Facilitating the Analysis of Ontology Differences

Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler

School of Computer Science,
University of Manchester,
Manchester, United Kingdom

Abstract. The analysis of changes between OWL ontologies (in the form of a *diff*) is an important service for ontology engineering. A purely syntactic analysis of changes is insufficient to distinguish between changes that have logical impact and those that do not. The current state of the art in semantic diffing ignores logically ineffectual changes and lacks any further characterisation of even significant changes. We present and demonstrate a diff method based on an exhaustive categorisation of effectual and ineffectual changes between ontologies. In order to verify the applicability of our approach we apply it to 88 versions of the National Cancer Institute (NCI) Thesaurus (NCIt), and 5 versions of SNOMED CT, demonstrating that all categories are realized throughout the corpus. Based on the outcome of these studies we argue that the devised categorisation of changes is helpful for ontology engineers and their understanding of changes carried out between ontologies.

Keywords: OWL, Ontologies, SNOMED CT, NCI Thesaurus, Diff, Ontology Evolution

1 Introduction

The comparison of ontologies is a valuable service whether used as a subroutine in a version control system or to support users in understanding changes. Different diff methods vary in their sensitivity to changes: e.g., a diff method based on character differences will find that two notationally distinct serializations of the same ontology are radically different. If a diff method is too sensitive to irrelevant changes then the user will be faced with determining which reported changes are actually significant. On the other hand, a hard requirement is that the diff method does not miss any change of significance.

The OWL 2 specification[1] defines a high level notion of syntactic equivalence, so-called "structural equivalence" (and thus the associated notion of *structural difference*), which abstracts from certain neglectable changes such as the order of axioms or concrete syntax. A different syntactic approach is that of an edit-based diff, wherein change records are produced within the ontology editor being used, thereby capturing the history and intent of change, as implemented in Swoop [8]. The diffs mentioned so far, as well as PROMPTDIFF [12] and

[1] http://www.w3.org/TR/2009/CR-owl2-syntax-20090611/

Bubastis [11], do not perform any further characterisation of reported changes (e.g., whether these have logical impact). This forces the user to analyse each change in the diff, and determine whether it affects the set of entailments of an ontology; thus whether the change is logically *effectual*. When analysing a set of changes it would be useful to not only distinguish between those logically effectual and ineffectual ones, but also to characterise reported changes according to their impact. Certain classes of ineffectual changes are neglectable, e.g., order of conjuncts in an axiom. However other ineffectual changes provide useful insights, such as the introduction of redundant or rewritten axioms. Semantic diffs, such as CEX [10] or ContentCVS [7], regard all ineffectual changes as neglectable, though in general knowing the relative proportions of effectual and meaningful ineffectual changes gives us a better understanding of what has changed between ontologies. For example, if most changes are meaningfully ineffectual it might be a sign of wasted effort, and therefore it would be useful to know if and why this happens. Contrariwise, while those changes might be logically ineffectual, other tools might be sensitive to the variant syntactic forms. So on the one hand, syntactic diffs report without distinction both effectual and ineffectual changes, and on the other hand semantic diffs do not present ineffectual changes.

A major problem with the output of change sets is that the user is given a (possibly large) set of axioms (or terms) to analyse, with no indication as to what kind of change each of them represents. A reasonable presentation of changes will cluster changes according to relevant properties. In this paper we discuss and elaborate on the diff method presented in [4],[2] referred to as *intentional difference*, which incorporates a categorisation of changes. This categorisation attempts to capture the impact of each change (e.g., whether it is a rewrite of another axiom). Aside from the intuitive appeal of categorising changes, our approach rectifies a problem with existing diffs, in that it aligns changes with what they are a change of. E.g., axioms in the category of 'rewritings' are shown together with the rewritten axioms.

For the purpose of verifying the suitability of the approach, we collected 88 versions of the National Cancer Institute (NCI) Thesaurus (NCIt) available in OWL, as well as 5 of the latest Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT)[3] versions, and conducted a diachronic study of each corpus. These studies aimed at showing the computational feasibility of our approach and confirming that the devised categories are realized throughout each corpus. Additionally we investigate whether this diff method helps us understand the evolution of both the NCIt and SNOMED CT. Moreover we demonstrate via walkthroughs how the categorisation can support change analysis by users: for the first walkthrough we use toy ontologies, while for the second we use two versions of the NCIt.

---

[2] In addition to material in [4], we carry out a cognitive walkthrough of a particular diff instance, discuss tool implementation, and provide a study of the Systematized Nomenclature of Medicine – Clinical Terms (SNOMED CT).

[3] `http://www.ihtsdo.org/index.php?id=545`

2 Preliminaries

We assume the reader to be reasonably familiar with ontologies and OWL, as well as the underlying description logics (DLs) [5], though detailed knowledge is not required. We do use the notion of entailment [2], which is identical to the standard first order logic entailment (an axiom α entailed by an ontology $\mathcal{O}$ is denoted $\mathcal{O} \models \alpha$). The *signature* of an ontology $\mathcal{O}$ is denoted $\widetilde{\mathcal{O}}$. The diff categories discussed in the paper are defined in [4], though the categories will be explained by means of examples.

3 Ontology Diffing

The problem of computing the difference between pairs of ontologies has been approached both syntactically and semantically. We distinguish two major aspects of ontology diffing: *(i)* the detection of changes, and *(ii)* the presentation of changes to the user. We note that most effort has been largely dedicated to *(i)*. It is often the case that the output *(ii)* of diff operations is the set of axioms or terms in the diff. While this may reflect the desired detection of change, it does not necessarily convey sufficient information to the user w.r.t. the impact of changes.

3.1 Related Work and Diff Desiderata

Within the detection of changes *(i)*, one would expect a preliminary distinction of axioms in the diff according to their logical effectuality. As such, a purely syntactic change analysis does not suffice to achieve this. Standard semantic diff tools treat all ineffectual changes as neglectable. This goes too far, for example: consider the case where two ontologies differ in a substantial number of axioms, but the axioms in the diff are only equivalences rewritten into subsumptions. Semantic diffs would point out that there are no differences, which may seem counter-intuitive to the user since a shallow inspection of the ontologies would reveal a discrepancy in number of axioms. In this case the ineffectual changes point to possibly unnecessary work. Though if these were intentional, then other developers should be aware of it rather than rewriting the axioms once again. So for this kind of change we would expect a more granular analysis to be helpful.

The second fundamental aspect of any diff is the presentation of changes to the user *(ii)*. Currently ontology diffs return as the output an unstructured, uncharacterised set of changes. As a consequence the task of change analysis is not particularly appealing. Consider the fact that the average diff size across the NCIt corpus is over 6,000 changes; relying on current diff methods for change analysis would be frightening, to say the least. At this point it would be useful to determine further properties of individual changes which might help the user understand changes. A change could relate to, e.g, a newly introduced term, or an adjustment to the class hierarchy. Whatever it may be, the current state-of-the-art in ontology diffing does not carry out such a characterisation of changes.

In terms of computability, one would expect an ontology diff to be efficiently computable for OWL 2 ontologies.[4] CEX [10], e.g., computes differences efficiently only for a fragment of OWL.[5]

Overall there is little tool support available to end-users for analysing and understanding differences between ontologies. Particularly when it comes to large ontologies, such as SNOMED CT, browsing through substantial change sets while inspecting two of its versions is not only tedious, but also requires above-average hardware to browse through SNOMED CT seamlessly. Based on the diff method described in [4], given two ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$ we obtain categorised sets of changes according to their apparent impact from $\mathcal{O}_1$ to $\mathcal{O}_2$, and vice-versa. This approach seems more reasonable to grasp what has actually changed from one ontology to another. In this paper we demonstrate via walkthroughs how one can have a better understanding of a change set based on such a categorisation, while for the actual definitions we refer the reader to [4]. We will describe how the output of this diff is obtained based on examples, as well as how different categories can be interpreted by users.

3.2 Intentional Diff Walkthrough

In order to demonstrate the usefulness of the diff categories, let us compare ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, defined in Table 1.

Table 1. Example ontologies.

$\mathcal{O}_1$		$\mathcal{O}_2$			
$\alpha_1:$	$A \sqsubseteq C$	$\beta_1:$	$A \sqsubseteq B \sqcup C$	$\beta_9:$	$F \sqsubseteq G \sqcap I$
$\alpha_2:$	$B \sqsubseteq C$	$\beta_2:$	$A \sqsubseteq B$	$\beta_{10}:$	$K \sqsubseteq \exists r.F$
$\alpha_3:$	$E \equiv D$	$\beta_3:$	$B \sqsubseteq C$	$\beta_{11}:$	$D \sqsubseteq F \sqcap \exists s.A$
$\alpha_4:$	$D \sqsubseteq F$	$\beta_4:$	$E \sqsubseteq D$	$\beta_{12}:$	$D \sqsubseteq F \sqcap \exists p.\top$
$\alpha_5:$	$F \sqsubseteq G$	$\beta_5:$	$D \sqsubseteq E$	$\beta_{13}:$	$B \sqsubseteq K$
$\alpha_6:$	$G \sqsubseteq H \sqcap \exists s.H$	$\beta_6:$	$E \sqsubseteq B \sqcup \exists r.C$		
$\alpha_7:$	$F \sqsubseteq I$	$\beta_7:$	$D \sqsubseteq E \sqcup G$		
$\alpha_8:$	$F \sqsubseteq G \sqcap I \sqcap J$	$\beta_8:$	$G \sqsubseteq \exists s.H \sqcap H$		

From $\mathcal{O}_1$ and $\mathcal{O}_2$ we have the following structural differences:

- ⋄ Additions$(\mathcal{O}_1, \mathcal{O}_2) = \{\beta_1, \beta_2, \beta_4, \beta_5, \beta_6, \beta_7, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}\}$
- ⋄ Removals$(\mathcal{O}_1, \mathcal{O}_2) = \{\alpha_1, \alpha_3, \alpha_4, \alpha_5, \alpha_7, \alpha_8\}$

Note that α_6 is not syntactically equal to β_8 ($\alpha_6 \neq \beta_8$), however they are structurally equivalent ($\alpha_6 \equiv_s \beta_8$). Therefore these axioms are not reported as changes. Given the sets of structural additions and removals from $\mathcal{O}_1$ to $\mathcal{O}_2$, we check which axioms in the Removals$(\mathcal{O}_1, \mathcal{O}_2)$ are entailed by $\mathcal{O}_2$ (ineffectual removals), and vice-versa for the Additions$(\mathcal{O}_1, \mathcal{O}_2)$. Thus we obtain a distinction between effectual and ineffectual changes, as follows:

[4] Based on DLs up to $\mathcal{SROIQ}$ [5].

[5] Specifically acyclic $\mathcal{EL}$-terminologies [10].

- ⋄ $\text{EffectualAdditions}(\mathcal{O}_1, \mathcal{O}_2) = \{\beta_2, \beta_6, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}\}$
- ⋄ $\text{EffectualRemovals}(\mathcal{O}_1, \mathcal{O}_2) = \{\alpha_8\}$
- ⋄ $\text{IneffectualAdditions}(\mathcal{O}_1, \mathcal{O}_2) = \{\beta_1, \beta_4, \beta_5, \beta_7, \beta_9\}$
- ⋄ $\text{IneffectualRemovals}(\mathcal{O}_1, \mathcal{O}_2) = \{\alpha_1, \alpha_3, \alpha_4, \alpha_5, \alpha_7\}$

There are several ineffectual changes in the change set, while effectual changes are mostly additions (and a single removal). The changes are categorised as shown in Table 2.

Table 2. Categorisation of axioms in diff($\mathcal{O}_1$, $\mathcal{O}_2$).

Removals		Axioms	Pairing
Ineffectual	Rewritten	α_3	$\{\beta_4, \beta_5\}$
	Strengthened	α_1	$\{\beta_2, \beta_3\}$
		α_4	$\{\beta_{11}\}, \{\beta_{12}\}$
	Redundant	α_1	$\{\beta_2, \beta_3\}$, $\{\beta_1, \beta_3\}$
		α_3	$\{\beta_4, \beta_5\}$
		α_5, α_7	$\{\beta_9\}$
E.	WeakeningRT	α_8	of β_9 ($\alpha_8 \models \beta_9$) and J is not in $\widetilde{\mathcal{O}}_2$
Additions		**Axioms**	**Pairing**
Ineffectual	Rewritten	β_9	$\{\alpha_5, \alpha_7\}$
	Weakened	β_9	$\{\alpha_8\}$
		β_7	$\{\alpha_3\}$, $\{\alpha_4, \alpha_5\}$
	Redundant	β_1	$\{\alpha_1\}$
		β_4, β_5	$\{\alpha_3\}$
		β_7	$\{\alpha_3\}$, $\{\alpha_4, \alpha_5\}$
		β_9	$\{\alpha_5, \alpha_7\}$
Effectual	Strenthening	β_{11}	of α_4 ($\beta_{11} \models \alpha_4$)
	StrentheningNT	β_{12}	of α_4 ($\beta_{12} \models \alpha_4$) and p is new in $\widetilde{\mathcal{O}}_1$
	NewDescription	β_{10}	K is defined in β_{10} and new in $\mathcal{O}_2$
	PureAddition	β_2	is a new axiom about shared terms (in $\widetilde{\mathcal{O}}_1 \cap \widetilde{\mathcal{O}}_2$)
	PureAdditionNT	β_6	is a new axiom involving a new term K in $\widetilde{\mathcal{O}}_2$
		β_{13}	is a new axiom involving a new term r in $\widetilde{\mathcal{O}}_2$

Let us consider two ineffectual additions; β_9 is a rewrite of $\{\alpha_7, \alpha_5\}$, as well as an avoided redundancy (i.e., had it been added to $\mathcal{O}_1$ it would be redundant). The axiom is also weakened, due to α_8. This may seem like an unintentional change, since now we face a loss of information regarding J, which is no longer mentioned in $\mathcal{O}_2$. Such a change may be worth revising. The axiom β_1 is redundant, since we have from $\mathcal{O}_1$ that $A \sqsubseteq C$, which is also entailed from $\mathcal{O}_2$. Therefore the user can dispose of this axiom.

Bear in mind that the existence of a rewritten axiom from $\mathcal{O}_1$ to $\mathcal{O}_2$ does not imply that the same holds in the opposite direction. This is applicable to all categories. Also we can have that an axiom is in more than one category. Consider axiom α_1; a justification $\mathcal{J}_1$ for α_1 is $\mathcal{J}_1 = \{\beta_2, \beta_3\}$, which indicates a strengthening (since we have that $\beta_2 \in \text{EffectualAdditions}(\mathcal{O}_1, \mathcal{O}_2)$), as well as a redundancy ($\beta_3 \in \mathcal{O}_1 \cap \mathcal{O}_2$). Another justification $\mathcal{J}_2 = \{\beta_1, \beta_3\}$ indicates a redundancy; $\beta_1 \in \text{IneffectualAdditions}(\mathcal{O}_1, \mathcal{O}_2)$.

In terms of effectual changes there is only one removal, and six additions. The effectual removal (α_8) represents a weakening of β_9 with retired terms (J is not mentioned in $\mathcal{O}_2$). In the analysis of the ineffectual changes it was already noted that axiom α_8 should be revised. The pure additions appear to be adjustments to the class hierarchy, some associated with new terms in $\mathcal{O}_2$. Both axioms β_{11} and β_{12} are strengthenings of α_4, which suggests that they could be merged, especially since there is intra-axiom redundancy. Finally there is a new term K in $\mathcal{O}_2$ being described via axiom β_{10}.

Generally speaking, with such a categorisation it becomes conceivably more intuitive to navigate and understand sets of changes. In addition to this, we gathered from the analysis of ineffectual changes useful information about the changes between $\mathcal{O}_1$ and $\mathcal{O}_2$, e.g., that axiom α_4 is strengthened in two distinct, yet partially superfluous axioms (β_{11} and β_{12}). Similarly we discover that axiom β_9 is weakened, from α_8, which should be reconsidered as we now have that $\mathcal{O}_2 \not\models F \sqsubseteq J$ (J becoming a retired term).

3.3 Implementation

The algorithm to compute the diff and its categories is straightforwardly derivable from the definitions in [4], and heavily relies on decision procedures for entailments [6], justification finding [9], and module extraction algorithms [1]. The diff itself is implemented in Java, relying on the OWL API, and is made available on the Web as a Java Servlet.[6] The output of each diff is an XML file,[7] containing the axioms in the diff and their respective categories. In order to present this output in a more sensible way, we use an XSL Transformation (XSLT)[8] and a Cascading Style Sheet (CSS),[9] which given the XML file presents the axioms in the diff according to their respective categories. The resulting web page also contains the pairing of each difference, as well as the new or retired terms used, where applicable. An example of the Servlet output is shown in Figure 1, where axioms are represented in Manchester Syntax.[10]

[6] `http://owl.cs.manchester.ac.uk/diff`
[7] `http://www.w3.org/XML/`
[8] `http://www.w3.org/TR/xslt`
[9] `http://www.w3.org/Style/CSS/`
[10] `http://www.w3.org/TR/owl2-manchester-syntax/`

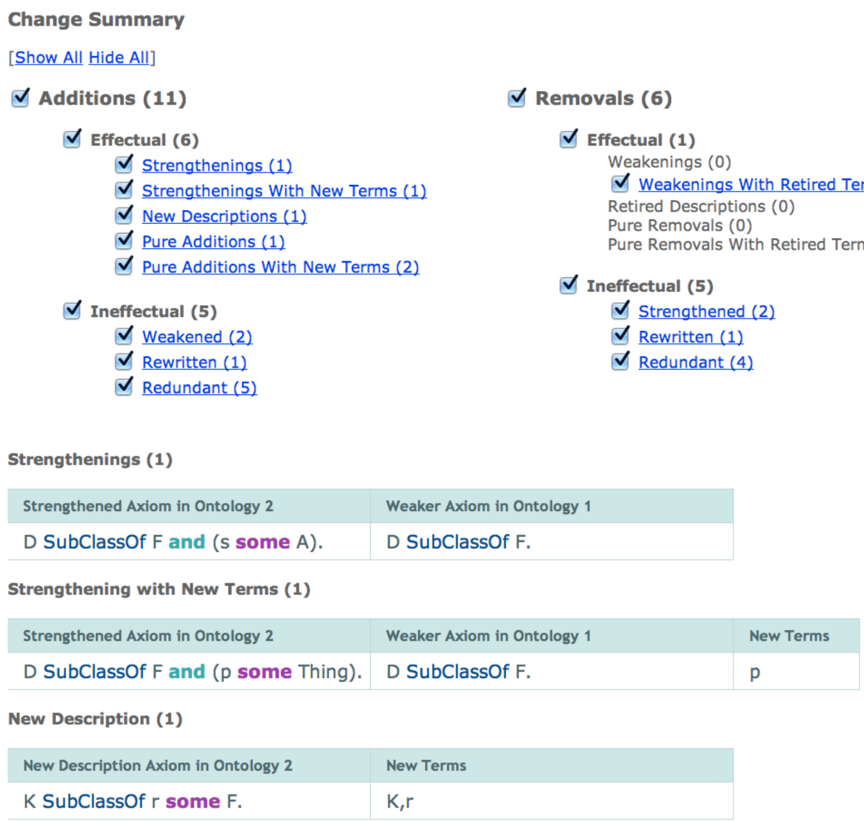

Fig. 1. Output of the diff as a HTML web page (Manchester Syntax).

4 Case Studies

Throughout the case studies we have excluded a class of ineffectual changes: changes to annotations. Since there are ontologies whose major focus is the annotations, as the NCIt, these represent a topic of study in themselves. E.g., in the NCIt the average proportion of axiomatic changes throughout the corpus is 15%, while the remaining are annotation changes. But for the purposes of this paper we focus on axiom changes only. The experiment machine is an Intel Xeon Quad-Core 3.20GHz, with 16Gb DDR3 RAM dedicated to the Java Virtual Machine (JVM v1.5). The system runs Mac OS X 10.6.8, and all tests were run using the OWL API (v3.1.0).

The NCIt archive[11] contains 88 versions of the ontology in OWL format, two of which were unparsable (releases 05.03F and 05.04d) with the OWL API,[12]

[11] http://evs.nci.nih.gov/ftp1/NCI_Thesaurus
[12] http://owlapi.sourceforge.net/

and consequently Protégé.[13] The test data is published on Google Public Data Explorer,[14] and can be visualised at `http://bit.ly/leZ6fM`.

SNOMED CT is not readily available in OWL, however IHTSDO[15] supplies a Perl[16] script to transform the published concept and stated relationships tables into OWL or KRSS formats. We collected all 5 International Releases of SNOMED CT that can be converted into OWL via this script, since January 2009 through to January 2011 (the versions before 2009 are not published with this transformation script into OWL). Upon executing the conversion of SNOMED CT releases into OWL we discovered a discrepancy between the OWL and KRSS transformation outputs. Specifically in the January and July 2009 releases (bundled with version 1.1 of the Perl script), the OWL and KRSS serializations differ in over 100,000 subclass axioms. The OWL transformation of the January 2009 release contains 115,941 more subclass axioms than the KRSS output, while in the July 2009 release the OWL version has 113,665 more subclass axioms. This situation no longer presents itself with subsequent versions of the transformation script; the 2010 releases are bundled with version 2.0 of the script, while the 2011 release comes with version 2.1. Both these versions yield the same OWL and KRSS outputs. By applying version 2.1 of the Perl transformation script to the 2009 releases we get a match on the output (of OWL and KRSS). As such, we used version 2.1 of the transformation script on the 2009 releases in order to carry out this study.

4.1 NCIt Diff Results

As reported in [3], the logical difference throughout the NCIt time-line consists of an average of 87% effectual changes, while the remaining are ineffectual (see Figures 2 and 3 and Table 3). There are more additions than removals throughout the corpus, with an average of 60% additions versus 40% removals, which is hardly surprising for a constantly evolving ontology.

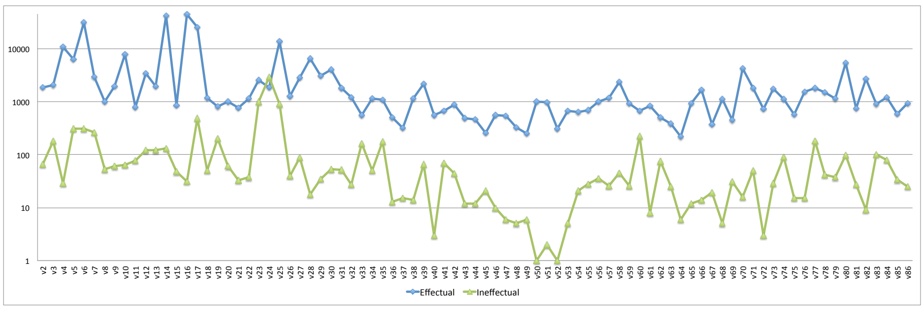

Fig. 2. Effectual vs ineffectual additions (logarithmic scale, number of axioms).

[13] `http://protege.stanford.edu/`
[14] `http://www.google.com/publicdata/home`
[15] `http://www.ihtsdo.org/`
[16] `http://www.perl.org/`

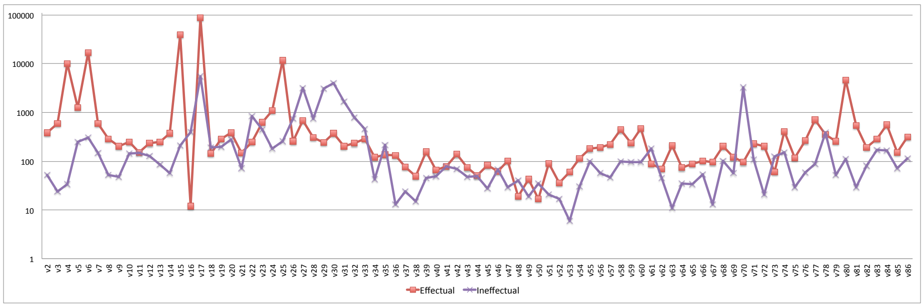

Fig. 3. Effectual vs ineffectual removals (logarithmic scale, number of axioms).

Throughout the corpus there is a high number of ineffectual removals, with an average of 35% of all removals. Out of these ineffectual removals 92% turned out to be strengthened axioms, which indicates a continuous refining of information throughout the corpus. There is a high number of removed redundancies as well, constituting 49% of ineffectual removals. This tells us that there is some pruning of redundant information going on in the corpus. On average 5% of additions are ineffectual; among these, 73% are added redundancies, and 82% are weakened axioms. The latter occur typically due to adjustments in the class hierarchy. Despite being a high percentage, the number of ineffectual additions is generally low (Table 3). In cases where the number of ineffectual changes is quite high (e.g., $\mathcal{O}_{24}$ where 52% of changes are ineffectual, $\mathcal{O}_{27}$, $\mathcal{O}_{29}$ and $\mathcal{O}_{30}$ with 48% each) note that semantic diffs would significantly understate the amount of activity performed. While structural diff captures this, it does not analyse the logical impact of such changes.

Table 3. Results of $\mathrm{diff}(\mathcal{O}_i, \mathcal{O}_{i+1})$, for $1 \leqslant i \leqslant 86$.

Change Type	**Effectual**				**Ineffectual**	
	Axiom Terms	*Shared*	*New*	*Retired*		
Additions	*Total*	285,524			*Total*	9,997
	Strengthening	4,649	4,948	N/A	*Rewrite*	115
	New Description	N/A	170,095	N/A	*Weakened*	8,576
	Pure Addition	36,499	69,333	N/A	*Redundant*	4,186
Removals	*Total*	102,515			*Total*	31,727
	Weakening	1,122	N/A	508	*Rewrite*	114
	Retired Description	N/A	N/A	52,229	*Strengthened*	30,370
	Pure Removal	25,022	N/A	23,634	*Redundant*	8,012

We also identified a number of rewrites in the corpus. Particularly in $\mathcal{O}_{33}$, there are 227 rewritten axioms. Upon inspecting the rewritten axioms, we noticed that these changes are not only syntactic but also trivial and easily detected. While ideally the underlying structural diff would not include these, at least with our categorisation and alignment with source axioms, it is easy to spot and recognise the triviality.

Among the categories of effectual changes we discovered that the majority of these are new and retired descriptions. In terms of effectual additions the average of new descriptions is 60%, while retired descriptions average 51% of effectual removals. This high number of new descriptions is unsurprising, as the terminology keeps increasing. Despite the high values of retired descriptions, it does not mean that such concepts are deleted, it could instead suggest concept renamings. Strengthenings average around 4% of all additions (with and without new terms), indicating refinements of concepts with additional constraints. It is natural that upon introducing new terms, others need to be re-described, thus explaining the strengthenings with new terms. There are not as many weakenings in the corpus as there are strengthenings. This tells us that typically there is not much reduction of information from version to version. The average of weakenings (with or without retired terms) throughout is below 2% of all removals. Pure additions account for 37% of all additions, divided between 24% of additions with new terms and 13% without. Typically pure changes with shared terms suggest adjustments to the class hierarchy, while pure additions involving new terms point to the insertion of said terms and subsequent re-adjustment of the hierarchy accordingly. The average of pure removals throughout the corpus is 47%, split between 23% with retired terms and 24% without.

4.2 SNOMED CT Diff Results

Throughout the 5 versions of SNOMED CT there are typically more effectual changes (74%) than ineffectual (26%), as shown in Figure 4. However, within the removals there is a high number of ineffectual changes (10,214), averaging 37% of all removals (see Table 4). In the additions there are on average more effectual changes (84%) than ineffectual (16%).

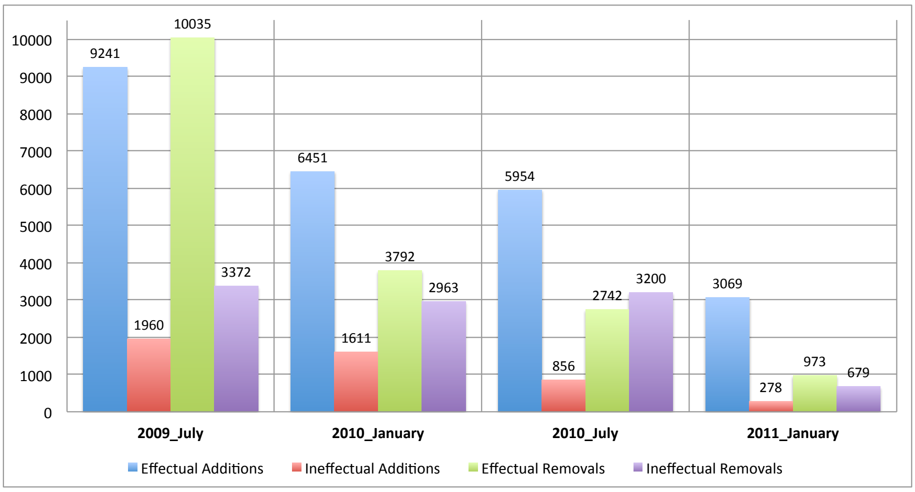

Fig. 4. Breakdown of effectual and ineffectual changes (number of axioms).

The majority of effectual additions (43% of all changes) are pure additions, suggesting numerous alterations to the class hierarchy, possibly as a result of

Table 4. Results of diff($\mathcal{O}_i$, $\mathcal{O}_{i+1}$), for $1 \leqslant i \leqslant 5$.

Change Type	**Effectual**				**Ineffectual**	
	Axiom Terms	*Shared*	*New*	*Retired*		
Additions	*Total*	24715			*Total*	4705
	Strengthening	4657	481	N/A	*Rewrite*	0
	New Description	N/A	6266	N/A	*Weakened*	3897
	Pure Addition	9960	3351	N/A	*Redundant*	2561
Removals	*Total*	17542			*Total*	10214
	Weakening	1809	N/A	70	*Rewrite*	0
	Retired Description	N/A	N/A	4602	*Strengthened*	9386
	Pure Removal	9836	N/A	1225	*Redundant*	4395

the addition of new terms, as there is also a high number of pure additions involving new terms. Naturally when introducing a new, non-leaf term in the class hierarchy it causes shifts throughout one or more branches. Thus, given the high number of new descriptions (6,266) throughout SNOMED CT, the detection of many pure additions is not surprising.

Within the effectual removals (31% of all changes) we see a similar pattern as in the additions: the majority (63%) of effectual removals are pure removals, some with retired terms. There is also a high number of retired descriptions, suggesting that as terms are retired (either deleted or declared as 'retired concepts') this causes modifications to the class hierarchy. This also explains the high number of pure additions, with shared and retired terms. Weakenings are not as common, amounting to 10% of effectual removals.

In terms of ineffectual additions (8% of all changes), there are mostly weakened axioms (3,897), as well as a high number of avoided redundancies (2,561). The latter are derived from modifications to the class hierarchy, which in this case can be ignored. Had there been redundancies it would be advisable to inspect them, and possibly dispose of these from the ontology. The weakened axioms, on the other hand, should be verified for correctness, as they reveal a loss of constraints w.r.t. the terms described in such axioms.

The number of ineffectual removals (18% of all changes) is much higher than the additions, and consists of mostly strengthened axioms (9,386). This suggests that a constant tightening of the ontology is taking place; despite the fact that these axioms are removed, it is only due to the introduction of stronger constraints on the meaning of such axioms. The remaining ineffectual removals are avoided redundancies (4,395).

Overall we see a good balance of effectual additions vs removals, and a lot more ineffectual removals than additions. By inspecting the different types of changes throughout the evolution of SNOMED CT, we can single out particular occurrences in more detail; e.g., when a new term is introduced in the class hierarchy (new description), it will typically lead to adjustments to the hierarchy (pure additions) involving both shared terms which have to be switched around, and the new term itself (or other new ones). Using a standard syntactic or semantic diff such an analysis would be impractical, and much of the work would be left up to the user (such as checking which changes have logical impact).

With this categorisation even a user who is not a domain expert or an ontology engineer can spot trends in the diff between ontology versions, as we did here with SNOMED CT.

4.3 NCIt vs SNOMED CT

Note that in SNOMED CT there is a bigger proportion of ineffectual changes than in the NCIt, and the number of certain types of change is similar in both cases, e.g., within ineffectual removals there are mostly strengthened axioms, followed by redundancies. However, in the case of SNOMED CT, there are only avoided redundancies, while in the NCIt there are redundancies which could be disposed of. Clearly certain ineffectual changes are in fact "refactorings", albeit in the case of strengthened and weakened axioms the refactoring would have to be of a set of axioms rather than a single axiom. Thus a strengthened axiom does not necessarily mean strengthening of the ontology, since the change might either introduce a redundancy or redistribute information from other axioms. Consider an ontology $\mathcal{O}_1 = \{\alpha_1 : A \sqsubseteq B, \alpha_2 : A \sqsubseteq C\}$, and a change of α_1 into $A \sqsubseteq B \sqcap C$. The axiom α_1 was strengthened, but the resulting ontology $\mathcal{O}_2 = \{\alpha_1 : A \sqsubseteq B \sqcap C, \alpha_2 : A \sqsubseteq C\}$ was not. However, if we change $\alpha_2 \in \mathcal{O}_2$ into $A \sqsubseteq C \sqcap D$, we can say both the axiom α_2 and the ontology $\mathcal{O}_2$ are strengthened.

In terms of effectual changes there are mostly new and retired descriptions in the NCIt, while in SNOMED CT the majority of effectual changes are pure additions or removals. So we see that a major focus in SNOMED CT seems to be class hierarchy oriented, with many adjustments going on. In the NCIt, on the other hand, we see that there are a lot of new terms being introduced, and since we know that terms are never deleted, the high number of retired descriptions suggests there are many renamings of term names.

4.4 NCIt Walkthrough

In order to elaborate on the potential usefulness of the devised categories, let us look at a particular comparison, diff($\mathcal{O}_{32}, \mathcal{O}_{33}$), and walk through the output of the diff from the perspective of a user. The output of diff($\mathcal{O}_{32}, \mathcal{O}_{33}$) is outlined in Table 5. A user whose role is to assure the overall progress and quality of changes is particularly interested in ensuring that effectual changes are appropriate, and may also be interested in understanding why committed changes have no logical effect. Looking at the diff, the user immediately sees that the changes are balanced between effectual and ineffectual, with most of the action in the effectual additions and ineffectual removals. Being more concerned with added information, the user begins by inspecting the effectual additions. Particularly by looking through new descriptions, the user is immediately aware of the new terms that have been introduced, as the following axioms demonstrate:

α_1: $Oxidized\_Glutathione \sqsubseteq Protective\_Agent$
α_2: $CHP\text{-}HER\text{-}2\_Peptide\_Vaccine \sqsubseteq \exists Chemical\_Or\_Drug\_Has\_Physiologic\_Effect.Immunopotentiation\_Effect$

Table 5. Results of diff($\mathcal{O}_{32}$, $\mathcal{O}_{33}$).

Change Type	Effectual				Ineffectual	
	Axiom Terms	*Shared*	*New*	*Retired*		
Additions	*Total*	553			*Total*	164
	Strengthening	105	8	N/A	*Rewrite*	114
	New Description	N/A	110	N/A	*Weakened*	30
	Pure Addition	300	30	N/A	*Redundant*	157
Removals	*Total*	284			*Total*	456
	Weakening	2	N/A	0	*Rewrite*	113
	Retired Description	N/A	N/A	8	*Strengthened*	332
	Pure Removal	269	N/A	5	*Redundant*	202

The axiom α_1 introduces the term *Oxidized_Glutathione*, while α_2 introduces *CHP-HER-2_Peptide_Vaccine*. The user can restrict his/her attention to those additions that concern his domain expertise, thus being also interested in strengthenings within this domain. Let us look at the latter: axiom α_3 is a strengthening of β_3, and α_4 a strengthening with new terms (*Anterior_Foramen_Magnum*) of β_4, as follows:

α_3: *Skin_Appendage_Adenoma* $\equiv$ *Adenoma* $\sqcap$ *Benign_Epithelial_Skin_Neoplasm* $\sqcap$ *Benign_Skin_Appendage_Neoplasm*
β_3: *Skin_Appendage_Adenoma* $\sqsubseteq$ *Adenoma*

α_4: *Anterior_Foramen_Magnum_Meningioma* $\equiv$ *Foramen_Magnum_Meningioma* $\sqcap \forall$*Disease_Has_Primary_Anatomic_Site.Anterior_Foramen_Magnum*
β_4: *Anterior_Foramen_Magnum_Meningioma* $\sqsubseteq$ *Foramen_Magnum_Meningioma*

The user sees in both axioms an addition of information w.r.t. the previous version, and can narrow down the search to axioms concerning his/her domain area to verify correctness. In the same manner the user goes through those alterations with and without new terms. The following axiom α_5 is an alteration, and α_6 is an alteration with a new term *c_Concept_Status*:

α_5: *Epidermal_Involvement* $\sqsubseteq$ *Cutaneous_Involvement*
α_6: *UMLS_Cross-Reference_Concept* $\sqsubseteq$ *c_Concept_Status*

This type of axiom generally indicates adjustments to the class hierarchy, which the user may want to verify in his respective domain area. Switching over to ineffectual changes, the user begins by inspecting the rewrites. Since rewritten axioms are supposed to convey the same logical meaning in both $\mathcal{O}_{32}$ and $\mathcal{O}_{33}$, the user wants to understand why these are presented by the diff. The users finds axiom α_7 rewritten into β_7, as follows:

α_7: $Renal\_Cell\_Carcinoma\_with\_t\_X\_1\_p11\_p34 \equiv Xp11\_Translocation\text{-}Related\_Renal\_Cell\_Carcinoma \sqcap ((\forall Disease\_Has\_Cytogenetic\_Abnormality.t\_X\_1\_p11\_p34) \sqcap (\forall Disease\_Has\_Molecular\_Abnormality.PSF\text{-}TFE3\_Fusion\_Protein\_Expression))$

β_7: $Renal\_Cell\_Carcinoma\_with\_t\_X\_1\_p11\_p34 \equiv Xp11\_Translocation\text{-}Related\_Renal\_Cell\_Carcinoma \sqcap (\forall Disease\_Has\_Cytogenetic\_Abnormality.t\_X\_1\_p11\_p34) \sqcap (\forall Disease\_Has\_Molecular\_Abnormality.PSF\text{-}TFE3\_Fusion\_Protein\_Expression)$

The change from α_7 to β_7 is purely syntactic, and thus both axioms carry the same logical meaning. As discussed in Section 4.1, we find here a particularity of OWL's notion of structural equivalence that could be refined. Seeing as the inspected rewrites exhibit this form, the user skips further analysis of this type of change. Next the user looks at redundant axioms; since redundancies do not add any logical meaning the user may want to prune them. Such redundancies are guaranteed not to alter the semantics of the ontology alone, and so can be immediately disposed of. The user finds redundancies of similar form to α_8, with a justification $\mathcal{J}_{\alpha_8} \subseteq \mathcal{O}_{33}$, as follows:

α_8: $CS\text{-}1008 \sqsubseteq \exists Chemical\_Or\_Drug\_Has\_Mechanism\_Of\_Action.Antigen\_Binding\_Interaction$

$\mathcal{J}_{\alpha_8} = \{CS\text{-}1008 \sqsubseteq Monoclonal\_Antibody, Monoclonal\_Antibody \sqsubseteq \exists Chemical\_Or\_Drug\_Has\_Mechanism\_Of\_Action.Antigen\_Binding\_Interaction\}$

Upon finding these, the user proceeds to remove them from the ontology. Meantime it would be helpful to investigate with the corresponding developers why these redundancies were being added in the first place. The user then carries on inspecting avoided redundancies; among these the user comes across several of the same kind as α_9, with corresponding justification $\mathcal{J}_{\alpha_9} \subseteq \mathcal{O}_{32}$:

α_9: $Lactic\_Acid\_L \sqsubseteq Industrial\_Product$

$\mathcal{J}_{\alpha_9} = \{Lactic\_Acid\_L \sqsubseteq Pharmaceutical\_Excipient, Pharmaceutical\_Excipient \sqsubseteq Industrial\_Aid, Industrial\_Aid \sqsubseteq Industrial\_Product\}$

This change reveals to the user that some adjustments to the class hierarchy took place involving terms in $\widetilde{\alpha_9}$. The user can inspect the class hierarchy to confirm this, and recognise that such changes would be redundant in the previous ontology. However, while α_9 is redundant w.r.t. to $\mathcal{O}_{32}$ it is still the case that removing it from $\mathcal{O}_{33}$ would cause loss of information. As such, the user leaves these avoided redundancies behind and begins to verify weakened axioms. A weakened axiom implies a reduction of constraints on models of the ontology, and so should be carefully reviewed. The user comes across α_{10} with a justification $\mathcal{J}_{\alpha_{10}} \subseteq \mathcal{O}_{32}$:

α_{10}: $\quad Metachronous\_Wilms\_Tumor \sqsubseteq Renal\_Wilms\_Tumor$

$\mathcal{J}_{\alpha_{10}} = \{Metachronous\_Wilms\_Tumor \equiv Metachronous\_Malignant\_Neoplasm \sqcap Renal\_Wilms\_Tumor \sqcap \exists Disease\_May\_Have\_Molecular\_Abnormality.WT\text{-}1\_Tumor\text{-}Suppressor\_Gene\_Inactivation \sqcap \forall Disease\_ Has\_Finding.Bilateral\_Disease\}$

The user realises that α_{10} is indeed weaker than the axiom in $\mathcal{J}_{\alpha_{10}} \subseteq \mathcal{O}_{32}$, and not being the domain expert dispatches this change to the respective expert for review and confirmation of the original intent. Subsequently the user inspects the strengthened axioms, since these are more constricting in $\mathcal{O}_{33}$ than they were in $\mathcal{O}_{32}$. Consider axiom α_{11} and its justification $\mathcal{J}_{\alpha_{11}} \subseteq \mathcal{O}_{33}$:

α_{11}: $\quad Sporadic\_Cylindroma \sqsubseteq Cylindroma$

$\mathcal{J}_{\alpha_{11}} = \{Sporadic\_Cylindroma \equiv Cylindroma \sqcap \forall Disease\_Has\_Finding.Non\text{-}Hereditary\_Lesion\}$

Given changes of this type it may be desirable to confirm them, and so the user takes the same action as for weakened axioms and delegates them to appropriate domain experts while inspecting other changes of interest.

We have shown here an example of how such categorisation makes change analysis far more manageable, as opposed to having users inspect a whole set of axioms or terms. The devised categories allow users to focus on specific types of change, and particularly see what these are a change of (in the previous or subsequent version). Moreover a categorisation of this type produces a reasonable division of labour, thus making collaboration efforts more practical.

5 Discussion and Outlook

The diachronic studies of the NCIt and SNOMED CT revealed that all the categories of changes occur throughout each corpus, providing indications as to the impact of changes. We showed that such a categorisation of change sets is useful for change analysis, with particular benefits for division of labour by ontology engineers. By means of this categorisation we can group changes according to their impact, allowing users to shift their attention to specific types of changes, rather than going through a change set while inspecting both ontologies. With our correspondence of changes between ontologies we can show the changed axioms and what they are a change of. Consequently, by analysing changes in this way there is no need for constantly having to inspect ontologies manually. As such, we can support users in understanding the impact of their changes (or lack thereof), and refine these before publishing newer versions.

We found that ineffectual changes account for a significant amount of changes throughout the NCIt, as well as in SNOMED CT with an even higher proportion. Despite the fact that semantic diffs ignore these changes in their output, we show that they provide helpful modelling insights, and thus are worth examining. For instance, we discovered a high number of redundant axioms in the NCIt, some of which could be disposed of. Also we found a number of structurally distinct ineffectual changes that are clearly neglectable: the rewrites in the NCIt. These indicate the need for improvement of the underlying diff. In general the inspection of ineffectual changes is helpful to prevent, e.g., re-doing work or introducing redundancy. Relying on semantic diffs one would be missing out on these meaningful ineffectual changes, which in turn could help users recognise the impact of certain types of change.

The next step in our study is to evaluate the diff tool with users. In particular we expect to confirm that the categorisation helps users in understanding changes between ontologies. In terms of further analysis of the NCIt and SNOMED CT we intend to inspect the history of axioms, as in checking the progress of each axiom throughout the corpus since it was introduced (e.g., changes in constructors or strengthenings that the axiom went through). Another future survey involves checking for patterns of change throughout the corpus, e.g., if all strengthenings exhibit a common form.

References

1. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. of Artificial Intelligence Research 31 (2008)
2. Cuenca Grau, B., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. J. of Web Semantics (2008)
3. Gonçalves, R.S., Parsia, B., Sattler, U.: Analysing the evolution of the NCI Thesaurus. In: Proc. of DL 2011. pp. 147 – 157. CEUR Workshop Proceedings (2011)
4. Gonçalves, R.S., Parsia, B., Sattler, U.: Categorising logical differences between OWL ontologies. In: Proc. of CIKM 2011 (2011)
5. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. of KR-06 (2006)
6. Horrocks, I., Sattler, U.: A tableaux decision procedure for $\mathcal{SHOIQ}$. J. of Automated Reasoning 39, 249–276 (2007)
7. Jiménez-Ruiz, E., Cuenca Grau, B., Horrocks, I., Berlanga Llavori, R.: Supporting concurrent ontology development: Framework, algorithms and tool. Data and Knowledge Engineering 70(1), 146–164 (2011)
8. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B., Hendler, J.: Swoop: A Web ontology editing browser. J. of Web Semantics 4(2) (2006)
9. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Proc. of ISWC/ASWC-07 (2007)
10. Konev, B., Walther, D., Wolter, F.: The logical difference problem for description logic terminologies. In: IJCAR-08. LNCS, vol. 5195. Springer-Verlag (2008)
11. Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., Zhukova, A., Brazma, A., Parkinson, H.E.: Modeling sample variables with an experimental factor ontology. Bioinformatics 26(8), 1112–1118 (2010)
12. Noy, N.F., Musen, M.A.: PROMPTDIFF: A fixed-point algorithm for comparing ontology versions. In: Proc. of AAAI-02 (2002)