncertain cases

In cases (3.a, 3.b, 3.c and 3.d) from LOV and Bioportal, there was no change of terms in the two versions of O'. This indicates that the co-evolution process has no problem to report. Some terms are shared between v'_1 and v'_2 so there was no addition or deletion over them.

Cases 4.a and 4.d in both LOV and BioPortal show the number of terms that were added during the evolution of O'. These terms were not used in any of O's versions. These cases can be explained in two ways:

1. The owners of O did not notice the addition of these terms, however they might be interested in using some of these new terms. This might reflects

[15] More details about the extensions managing of schema.org can be found at: `https://schema.org/docs/extension.html`

[16] namespace of *Schema.org* is http://schema.org/

[17] Some examples of discarded namespaces: https://health-lifesci.schema.org/, https://pending.schema.org/, https://meta.schema.org/

a problem, thus further content analysis should be introduced to possibly recommend changes to the owners.
2. The owners of O noticed the addition of these terms and they decided not to add them.

In the next section, we present a catalogue of pitfalls that are related to versioned and networked ontologies.

5 The Set of Pitfalls over Ontology Networks or Versioned Ontologies

Hitherto, we have introduced the different cases (good, pitfalls and uncertain) that could occur inside one particular setting of ontologies combining networked and versioned ontologies (i.e. ontology co-evolution). In this section, we enrich previous work to investigate ontology pitfalls in distinguishing between the two situations: 1. versioned ontologies and, 2. ontology networks. Hence, in Section 5.1 we describe 9 candidate pitfalls that are related to versioned ontologies and ontology networks.

5.1 Candidate pitfalls

Pitfall 1. Ontology is not accessible at its IRI.
This pitfall is related to the ontology relationship $type(e) = \{\text{“}uses\text{”}, \text{“}imports\text{”}\}$. It can occur in the following cases:

1. If an ontology was never published on-line, for instance, if an ontology is used internally by a company, and/or if an ontology file becomes private and it is not accessible anymore.
2. If the ontology is not available at its IRI. For example: the IRI of the *pizza* ontology is `http://www.co-ode.org/ontologies/pizza/pizza.owl#`, but it is not accessible at this IRI.
3. If the IRI of the ontology has been changed. For example: the IRI of the *DOLCE Ultralite upper* ontology was originally `http://www.loa-cnr.it/ontologies/DUL.owl#`. The website loa-cnr.it closed, and the ontology is now available at `http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#`.

This pitfall affects ontology networks. Any import of an ontology that has this pitfall will fail. To solve or avoid this pitfall, we suggest to verify the imported IRIs for any changes that could occur or to locally maintain a copy of the ontology and use it offline.

Pitfall 2. Importing an ontology using a non persistent IRI or the IRI of a representation (the file URL)
This pitfall is related to the ontology relationship $type(e) = \{\text{“}imports\text{”}\}$. Persistent IRIs are important, as they ensure that the ontology will be always accessible at the same IRI. This pitfall occurs in two cases:

1. If a knowledge engineer imports a non persistent IRI, for example: the *SEAS* ontology has persistent IRI https://w3id.org/seas/, which no longer redirects to the location https://ci.emse.fr/seas/. Assume an ontology imported the IRI https://ci.emse.fr/seas/. Due to the renaming of the EMSE institution, the IRI now redirects to the location https://ci.mines-stetienne.fr/seas/, thus the import would break.
2. If a knowledge engineer imports the file URL instead of the ontology IRI, for example: the W3C organization ontology has persistent IRI https://www.w3.org/ns/org, with two representations at https://www.w3.org/ns/org.rdf and https://www.w3.org/ns/org.ttl. Assume an ontology imports the ontology representation https://www.w3.org/ns/org.rdf instead of the ontology series https://www.w3.org/ns/org. In case of the deletion of the RDF/XML representation any import would break.

This pitfall affects ontology networks. Any import of an ontology that has this pitfall will fail. To solve or avoid this pitfall, we suggest: 1. to use only persistent IRIs when importing ontologies, and 2. to always use the IRI of the ontology, and not the URL of the file representation.

Pitfall 3. Importing an inconsistent ontology
This pitfall is related to the ontology relationship $type(e) = \{\text{“}imports\text{”}\}$. It occurs if a knowledge engineer imports an inconsistent ontology, for example: the *SAREF4ENER* ontology (EEbus/Energy@home) https://w3id.org/saref4ee is inconsistent. The importing ontology would become inconsistent too. This pitfall affects both ontology networks and versioned ontologies. To solve or avoid this pitfall, we suggest: 1. to check the consistency of an ontology before importing it, 2. to use only the specific terms that are needed from the ontology (i.e. by their IRIs) instead of importing the whole ontology, and/or 3. to try contacting the ontology owners so that they solve the inconsistency.

Pitfall 4. Only the latest version of the ontology is available online
This pitfall is related to the ontology relationship $type(e) = \{\text{“}uses\text{”}, \text{“}imports\text{”}\}$. It occurs when the only available version of the ontology is the latest version. For example, the *S4WATR* ontology is published at https://w3id.org/def/S4WATR, but only the latest version is available online. Let's assume an ontology imports the *S4WATR* ontology at a certain point in time. Later, some terms are deleted or added in S4WATR ontology. Then the importing ontology could break or become inconsistent.

This pitfall affects both ontology networks and versioned ontologies. To solve or avoid this pitfall, the following practices could be followed: 1. to import an ontology with its version URI, and/or 2. to monitor the evolution of the imported ontology to react appropriately.

Pitfall 5. Importing an ontology series IRI instead of an ontology version IRI
This pitfall is related to the ontology relationship $type(e) = \{\text{“}uses\text{”}, \text{“}imports\text{”}\}$. It occurs if a knowledge engineer imports an ontology series IRI instead of

an ontology version IRI. For example, the *SAREF* ontology series has IRI https://saref.etsi.org/saref#, and version 2.1.1 has IRI https://saref.etsi.org/saref/v2.1.1/saref#. A new version 3.1.1 is under development and will delete terms from version 2.1.1. Let O' be an ontology that imports *SAREF* ontology 2.1.1 using https://saref.etsi.org/saref#. When the new version 3.1.1 is released, the importing ontology O' could break or become inconsistent.

This pitfall affects both ontology networks and versioned ontologies. To solve or avoid this pitfall we recommend to import the ontology version IRI instead of the ontology series IRI.

Pitfall 6. Ontology series IRI is the same as the ontology version IRI This pitfall is related to the ontology relationship $type(e) = \{\text{“}imports\text{”}\}$. It occurs whenever a IRI refers to a specific version of the ontology. For example, the *Units of Measure (OM)* ontology version 1.8 has IRI http://www.wurvoc.org/vocabularies/om-1.8/, and version 2.0 has IRI http://www.ontology-of-units-of-measure.org/resource/om-2/. Each time a new version is published, the ontology IRI should be updated, this does not conform to the OWL2 specification.[18] This pitfall affects both ontology networks and versioned ontologies. To solve or avoid this pitfall we recommend to delete the version number from the IRI of the ontology, or to send a notification message with the new IRI when a new version of the ontology is released.

Pitfall 7. A term is moved from one ontology module to another This pitfall is related to the ontology relationship $type(e) = \{\text{“}uses\text{”}\}$. It occurs when a term is moved from one ontology module to another, which causes the change in its IRI. For example, the *SAREF* ontologies [31] consist of 1. *SAREF core*, 2. *SAREF4SYST*, and 3. several ontologies for the verticals, such as SAREF4ENER, SAREF4BLDG, and SAREF4ENVI. In *SAREF-core 1.1.1*, created in 2015, the authors defined the term `saref:BuildingObject`. Later in 2016, SAREF-core 2.1.1 was published without the term `saref:BuildingObject`. However, another ontology SAREF4BLDG was created with the term `s4bldg:BuildingObject`, with the same definition as `saref:BuildingObject`.

In this case, the IRI of the term `BuildingObject` has been changed. This might have a functional impact over the artifacts that are reusing the term (e.g. some queries might be affected by the change of the IRI).

This pitfall affects ontology networks. To solve or avoid this pitfall we recommend to take extra care while moving terms between the different modules, and to notify the users of the ontology in case of changes.

Pitfall 8. Namespace hijacking [from [24]] This pitfall is related to the ontology relationship $type(e) = \{\text{“}uses\text{”}\}$. It refers to reusing or referring to terms from another namespace that are not defined in such namespace [24]. For example, the description of classes `qudt-1.1:QuantityValue` and `qudt-1.1:Quantity` are not available at their

[18] https://www.w3.org/TR/owl2-syntax/, sections 3.1 and 3.3

own IRIs. Instead, they are defined in the ontology http://qudt.org/1.1/schema/quantity#.

This pitfall affects ontology networks as it prevents the retrieval of valid information when looking for the hijacked terms, which violates the Linked Data publishing guidelines [14]. To solve or avoid this pitfall we recommend to define new terms in the namespace that is owned and controlled by the knowledge engineer.

Pitfall 9. The IRI of a term contains a file extension
This pitfall is related to the ontology relationship $type(e) = \{\text{``}uses\text{''}\}$. It occurs if a term's IRI contains a file extension. For example, the terms in the *Dolce ultra lite* ontology have the namespace http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#. Let's assume that some day the publisher of *dolce-very-lite* wants to set up content negotiation to expose an html documentation of their ontology. As the IRI of the terms contains the file extension ".owl", no content negotiation can take place. If a human looks up the term IRI, he/she will access the OWL file, and not the html documentation. To solve or avoid this pitfall we recommend to stop using file extensions inside IRIs, and follow the rules of cool URIs for the Semantic Web. [19]

As a summary, Table 4 presents the set of pitfalls based on the following criteria: 1. Affect: whether the pitfall affects ontology networks and/or versioned ontologies, 2. problems that might occur as a consequence of having the pitfall, and 3. recommendations to avoid or solve the pitfall.

6 Evaluating the Importance and Impact of the Candidate Pitfalls

We evaluated the importance and potential impact of the candidate pitfalls using a survey we conducted in the semantic web community. Section 6.1 describes the survey, then Section 6.2 quantitatively evaluates the answers. Finally Section 6.3 reports on the different opinions and suggestions we gathered from the participants.

6.1 Description of the survey

The survey[20] first requests some information about the level of expertise of the participant in (1) ontology engineering in general, (2) versioned ontologies, and (3) networked ontologies. We used a Likert scale with values from 1 (beginner) to 10 (expert). Then, each pitfall is described with an illustrative example, and the participant is asked to answer to the following questions:

1. How often have you encountered this pitfall before?
2. How problematic is this pitfall?

[19] Cool URIs can be found at: https://www.w3.org/TR/cooluris/
[20] The survey can be found at: http://bit.ly/36JQfgO

Table 4: A summary for the set of pitfalls

Code	**Description**	**Affects**	**Problem caused**	**Recommendations to solve or to avoid**
P1	Ontology is not accessible at its IRI	Ontology networks and versioned ontologies	Import failure	Keep the ontology always available at its IRI by controlling and managing the possible changes during the evolution
P2	Importing an ontology using a non persistent IRI or the IRI of a representation (the file URL)	Ontology networks	Import failure	Try to always use a persistent IRI while importing an ontology
P3	Importing an inconsistent ontology	Ontology networks and versioned ontologies	Inheritance of the inconsistency	Make sure to import consistent ontologies or using specific terms instead of importing the whole ontology
P4	Only the latest version of the ontology is available online	Ontology networks and versioned ontologies	Import failure and/or the ontology become inconsistent	Import an ontology with its version URI, and/or follow the evolution of the imported ontology and change your ontology accordingly
P5	Importing an ontology series IRI instead of an ontology version IRI	Ontology networks and versioned ontologies	Import failure and/or the ontology become inconsistent	Import the ontology version IRI instead of the ontology series IRI
P6	Ontology series IRI is the same as the ontology version IRI	Ontology networks and versioned ontologies	Import failure	Delete the version number from the IRI, and/or to send a notification message with the new IRI when a new version is released.
P7	A term is moved from one ontology module to another with different IRI	Ontology network	Functional impact, research queries might break.	Take extra care when moving terms, and notify the different users in case of changes.
P8	Namespace hijacking	Ontology networks	Retrieve invalid information while looking up for the hijacked terms	Define terms in the namespace that is owned and controlled by the knowledge engineer
P9	The IRI of a term contains a file extension	Stand-alone ontology pitfall, however can affect ontology networks	No content negotiation can take place	Stop using file extensions inside IRIs

3. How would you rate the impact on subsequent versions of the ontology?
4. How problematic is it to import ontologies that have this pitfall?

For the answers, we also use a Likert scale from 1 (Strongly Disagree) to 5 (Strongly Agree). For each pitfall, participants may additionally share known occurrences of the pitfall, and ideas or recommendations to solve or avoid it. Finally, we ask the participant to what extent he/she agrees or not with our pitfalls categorization, and to rate about his/her overall confidence while filling the survey.

6.2 Quantitative evaluation of pitfalls

A total of 27 participants answered the survey between November 2019 to January 2020.[21] As shown in Figure 4, the most of the participants declared expertise in ontology engineering, ontology networks and ontology versioning.

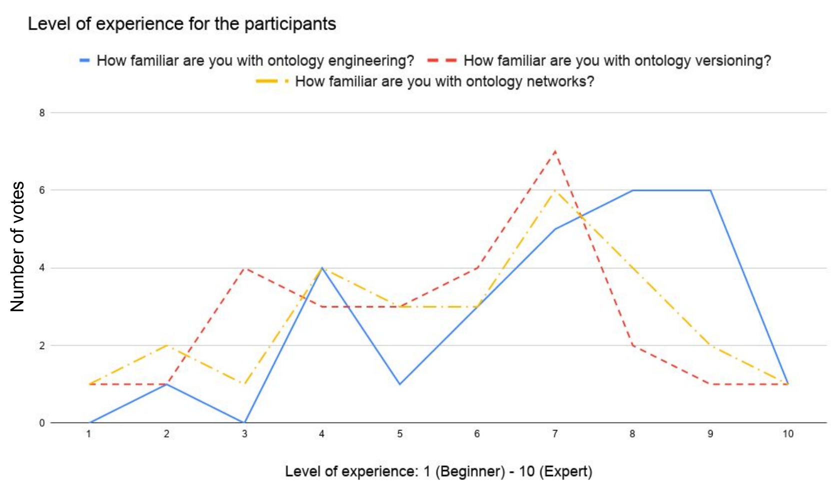

Fig. 4: Level of experience for the participants (weighted average)

We consider that the value of the participants' opinion is increasing with his/her level of experience. Thus, we calculated the weighted average (WA) for the different answers for each pitfall:

$$\mathrm{WA} = \frac{\sum_{i=1}^{N} w_i \cdot x_i}{\sum_{i=1}^{N} w_i}$$

[21] Raw results can be found at: http://bit.ly/2RztHKq

where w_i is the value of expertise of participant i and x_i his response. Then, we assess the agreement level between the different participants using the consensus measure (Cns) proposed by [36]:

$$\mathrm{Cns}(X) = 1 + \sum_{j=1}^{n} p_j \cdot \log_2 \left(1 - \frac{|X_j - \mu_X|}{d_X}\right)$$

where X is the values vector (i.e. values from 1-5), p_j is the relative frequency of answer j, μ_X is the mean of X, and $d_X = X_{max} - X_{min}$ is the width of X.

The value of the consensus measure ranges between 0 (total disagreement) and 1 (total agreement). Authors of [17] proposed the following interpretation for intermediate values: a) Less than 0: poor agreement, b) 0.01–0.20: slight agreement, c) 0.21–0.40: fair agreement, d) 0.41–0.60: moderate agreement, e) 0.61–0.80: substantial agreement, and f) 0.81–1.00: almost perfect agreement.

We adapted the definition of p_j in the consensus (Cns) formula to account for the level of expertise of each participant:

$$p_j = \frac{\sum_{i=1}^{N} w_i \cdot \delta_{vote(i),j}}{\sum_{i=1}^{N} w_i}$$

where $\delta_{vote(i),j} = 1$ if participant i voted j, and 0 otherwise. Table 5 presents the weighted average and the consensus value for the different questions of the survey, computed using R.[22]

Table 5: Weighted average and consensus ratio for the survey's answers

	Weighted average (/5) VS Consensus value (/100)					
	How problematic is it?		Impact on versioned ontologies		Impact on ontology networks	
Pitfall	Weighted Avg.	Consensus	Weighted Avg.	Consensus	Weighted Avg.	Consensus
P1	4.06	68.61	4.20	77.10	4.18	69.80
P2	3.46	64.81	3.53	58.59	3.59	59.64
P3	3.93	62.52	3.63	55.25	4.10	61.59
P4	3.66	63.96	3.88	61.55	3.60	55.16
P5	3.64	66.61	3.57	54.03	3.57	60.38
P6	2.60	60.63	2.65	68.80	2.59	62.08
P7	3.51	57.89	3.48	60.94	3.60	57.16
P8	3.54	52.86	3.31	62.45	3.32	59.02
P9	3.02	66.40	2.98	61.55	2.49	67.17

Agreement on the classification (/100)	75.59
Level of confidence (/100)	67.85

[22] The source code found in *resources/SurveyExperiments* at https://github.com/OmarAlqawasmeh/coEvolutionTermsExtraction

The overall level of confidence while filling the survey is around 68%, and the participants substantially agreed (Cns = 76%) with the new categorization we proposed for ontology pitfalls (i.e. stand-alone ontology pitfalls, versioned ontologies pitfalls, and pitfalls inside ontology networks). Table 6 categorizes the pitfalls based on their estimated impact into: 1. Major impact (WA > 3.5), 2. Middle impact ($3 <$ WA < 3.5), and 3. Less impact (WA < 3). We rank the pitfalls' impact in descending order (i.e. high to less). As shown in Table 6, there is a substantial agreement that *P1*[23] and *P4* have a major impact on versioned ontologies, and *P1* and *P3* have a major impact on ontology networks. Pitfalls *P1, P2, P3, P4, and P5* have a major impact on both versioned ontologies and ontology networks. *P7* has a middle impact on versioned ontology but a major impact on ontology networks. *P8* has a middle impact on both versioned ontologies and ontology networks. As for *P6 and P9* the participants substantially agree that they have less impact.

Table 6: Pitfalls ranked by their impact over versioned and networked ontologies

Impact on	Major	Middle	Less
Versioned ontologies	P1$^+$, P4$^+$, P3*, P5*, and P2*	P7*, and P8$^+$	P6$^+$, and P9$^+$
Ontology networks	P1$^+$, P3$^+$, P7*, P4*, P2*, and P5*	P8*	P6$^+$ and P9$^+$

+: substantial agreement
*: moderate agreement

Moreover, Figure 5 presents how often the participants encountered the different pitfalls. We can see that except for P6, P7, and P8, all participants encountered the different pitfalls before.

6.3 Analyzing the participants opinions

For each pitfall, participants could share known occurrences of the pitfall, and ideas or recommendations to solve or avoid it. We summarize the gathered opinions (OPN) below.

[23] P1. Ontology is not accessible at its IRI
P2. Importing an ontology using a non persistent IRI or the IRI of a representation
P3. Importing an inconsistent ontology
P4. Only the latest version of the ontology is available online
P5. Importing an ontology series IRI instead of an ontology version IRI
P6. Ontology series IRI is the same as the ontology version IRI
P7. Term is moved from one ontology module to another with different IRI
P8. Namespace hijacking
P9. The IRI of a term contains a file extension

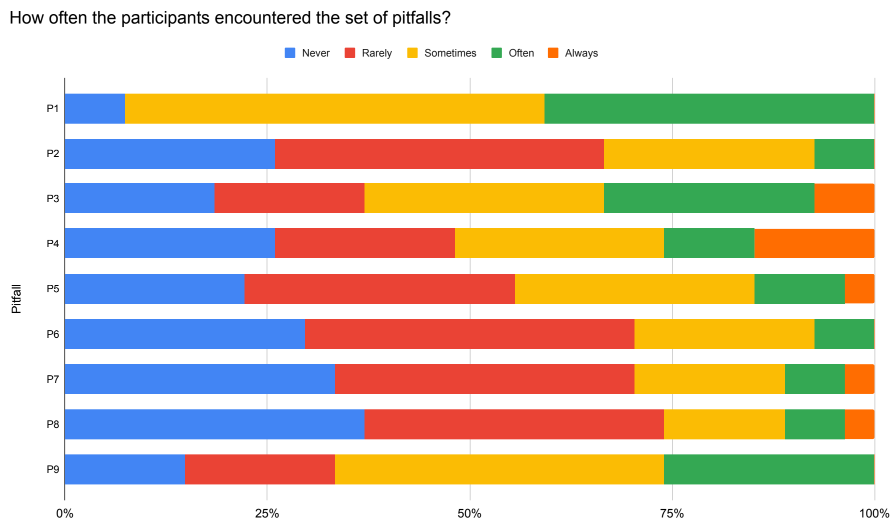

Fig. 5: How often the participants encountered the candidate pitfalls

OPN 1. *Persistent IRIs are important:* participants agreed about the importance of persistent IRIs when creating or reusing ontologies. Some suggest to use services or catalogues to ensure the usage of persistent IRIs. Using persistent IRIs can effectively help to avoid pitfalls P1 and P2.

OPN 2. *Consistency tests should be made on the imported ontologies:* ontology editors should applying consistency tests on the imported ontologies to avoid pitfall P3.

OPN 3. *When reusing terms, refer only to those that are needed:* Some of the participants suggest to avoid importing the whole ontology and only declare the required terms. Ontology editors should check that these terms are correctly declared (e.g., a term that is originally declared a datatype property should not be declared as an annotation property), and services should be developed to monitor the evolution of the ontologies to prevent pitfall P7.

OPN 4. *Import the ontology using its version IRI.* This point has both advantages and disadvantages, on the one hand, importing an ontology version IRI prevents any issue that may arise if the imported ontology evolves. On the other hand, it may be interesting to update an ontology when a new version of an imported ontology is issued. Again, services could be developed to notify ontology editors about any new version release of the imported ontologies.

OPN 5. *A notification message should be send in case of moving terms from one module to another.* A subscription mechanism could be used to notify the different external artifacts (e.g. systems, ontologies) when an ontology evolves.

OPN 6. *Focusing only on the ontology level is not sufficient enough.* A participant argued that focusing on the quality of ontologies is less important than focusing on the quality of their usage. The following questions have been raised:
- How to improve the integration of heterogeneous data that was designed independently of the ontologies?)
- What can go wrong when the data and the ontology become misaligned?
- How to deal with noisy knowledge situations where the logic embedded in the ontology becomes unusable?

These different points can be topics for future work.

7 Discussion and Conclusion

In a previous contribution, we presented a definition of a situation of ontology evolution which considers the evolution of an ontology O that uses terms that have the namespace of O' (i.e. ontology co-evolution). We systematically listed and described the different cases of the adaptation to ontology co-evolution. We observed occurrences of these cases over two ontology portals: the Linked Open Vocabulary (LOV), and BioPortal. As the outcome of this study, we identified good practices and pitfalls.

Where the state of the art studies addressed stand-alone ontology pitfalls, in this paper we identified 9 candidate pitfalls that may affect versioned ontologies, i.e. when an ontology O_1 evolves to O_2, and/or ontology networks, i.e. when an ontology O uses or imports another ontology O'. In order to measure the importance and potential impact of the candidate pitfalls, we distributed a survey to the semantic web community. Participants agreed that listing and investigating in ontology pitfalls can effectively enhance the quality of ontologies which reflects in a positive way in using these ontologies for the different tasks (e.g. question answering). Moreover, we suggested a set of best practices to be followed in order to prevent or solve the candidate pitfalls.

As for future work, we are interested in targeting the following issues we identified:

1. *Some existing tools leads to the creation of pitfalls.* There exist some tools that could lead to the creation of some pitfalls. For example OnToology tool[24] publishes only the latest version of an ontology, and the documentation of this latest version (even if the ontology includes provenance information and information about the previous versions). It is important to update this tool so that all the versions are published. In case the owners of O' uses OnToology to publish it. Any other ontology that uses O' will be forced to import O' using its ontology series IRI or the latest ontology version IRI.

[24] http://ontoology.linkeddata.es/

Then using another ontology version IRI will have the risk to break this import in the future.

2. *The inheritance of a pitfall*: Some pitfalls can be inherited either when they evolve or when they are used in other ontologies. For example, if an ontology O' has the pitfall "creating the relationship (is) instead of using rdfs:subClassOf, rdf:type or owl:sameAs" [25], it means that O' has a property called *is*. If another ontology O uses O', then this pitfall will propagate to O automatically.

Moreover, we plan to develop a service that can automatically observe and notify the ontologies' owners during the evolution process. Having such tool can help to keep track of the ontologies in the different setting we proposed (i.e. ontology co-evolution, ontology versioning, and ontology networking). This will reflect positively on the quality of the different ontologies.

Finally, we plan to investigate further on the set of pitfalls that might occur on the data level (suggested by some of the survey's participants), mainly to focus on the set of pitfalls that might occur between the data and the ontologies, such as when the data and the ontology become misaligned due to the evolution of the ontology.

Acknowledgment

The authors would like to thank María Poveda (Universidad Politécnica de Madrid) for her suggestions and comments during the creation of the candidate pitfalls. Additionally, the authors would also like to thank the 27 anonymous participants for their valuable contribution to the experimental evaluation of the candidate pitfalls.

References

1. Abdel-Qader, M., Scherp, A., Vagliano, I.: Analyzing the evolution of vocabulary terms and their impact on the LOD cloud. In: The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings. pp. 1–16 (2018)
2. Abgaz, Y.M., Javed, M., Pahl, C.: Analyzing impacts of change operations in evolving ontologies. In: Proceedings of the 2nd Joint Workshop on Knowledge Evolution and Ontology Dynamics, USA (2012)
3. Bernaras, A., Laresgoiti, I., Corera, J.M.: Building and reusing ontologies for electrical network applications. In: 12th European Conference on Artificial Intelligence, Budapest, Hungary, August 11-16, 1996, Proceedings. pp. 298–302 (1996)
4. Brickley, D., Miller, L.: FOAF vocabulary specification 0.91 (2010)
5. Castano, S., Ferrara, A., Hess, G.N.: Discovery-driven ontology evolution. In: SWAP 2006 - Semantic Web Applications and Perspectives, Proceedings of the 3rd Italian Semantic Web Workshop, Scuola Normale Superiore, Pisa, Italy (2006)

[25] Pifall number 3 from http://oops.linkeddata.es/catalogue.jsp

6. Doran, P., Tamma, V.A.M., Iannone, L.: Ontology module extraction for ontology reuse: an ontology engineering perspective. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007. pp. 61–70 (2007)
7. Dragoni, M., Ghidini, C.: Evaluating the impact of ontology evolution patterns on the effectiveness of resources retrieval. In: Proceedings of the 2nd Joint Workshop on Knowledge Evolution and Ontology Dynamics, Boston, MA, USA (2012)
8. Gaudet, P., Dessimoz, C.: Gene ontology: pitfalls, biases, and remedies. In: The Gene Ontology Handbook, pp. 189–205. Humana Press, New York, NY (2017)
9. Groß, A., Hartung, M., Prüfer, K., Kelso, J., Rahm, E.: Impact of ontology evolution on functional analyses. Bioinformatics **28**(20), 2671–2677 (2012)
10. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2), 199–220 (1993)
11. Guha, R.V., Brickley, D., Macbeth, S.: Schema.org: evolution of structured data on the web. Communications of the ACM **59**(2), 44–51 (2016)
12. Haase, P., Rudolph, S., Wang, Y., Brockmans, S.: D1. 1.1 networked ontology model
13. Hartung, M., Groß, A., Rahm, E.: CODEX: exploration of semantic changes between ontology versions. Bioinformatics (2012)
14. Heath, T., Bizer, C.: Linked data: Evolving the web into a global data space. Synthesis lectures on the semantic web: theory and technology **1**(1), 1–136 (2011)
15. Kupfer, A., Eckstein, S.: Coevolution of database schemas and associated ontologies in biological context. In: 22nd British National Conference on Databases (2006)
16. Kupfer, A., Eckstein, S., Neumann, K., Mathiak, B.: A coevolution approach for database schemas and related ontologies. In: 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2006), Salt Lake City, Utah, USA. pp. 605–610 (2006)
17. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. biometrics pp. 159–174 (1977)
18. Lebo, T., Sahoo, S., McGuinness, D.: PROV-O: The PROV ontology. W3C recommendation **30** (2013)
19. Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P.N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S., Bizer, C.: Dbpedia - A large-scale, multilingual knowledge base extracted from wikipedia. Semantic Web **6**(2), 167–195 (2015)
20. Mihindukulasooriya, N., Poveda-Villalón, M., García-Castro, R., Gómez-Pérez, A.: Collaborative ontology evolution and data quality - an empirical analysis. In: 13th International Workshop, OWLED, and 5th International Workshop, Bologna, Italy. pp. 95–114 (2016)
21. Noy, N.F., Musen, M.A.: PROMPTDIFF: A fixed-point algorithm for comparing ontology versions. In: The Eighteenth National Conference on AI and Fourteenth Conference on Innovative Applications of AI, Canada. (2002)
22. Ottens, K., Aussenac-Gilles, N., Gleizes, M.P., Camps, V.: Dynamic ontology co-evolution from texts: Principles and case study. In: Proceedings of the First International Workshop on Emergent Semantics and Ontology Evolution, ESOE 2007, co-located with ISWC 2007 + ASWC 2007, Busan, Korea. pp. 70–83 (2007)
23. Papavassiliou, V., Flouris, G., Fundulaki, I., Kotzinos, D., Christophides, V.: On detecting high-level changes in RDF/S kbs. In: ISWC 2009, 8th International Semantic Web Conference,USA. pp. 473–488 (2009)
24. Poveda Villalón, M.: Ontology Evaluation: a pitfall-based approach to ontology diagnosis. Ph.D. thesis, ETSI_Informatica (2016)

25. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: Oops! (ontology pitfall scanner!): An on-line tool for ontology evaluation. Int. J. Semantic Web Inf. Syst. **10**(2), 7–34 (2014)
26. Poveda Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: The landscape of ontology reuse in linked data (2012)
27. Qawasmeh, O., Lefrançois, M., Zimmermann, A., Maret, P.: Observing the impact and adaptation to the evolution of an imported ontology. In: Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2019, Volume 2: KEOD, Vienna, Austria, September 17-19, 2019. pp. 76–86 (2019)
28. Sabou, M., Fernández, M.: Ontology (network) evaluation. In: Ontology Engineering in a Networked World., pp. 193–212 (2012)
29. Savic, M., Ivanovic, M., Jain, L.C.: Complex Networks in Software, Knowledge, and Social Systems, Intelligent Systems Reference Library, vol. 148. Springer (2019)
30. Simperl, E.P.B.: Reusing ontologies on the semantic web: A feasibility study. Data Knowl. Eng. **68**(10), 905–925 (2009)
31. SmartM2M, E.: Saref consolidation with new reference ontology patterns, based on the experience from the seas project. 2019 jul. report no.: Ts 103 548 v1. 1.1
32. Stojanovic, L.: Methods and tools for ontology evolution. Ph.D. thesis, Karlsruhe Institute of Technology, Germany (2004)
33. Stojanovic, L., Maedche, A., Stojanovic, N., Studer, R.: Ontology evolution as reconfiguration-design problem solving. In: Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP 2003), October 23-25, 2003, Sanibel Island, FL, USA (2003)
34. Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A.: Introduction: Ontology engineering in a networked world. In: Ontology Engineering in a Networked World., pp. 1–6 (2012)
35. Tartir, S., Arpinar, I.B., Sheth, A.P.: Ontological evaluation and validation. In: Theory and applications of ontology: Computer applications. Springer (2010)
36. Tastle, W.J., Wierman, M.J.: Consensus and dissention: A measure of ordinal dispersion. Int. J. Approx. Reasoning **45**(3), 531–545 (2007)
37. Vandenbussche, P., Atemezing, G., Poveda-Villalón, M., Vatant, B.: Linked open vocabularies (LOV): A gateway to reusable semantic vocabularies on the web. Semantic Web (2017)
38. Vigo, M., Bail, S., Jay, C., Stevens, R.D.: Overcoming the pitfalls of ontology authoring: Strategies and implications for tool design. Int. J. Hum.-Comput. Stud. **72**(12), 835–845 (2014)
39. Whetzel, P.L., Noy, N.F., Shah, N., Alexander, P.R., Dorf, M., Fergerson, R.W., Storey, M.D., Smith, B., Chute, C.G., Musen, M.A.: Bioportal: Ontologies and integrated data resources at the click of a mouse. In: Proceedings of the 2nd International Conf. on Biomedical Ontology, Buffalo, NY, USA (2011)
40. Zablith, F.: Ontology evolution: A practical approach. In: Workshop on Matching and Meaning at Artificial Intelligence and Simulation of Behaviour (2009)
41. Zablith, F., Antoniou, G., d'Aquin, M., Flouris, G., Kondylakis, H., Motta, E., Plexousakis, D., Sabou, M.: Ontology evolution: a process-centric survey. Knowledge Eng. Review **30**(1), 45–75 (2015)