Collaborative Ontology Evolution and Data Quality - An Empirical Analysis

Nandana Mihindukulasooriya, María Poveda-Villalón, Raúl García-Castro, and Asunción Gómez-Pérez

Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
{nmihindu,mpoveda,rgarcia,asun}@fi.upm.es

Abstract. Since more than a decade, theoretical research on ontology evolution has been published in literature and frameworks for managing ontology changes have been developed. However, there are less studies that analyze widely used ontologies developed in a collaborative manner to understand community-driven ontology evolution in practice. In this paper, we perform an empirical analysis on how four well-known ontologies (DBpedia, Schema.org, PROV-O, and FOAF) have evolved through their lifetime and an analysis of the data quality issues caused by some of the ontology changes.

Keywords: Ontology, Change, Evolution, Quality, DBpedia, Schema.org, PROV-O, FOAF

1 Introduction

Ontologies are defined as formal, explicit specifications of a shared conceptualization of a domain of interest [1] and inherently ontology development becomes a collaborative process due to the "shared conceptualization" aspect. Once an ontology is defined, changes to it are inevitable. The main causes of ontology change include (i) changes in the domain, (ii) changes in conceptualization, and (iii) changes in the explicit specification [2]. As the knowledge about the domain of interest evolves dynamically over time, ontology changes have to occur to reflect those changes in the conceptualization. Changes may occur not only because of the changes in the reality but also because of the decisions of the ontology engineer. The ontology engineer may decide to expand the scope of the conceptualization by adding new concepts and relations or to apply other design patterns and strategies requiring modifications to the existing ontology both in structure and in semantics, or to remove existing terms due to various reasons, for instance, because some existing terms become obsolete, cause confusion in users, or contain erroneous representations such as typos.

The elements that are subject to change in ontologies include concepts, properties, instances and axioms. These changes to the ontologies impact the entities that are described using these ontologies as well as any other ontologies that are dependent on the given ontology. In the ontology evolution topic, research looked into several problems and challenging tasks. These problems include consistency maintenance, backward compatibility, ontology manipulation, understanding of ontology evolution, change propagation, etc. In this paper, we focus on analyzing the changes in different versions of the ontology and on discussing their effect on data quality.

According to Stojanovic [3], ontology evolution refers to the activity of facilitating the modification of an ontology by preserving its consistency, while the ontology modification activity would consist on changing the ontology without considering the consistency. For the sake of clarity we will use the term "ontology evolution" regardless whether consistency is preserved.

Furthermore, ontology designers are not always aware of or have any centralized control over the ontology users. Thus, it is not always possible for the designers to notify the changes in an ontology to their users. And, even if those changes were notified, there could be data that use a previous version of the ontology.

Starting in the early nineties, theoretical research on ontology evolution has been published in literature [2, 4] and frameworks for managing ontology evolution [5–7] have been developed. However, to the best of the authors' knowledge there are less studies that analyze how such research was used in widely used ontologies developed in a collaborative manner. In this regard, this paper provides an empirical study on how four widely used ontologies have evolved through their lifetime.

The rest of the paper is organized as follows. Section 2 presents an overview of pioneer works on ontology evolution and a review of the quality issues that ontology evolution might cause. Next, Section 3 describes the ontologies selected to be analyzed. Section 4 details the obtained results and the analysis carried out for each of the selected ontologies. Finally, Section 5 is devoted to final conclusions and suggestions for improvements.

2 Ontology Evolution

First steps towards ontology evolution were proposed by Noy and Klein [2] in the early nineties. They presented an analysis of the traditional versioning and evolution dimensions that apply to databases and discussed whether they can be applied to ontologies. They also provides a comprehensive list of possible changes affecting the ontology and their impact on the instance data (mostly oriented to DAML-OIL ontologies).

From a methodological point of view, the DILIGENT [8] methodology has been one of the first approaches for supporting distributed ontology development in a distributed environment taking into account the evolution of ontologies. The general process proposed comprises five main activities: build, local adaptation, analysis, revision, and local update.

Regarding approaches based on OWL ontologies, we can mention the process proposed by Stojanovic [3] that enables handling ontology changes ensuring the consistency of the ontology. Along this work a taxonomy of ontology changes is proposed based on whether the changes are additive or subtractive applied to the following list of ontology elements or meta changes: – Concept – Concept Hierarchy – Property – Property Hierarchy – Property Domain – Property Range – Property Symmetric – Property Transitive – Property Inverse – Max Cardinality – Min Cardinality – Instance – InstanceOf – InstProp – OI-model

Finally, Palma [5] proposed an ontology metadata model to provide a high level overview of how an ontology has changed. Such model is the basis for the methods proposed in this work for ontology change management in distributed environments and the strategies for supporting collaborative ontology development. In this work, not only a plain list of possible changes is provided but a more complex model that includes atomic changes, entity changes and composite changes.

More recently, Dragoni and Ghidini [9] explored ontology evolution from the information retrieval tools point of view. In this work, the authors analyze how changes in ontologies might affect information retrieval performance in terms of precision and recall. In this case the modifications made on an ontology are defined in terms of patterns that might occur over taxonomical information. The three patterns considered are "Rename" (change of the label used for identifying the concept), "Delete" (removal of a concept from the ontology) and "Move" (change of the position of the concept in the ontology).

The above-mentioned works established definition and first steps towards ontology change managements and automation. However, to the best of the authors' knowledge there are less studies that analyze how such research was used in widely used ontologies developed in collaborative manner.

In this paper, we look at a subset of the change operations already proposed in the literature. First, we identify the main meta elements defined in the OWL 2 Web Ontology Language (RDF Semantics). In this study, we mainly focus on (a) rdfs:Class, (b) owl:Class, (c) rdf:Property, (d) owl:ObjectProperty, and (e) owl:DatatypeProperty.

2.1 Quality Issues Caused by Ontology Evolution

When parts in a given ontology change and evolve, they can cause impact on the entities described using them as well as on any other ontologies that are dependent on them [3].

On the one hand, when the changes cause inconsistencies in the ontology itself, they can be identified by validating, for instance, using a reasoner to check for inconsistencies. However, some problems are not so evident and are hard to detect automatically. For instance, if a redundant property is added that has the same semantics of an existing property (same meaning but different term/identifier), it decreases the conciseness quality but it is not straightforward to detect it.

On the other hand, when the changes in the ontology cause problems in data instances described using the ontology as well as in other ontologies that import the given ontology, it is hard to detect those problems. Most often the ontology developers are not aware of all the downstream users of the ontology and at the moment there are no well-established methods for automatically notifying the users of an ontology about changes in new versions of it. Though there are several approaches in the literature [10, 11], there is a lack of practical tools for determining the impact of changes of a given on ontology on data instances and dependent ontologies. If we take the analogy of software development, when a dependency changes in a software project compilation, errors, unit test and integration test failures reveal any potential problems due to changes in the dependency. Such practices are not yet well-established in the ontology engineering community.

In the literature, changes to the ontology are categorized into two main groups: structural changes and semantic changes [12]. On the one hand, structural changes include addition, deletion, renaming, splitting/merging a concept, property, restriction, or axiom. On the other hand, semantic changes include changes such as generalization or specialization of a concept or a property, increase or decrease of descriptiveness of a concept or a property, changes to restrictiveness of a property, etc. The semantic changes are result of one or more structural changes. For instance, adding or removing a sub-class relation could move a concept up or down in the concept hierarchy making it more generalized or specialized. In this paper, we mainly analyze the structural changes, which could lead to semantic changes.

Adding a new class or a property to an ontology generally does not cause problems in the data because there is no data loss. However, as it has been seen in a recent analysis of DBpedia[13], the new concepts and properties can introduce duplicates in the ontology. The duplicates (i.e., concepts or properties that have the same meaning but different identifiers) decrease the conciseness of the ontology and further make it harder to query or understand data. Adding a sub-class or sub-property can introduce some undesired effects if not done in a careful manner. When a new sub-property relationship is defined, one should take care that the domain and range restrictions of the sub-property are compliant with the ones of the super-property. Otherwise, undesired facts can be inferred by a reasoner. Further, if a data provider or an application used the given property without having this new hierarchical relation in mind, they can get undesired results.

3 Approach

This sections describes the approach followed in order to carry out the proposed analysis. For doing this, the section explains first the ontology selection process (Section 3.1) and then the data extraction implementation details (Section 3.2).

3.1 Ontology Selection

For the selection of the ontology several criteria were considered: (i) wide usage in the datasets in the LOD Cloud, (ii) availability of multiple versions of the ontology, (iii) whether the ontology was developed collaboratively, and (iv) the process used for collaborative development.

For checking the usage of the ontologies, we analyzed the data from the LOD Cloud State[1] which describes the usage of terms from a given ontology in the LOD Cloud. We have used the Linked Open Vocabularies (LOV[2]) dataset to verify the availability of multiple versions of a given vocabulary. We only considered ontologies with at least 5 versions and spanning at least a 3 year duration. Then we checked if the ontology was developed in a collaborative way with a large group of people. Finally, we tried to include ontologies developed following different processes of different organizations such as W3C, DBpedia Community, etc. The following ontologies were selected to be analyzed:

- **DBpedia**[3] which launched in 2007, and is maintained by the Free University of Berlin, the University of Leipzig and OpenLink Software, is a multilingual and cross-domain LOD dataset created by extracting structured information from Wikipedia. The DBpedia Ontology is a general ontology that covers multiple domains. It consists of a shallow class hierarchy of 320 classes and was created by manually deriving 170 classes from the most common Wikipedia infobox templates in the English edition. It also includes 750 properties resulting from mapping attributes from within these templates. There are 11 versions of DBpedia ontology.
- **Schema.org**[4] is a collaborative initiative that aims at promoting schemes for structured data on the web. It consists on a collection of schemes that are extended or proposed by an open community process. The schemes are a set of 'types' that have associated a set of properties. There are 26 versions of schema.org registered in LOV.
- **PROV-O**[5] is a W3C recommendation to represent and interchange provenance information generated in different systems and under different contexts. It can also be specialized to cover particular applications and domains. This ontology implements the PROV Data Model[6]. There are 8 versions of PROV-O registered in LOV.
- **FOAF**[7], which stands for "Frienf of a friend", is devoted to describe people and their relations on the Web. The core part describes characteristics of people and social groups and the social web part includes information about web accounts, address books and other web-based activities. FOAF is one of the most reused ontologies. There are 10 versions of FOAF registered in LOV.

3.2 Data extraction process

This section describes how this analysis was implemented. For extracting the versions of the different ontologies we used LOV. LOV stores versions of approx-

[1] http://lod-cloud.net/state/#terms
[2] http://lov.okfn.org
[3] http://dbpedia.org/ontology/
[4] http://schema.org/
[5] http://www.w3.org/ns/prov#
[6] https://www.w3.org/TR/2013/REC-prov-dm-20130430/
[7] http://xmlns.com/foaf/0.1/

imately 500 vocabularies and several versions for each. We used the querying facility in LOV to discover the versions of a given vocabulary and also to extract them.

For the analysis itself, we used the Loupe[14] tool for inspecting RDF datasets. Loupe allows to easily analyze datasets by creating virtual SPARQL endpoints via a Dockerized Virtuoso instance and a set of parameterized SPARQL queries.

4 Analysis

This section the results obtained from the study carried out over the four ontologies. In order to provide more context about the collaborative aspect of the development of each ontology, Section 4.1 provides some notions about how the community efforts involved in such developments. Then, Section 4.2 analyzes the results of our study for each of the ontologies.

4.1 Ontology Development Process

- The **DBpedia** community uses a wiki-based approach for developing the ontology. The DBpedia mappings wiki[8] provides guidance and templates for editing the DBpedia ontology mainly focusing on how to add classes and properties. Any community member with sufficient privileges can make modifications to the ontology. DBpedia provides some tool support for exploring the ontology and validating it after a modification. Until recently (February, 2015), it seems that DBpedia ontology issues have been tracked in an ad-hoc manner. But currently DBpedia uses an issue tracker[9] to raise and follow the progress of the DBpedia ontology issues.
- **PROV-O** was developed in accordance to the process[10] defined in the W3C for developing W3C recommendations. The W3C working group discussed the details of the ontology during the weekly teleconferences and also during the face-to-face meetings. Technical decisions are solved using consensus or voting-based process. Any member can raise issues which are recorded in the issue tracker and discussed in the group to come up with resolutions.
- **Schema.org** allows two types of extensions to be made by the community collaboratively, namely 'hosted' and 'external' extensions. In both cases typically subclasses and properties are added to the core schema. While hosted extensions are managed and reviewed as part of the schema.org project, the external ones are managed and reviewed by other groups.
- **FOAF** has been built by a collaborative effort of the users registered in the foaf-dev@lists.foaf-project.org mailing list. The vocabulary intends to be pragmatic and simple and to allow particular extensions. FOAF considers the stability of individual vocabulary terms, instead of the specification as a

[8] http://mappings.dbpedia.org/index.php/How_to_edit_the_DBpedia_Ontology
[9] https://github.com/dbpedia/ontology-tracker/
[10] https://www.w3.org/2015/Process-20150901/

whole. Terms progress through the categories 'unstable', 'testing' and 'stable'. Older terms might be considered 'archaic' which also allows them to become modern again.

4.2 Ontology change analysis

In the following, an analysis of the results obtained for each of the observed ontologies is provided. For each ontology, we provide a summary of changes in a table which illustrates the number (#) of classes and properties in each version of the ontology along with structural changes such as addition (+) or removal (-) of classes and properties compared to the previous version. Further, we also provide what is the effective change of both additions and removals, i.e., the difference (Δ) compared to the previous version.

More information about the ontology changes such as which classes and properties added in each version of the ontology is available in an external wiki page[11].

– The **DBpedia** ontology changes from version 3.2 (2008) until version 2016-04 (2016) are shown in Table 1. The ontology is gradually growing with respect to the number of classes and properties. However, when we look in detail, for instance, how many terms are added and removed, it can be seen that a large number of concepts and properties removed as well.
  When we analyze the classed removed, we can see classes such as "dbo:bibo:Book", or "dbo:Bullfighter" which do not follow the naming conventions for classes or duplicate of existing classes such as "dbo:Pornstar" which is a duplicate of "dbo:AdultActor" which already exists. The same with the properties that are removed. In version 3.5, we can see that 1198 properties (approximately half) are removed. This is because of a change in the convention of URI generation. For instance, before the version 3.5, there were a lot of properties in the form (http://dbpedia.org/ontology/Athlete/formerTeam) which were changed to (http://dbpedia.org/ontology/formerTeam). However, the reasons for removal or any other provenance metadata are not documented in any place. Further, we can see that from the 754 classes added to the ontology 514 classes (72%) are only used by less than 5 datasets[12].
  There are several key issues in the DBpedia ontology. First, being quite a large ontology, it is not sufficiently modular; this large monolithic nature hinders the proper reuse of it. As a consequence, the ontology editors introduce duplicate classes and properties for the same concepts and relations because they are not aware of or could not find the appropriate existing term. Another factor that contributes to the same problem is the minimum documentation in some classes or properties. Out of 2,849 properties, only 556 properties have a comment or description associated with them. Most of these problems exist because of the rather relaxed editorial process in minimal review and governance.

[11] `https://github.com/nandana/loupe/wiki/Ontology-Changes`
[12] `http://nandana.github.io/dbpedia/2015-10/class-langs.html`

Table 1. DBpedia - Classes and Properties

Version	OWL Class				RDF Property				Object Prop.			Datatype Prop.		
	#	Δ	(-)	(+)	#	Δ	(-)	(+)	#	(-)	(+)	#	(-)	(+)
3.2/3	**174**				**720**				**384**			**336**		
3.4	**204**	**30**	**-2**	**32**	**2168**	**1448**	**-271**	**1719**	**1144**	**-139**	**899**	**1024**	**-132**	**820**
3.5	**255**	**51**	**-6**	**57**	**1274**	**-894**	**-1198**	**304**	**601**	**-673**	**130**	**673**	**-525**	**174**
3.6	**272**	**17**	**0**	**17**	**1335**	**61**	**-37**	**98**	**629**	**-26**	**54**	**706**	**-11**	**44**
3.7	**319**	**47**	**-1**	**48**	**1643**	**308**	**-17**	**325**	**750**	**-6**	**127**	**893**	**-11**	**198**
3.8	**359**	**40**	**-1**	**41**	**1775**	**132**	**-3**	**135**	**800**	**-1**	**51**	**975**	**-2**	**84**
3.9	**529**	**170**	**-1**	**171**	**2333**	**558**	**-8**	**566**	**927**	**-6**	**133**	**1406**	**-2**	**433**
2014	**683**	**154**	**-5**	**159**	**2795**	**462**	**-46**	**508**	**1079**	**-9**	**161**	**1716**	**-37**	**347**
2015-04	**735**	**52**	**-5**	**57**	**2819**	**24**	**-103**	**127**	**1098**	**-23**	**42**	**1721**	**-80**	**85**
2015-10	**739**	**4**	**-5**	**9**	**2833**	**14**	**-9**	**23**	**1099**	**-3**	**4**	**1734**	**-6**	**19**
2016-04	**754**	**15**	**0**	**15**	**2849**	**16**	**-2**	**18**	**1103**	**-1**	**5**	**1746**	**-1**	**13**

- **Schema.org** is also a fairly large vocabulary that releases versions of the ontology frequently. Table 2 shows the ontology changes of 25 different versions of Schema.org since 2012. Similar to the DBpedia ontology, the Schema.org ontology is also gradually growing. However, in contrast to the DBpedia ontology we can notice a lower number of removals of concepts or properties. The only class that was removed "schema:OnSitePickup" was later reintroduced into the ontology. The four properties that have been removed ("oponent"[13], "supercedes"[14], "numberofEmployees"[15], and "isAcccessibleForFree"[16]) are all because of typos and the reasons for removing them are documented either in the issue tracker or the mailing list.
However, it is not the same when we analyze the subclass and subproperty relationships. As the new classes and properties are introduced and the hierarchy grows, the relations are changed frequently in the recent versions of schema.org. For instance, in the version "2015-05-12", the relationship "schema:BookSeries" is a subclass of "schema:Series" is removed. This is because a new hierarchy is introduced as "schema:CreativeWorkSeries" which is a subclass of "schema.org:CreativeWork". "schema:BookSeries" is moved to be a subclass of "schema:CreativeWorkSeries". These type of changes could have implications on the applications which are not aware of the change and expect the instances of "schema:BookSeries" to be a subclass of "schema:Series".
- **PROV-O** is a fairly small-sized ontology compared to DBpedia or Schema.org. It has a very focused scope and has been developed following the W3C process with close interactions between the working group members. Table 3 shows the ontology changes of seven versions of the PROV-O ontology from 2012 until 2015. In the case of PROV-O, we can see a lot of changes both as

[13] `https://lists.w3.org/Archives/Public/public-vocabs/2014Apr/0289.html`
[14] `https://github.com/schemaorg/schemaorg/issues/101`
[15] `https://github.com/schemaorg/schemaorg/issues/252`
[16] `https://github.com/schemaorg/schemaorg/issues/508`

Table 2. Schema.org - Classes and properties

Version		RDFS Class				RDF Property				Subclass			Subprop.		
Date	#	V.	Δ	(-)	(+)	#	Δ	(-)	(+)	#	(-)	(+)	#	(-)	(+)
2012-04-27	0.91	302				286				317			0		
2012-06-26	0.95	391	89	0	89	465	179	0	179	413	0	96	0	0	0
2012-07-26	0.97	393	2	0	2	466	1	0	1	415	0	2	0	0	0
2012-11-08	0.99	416	23	0	23	544	78	0	78	438	0	23	0	0	0
2013-04-05	1.0a	428	12	0	12	581	37	0	37	451	0	13	0	0	0
2013-07-24	1.0b	428	0	0	0	582	1	0	1	451	0	0	0	0	0
2013-08-07	1.0c	531	103	0	103	627	45	0	45	554	0	103	0	0	0
2013-11-19	1.0d	552	21	0	21	675	48	0	48	577	-1	24	0	0	0
2013-12-04	1.0e	558	6	0	6	711	36	0	36	583	0	6	0	0	0
2014-02-05	1.0f	558	0	0	0	711	0	0	0	583	0	0	0	0	0
2014-04-04	1.1	582	24	-1	25	777	66	0	66	607	1	25	1	0	1
2014-04-16	1.2	585	3	0	3	792	15	0	15	610	0	3	1	0	0
2014-05-16	1.4	585	0	0	0	794	2	-1	3	627	0	17	1	0	0
2014-05-27	1.5	585	0	0	0	798	4	0	4	627	0	0	1	0	0
2014-06-16	1.6	588	3	0	3	803	5	0	5	632	-2	7	36	-1	36
2014-07-08	1.7	589	1	0	1	806	3	0	3	633	0	1	36	0	0
2014-07-28	1.8	590	1	0	1	806	0	0	0	634	0	1	36	0	0
2014-08-19	1.9	593	3	0	3	816	10	0	10	636	-1	3	42	0	6
2014-09-12	1.91	593	0	0	0	816	0	-1	1	637	0	1	44	0	2
2014-12-11	1.92	618	25	0	25	878	62	0	62	663	-3	29	55	0	11
2015-02-04	1.93	620	2	0	2	891	13	-1	14	665	0	2	55	0	0
2015-05-12	2.0	638	18	0	18	965	74	0	74	676	-17	28	62	0	7
2015-08-06	2.1	645	7	0	7	976	11	-1	12	683	-2	9	63	0	1
2015-11-05	2.2	652	7	0	7	992	16	0	16	682	-10	9	69	0	6

additions and removals in the initial phase of the process but less changes in the later phases of the development. In fact, the last two versions of the ontology do not have additions or removals of concepts or properties and only improvements to the metadata.
When analyzing the classes added and removed in different versions, is seems they were done depending on the scope of the ontology agreed by the group. For example, there are several classes which are removed in the version "2012-07-24" such as "prov:Dictionary" or "Insertion" are readded in the version "2013-04-30". An expansion of scope can be seen in the version "2012-07-24" where one can find several concepts related to influence such as "prov:Influence", "prov:AgentInfluence", "prov:ActivityInfluence", and "prov:EntityInfluence".
- **FOAF** is the smallest and the oldest among the ontologies that have been analyzed. Table 4 shows the ontology changes of ten versions of the FOAF ontology since 2005 until 2014. Notably the FOAF vocabulary has not removed any classes or properties from the previous versions. Though it seems that two object properties were removed, those that refer to object properties which were transformed into datatype or annotation properties (foaf:membershipClass in v2007-01-14 and foaf:myersBriggs in v2009-12-15v2005-06-03).
With respect to subclass relationships, there are some of the relations that are removed. These are mainly subclass relations to external ontologies. For instance, in "2007-01-14" and "2009-12-15" the subclass relations to classes from "http://xmlns.com/wordnet/1.6/" are removed. In version "2014-01-14", subclass relation to "pim:Person" was removed.

Table 3. PROV-O - Classes and properties

Version	**OWL Class**				**RDF Property**				**Object Prop.**			**Datatype Prop.**		
	#	Δ	(-)	(+)	#	Δ	(-)	(+)	#	(-)	(+)	#	(-)	(+)
2012-05-03	**38**				**60**				**55**			**5**		
2012-07-24	**30**	**-8**	**-16**	**8**	**52**	**-8**	**-22**	**14**	**46**	**-20**	**11**	**6**	**-2**	**3**
2012-12-11	**30**	**0**	**-1**	**1**	**50**	**-2**	**-3**	**1**	**44**	**-3**	**1**	**6**	**0**	**0**
2013-03-12	**30**	**0**	**0**	**0**	**50**	**0**	**0**	**0**	**44**	**0**	**0**	**6**	**0**	**0**
2013-04-30	**50**	**20**	**0**	**20**	**68**	**18**	**0**	**18**	**59**	**0**	**15**	**9**	**0**	**3**
2014-06-07	**50**	**0**	**0**	**0**	**68**	**0**	**0**	**0**	**59**	**0**	**0**	**9**	**0**	**0**
2015-01-11	**50**	**0**	**0**	**0**	**68**	**0**	**0**	**0**	**59**	**0**	**0**	**9**	**0**	**0**

5 Conclusions and Future Work

Ontology changes are inevitable; the knowledge of a domain of interest changes and ontology evolution is needed for making sure that those changes are adapted by the ontology.

Table 4. FOAF - Classes and properties

Version		**OWL Class**		**RDF Property**		**Object Property**			**Datatype Property**		**Subclass**			**Subprop.**		
Date		#	(+)	#	(+)	#	(-)	(+)	#	(+)	#	(-)	(+)	#	(-)	(+)
2005-04-03		**12**		**52**		**0**			**0**		**15**			**10**		
2005-05-19		**12**	**0**	**52**	**0**	**0**	**0**	**0**	**0**	**0**	**15**	**0**	0	**10**	**0**	**0**
2005-06-03		**12**	**0**	**53**	**1**	**32**	**0**	**32**	**19**	**19**	**15**	**0**	0	**11**	**-1**	**2**
2007-01-14		**12**	**0**	**53**	**0**	**31**	**-1**	**0**	**20**	**1**	**16**	**-3**	4	**11**	**0**	**0**
2007-05-24	0.9	**12**	**0**	**53**	**0**	**31**	**0**	**0**	**20**	**0**	**16**	**0**	0	**11**	**0**	**0**
2007-10-02	0.91	**12**	**0**	**54**	**1**	**32**	**0**	**1**	**20**	**0**	**16**	**0**	0	**12**	**0**	**1**
2009-12-15	0.96	**12**	**0**	**58**	**4**	**32**	**-1**	**1**	**24**	**4**	**9**	**-7**	0	**12**	**0**	**0**
2010-01-01	0.97	**12**	**0**	**58**	**0**	**32**	**0**	0	**24**	0	**9**	**0**	0	**12**	**0**	**0**
2010-08-09	0.98	**13**	**1**	**62**	**4**	**33**	**0**	1	**27**	3	**10**	**0**	1	**13**	**0**	**1**
2014-01-14	0.99	**13**	**0**	**62**	**0**	**33**	**0**	0	**27**	0	**9**	**-1**	0	**13**	**0**	**0**

From the analysis carried out we can draw as a suggestion the incorporation to DBpedia of a rigorous editorial process and governance as well as a set of tools to facilitate the proper evolution of the ontology. As DBpedia makes use of crowdsourcing at many different stages and includes people from different levels and areas of knowledge about the ontology, having a proper editorial process becomes very important.

During the analysis, we have noticed that ontology evolution becomes more challenging when the ontology becomes larger and the development is performed in a collaborative manner. This is apparent in the DBpedia ontology; in this case, modularizing the ontology would help, so that it is easier to understand and extend it. For instance, in the DBpedia ontology there are more than 2,800 properties and a DBpedia member adding a new property needs to know all of them to avoid duplicates. However, if those were in a more modular manner it would help to avoid these problems.

In the same line, it would be helpful to have tools that could help in the process of ontology evolution. At the moment, there are tools that verify the consistency of an ontology after a change [15]. However, other functionalities and techniques such as autocomplete that look for similar existing terms in the ontology both using lexical or semantic similarity can be used to avoid duplicate terms.

Further, in evolving ontologies it is important to have provenance information about how the terms have evolved. Thus, in addition to the generic metadata, such as the label or description, it would be useful to add information such as when the term was added, who was the editor who added it, why a term has been deleted, etc. and references to any discussions about the reasons for the change.

As future work, we plan to extend our analysis to more change operations in ontologies including modifications to axioms, such as restrictions, and their impact on data quality. We also plan to produce a set of best practices for minimizing the data quality issues caused by ontology evolution based on the lessons learned.

Furthermore, we plan to do an extended analysis of the impact of the evolution of the ontologies discussed in this study using the datasets available in the Linked Open Data cloud that use such ontologies.

References

1. Gruber, T.R., et al.: A translation approach to portable ontology specifications. Knowledge acquisition **5**(2) (1993) 199–220
2. Noy, F.N., Klein, M.: Ontology evolution: Not the same as schema evolution. Knowledge and Information Systems **6**(4) (2004) 428–440
3. Stojanovic, L.: Methods and Tools for Ontology Evolution. Master's thesis, Karlsruhe Institute of Technology, Karlsruhe, Germany (2004)
4. Stojanovic, L., Maedche, A., Motik, B., Stojanovic, N.: User-driven ontology evolution management. In: International Conference on Knowledge Engineering and Knowledge Management, Springer (2002) 285–300
5. Palma, R.: Ontology metadata management in distributed environments. (2009)
6. Zablith, F.: Evolva: A comprehensive approach to ontology evolution. In: European Semantic Web Conference, Springer (2009) 944–948
7. Haase, P., Van Harmelen, F., Huang, Z., Stuckenschmidt, H., Sure, Y.: A framework for handling inconsistency in changing ontologies. In: International semantic web conference, Springer (2005) 353–367
8. Pinto, H.S., Staab, S., Tempich, C.: DILIGENT: Towards a fine-grained methodology for DIstributed, Loosely-controlled and evolvInG Engineering of oNTologies. In: Proceedings of the 16th Eureopean Conference on Artificial Intelligence, ECAI'2004, including Prestigious Applicants of Intelligent Systems, PAIS 2004, Valencia, Spain, August 22-27, 2004. (2004) 393–397
9. 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(EvoDyn-2012). Volume Vol-890. (2012)
10. 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(EvoDyn-2012). Volume Vol-890., CEUR-WS.org (2012)
11. Noy, N.F., Musen, M.A., et al.: PROMPTDIFF: A Fixed-Point Algorithm for Comparing Ontology Versions. Proceedings of the Eighteenth National Conference on Artificial Intelligence **2002** (2002) 744–750
12. Qin, L., Atluri, V.: Evaluating the validity of data instances against ontology evolution over the semantic web. Information and Software Technology **51**(1) (2009) 83–97
13. Mihindukulasooriya, N., Rico, M., García-Castro, R., Gómez-Pérez, A.: An Analysis of the Quality Issues of the Properties Available in the Spanish DBpedia. In: Proceeding of the 16th Conference of the Spanish Association for Articial Intelligence, Albacete, Spain, Springer International Publishing (2015) 198–209
14. Mihindukulasooriya, N., Poveda-Villalón, M., García-Castro, R., Gómez-Pérez, A.: Loupe-An Online Tool for Inspecting Datasets in the Linked Data Cloud. In: Demo at the 14th International Semantic Web Conference, Bethlehem, USA (2015)
15. Haase, P., Stojanovic, L.: Consistent evolution of owl ontologies. In: European Semantic Web Conference, Springer (2005) 182–197