Ecco: A Hybrid Diff Tool for OWL 2 ontologies

Rafael S. Gonçalves, Bijan Parsia, and Ulrike Sattler

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

Abstract. The detection and presentation of changes between OWL ontologies (in the form of a *diff*) is an important service for ontology engineering, being an active research topic. In this paper, we present a diff tool that incorporates structural and semantic techniques in order to, firstly, distinguish effectual and ineffectual changes between ontologies and, secondly, align and categorise those changes according to their impact. Such a categorisation of changes is shown to facilitate the navigation through, and analysis of change sets. The tool is made available as a web-based application, as well as a standalone command-line tool. Both of these output an XML change set file and a transformation into HTML, which allows users to browse through and focus on those changes of utmost interest using a web-browser.

1 Introduction

Detecting and presenting changes between any two documents (so-called *diff*) is an essential service that is hardly confined to software engineering. While regular textual diffs, such as UNIX's diff function, rely on the assumption that order matters, in ontologies that no longer holds; OWL [3] does not impose a systematized ordering of axioms, but instead defines a higher level notion of *syntactic equivalence* (so-called *structural equivalence* [13], and associated notion *structural difference*). In turn, this notion provides a basis for ignoring certain types of negligible changes, such as the order of axioms or concrete syntax (e.g., OWL/XML compared to RDF/XML serialisations of the same ontology).

There are a variety of diff services based on structural equivalence [7, 12, 11], which distinguish additions and removals, and subsequently align axiom changes with those class names found on the left-hand side of the axiom. However, no further characterisation of changes is typically carried out, e.g., whether changes produce any logical effect (thus *effectual*) or not (*ineffectual*), as conducted in [11], or whether there exist any relations or additional properties of changes that might help users analyse and understand them. Aside from this, such diffs lack a standard and essential feature of any diff: an alignment between the source and target of a change. This kind of data can be collected at development time via edit-based diffs, as implemented in SWOOP [9], although if there are no such change records then a post facto change analysis is impossible.

In this paper we present a diff tool that incorporates structural and semantic techniques to, firstly, distinguish which additions and removals (obtained via structural difference) are effectual or ineffectual, and, secondly, find the source of each change (where attainable), which in turn allows us to categorise and align (source with target of) changes between two ontologies. The categories follow from the different kinds of impact a change can have, e.g., by further constraining an axiom we can make it "stronger", and the relation between this stronger axiom and its preceding version is made explicit by our categorisation, and suitably presented by our tool. The latter is available either as a web-based application, or a command-line tool with more advanced features.

2 Preliminaries

We assume the reader to be reasonably familiar with ontologies and OWL, as well as the underlying description logics (DLs) [1], though detailed knowledge is not required. The diff categories discussed in the paper are defined in [4], and will be briefly discussed and exemplified. When comparing two ontologies we refer to them as $\mathcal{O}_1$ and $\mathcal{O}_2$, and their *signatures*, i.e., the set of entity (class, property and individual) names occurring in them, as $\widetilde{\mathcal{O}_1}$ and $\widetilde{\mathcal{O}_2}$, respectively. The signature of an axiom α is denoted $\widetilde{\alpha}$. A structural equivalence relation between two axioms α_1 and α_2 is denoted $\alpha_1 \equiv_s \alpha_2$. Throughout this paper we use the standard description and first order logic notion of entailment; an axiom α entailed by an ontology $\mathcal{O}$ is denoted $\mathcal{O} \models \alpha$. A *justification* $\mathcal{J}$ for an entailment α is a $\subseteq$-minimal subset of an ontology $\mathcal{O}$ that is sufficient for α to hold [8]. We refer to an *effectual* addition (removal) from $\mathcal{O}_1$ to $\mathcal{O}_2$ as an axiom α such that $\alpha \in \mathcal{O}_2$ and $\mathcal{O}_1 \not\models \alpha$ ($\alpha \in \mathcal{O}_1$ and $\mathcal{O}_2 \not\models \alpha$) [4].

3 Ecco: A Hybrid Diff for OWL 2

We present the diff tool Ecco, available on the web as a Java servlet, and accessible at `http://owl.cs.manchester.ac.uk/diff`. Alternatively there is also a command-line tool with more advanced features, which can be downloaded from `http://owl.cs.manchester.ac.uk/research/topics/diff/`. In order to demonstrate the functionality of the tool, as well as how its output can be interpreted, we start off with a comprehensive diff walkthrough on toy ontologies, and further on we show the output of Ecco on those same ontologies.

3.1 Diff Walkthrough

Consider ontologies $\mathcal{O}_1$ and $\mathcal{O}_2$, defined in Table 1. 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

Table 1: Example ontologies.

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

as changes. Given the sets of structural additions and removals from $\mathcal{O}_1$ to $\mathcal{O}_2$, we check which axioms in Removals$(\mathcal{O}_1, \mathcal{O}_2)$ are entailed by $\mathcal{O}_2$ (ineffectual removals), and vice-versa for Additions$(\mathcal{O}_1, \mathcal{O}_2)$. Thus we obtain a distinction between effectual and ineffectual changes, as follows:

- EffectualAdditions$(\mathcal{O}_1, \mathcal{O}_2) = \{\beta_2, \beta_6, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}\}$
- EffectualRemovals$(\mathcal{O}_1, \mathcal{O}_2) = \{\alpha_8\}$
- IneffectualAdditions$(\mathcal{O}_1, \mathcal{O}_2) = \{\beta_1, \beta_4, \beta_5, \beta_7, \beta_9\}$
- 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. Consider these 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 $\mathcal{O}_2$, and $\mathcal{O}_2 \models A \sqsubseteq B$. 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 $\beta_2 \in$ 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; $\beta_1 \in$ IneffectualAdditions$(\mathcal{O}_1, \mathcal{O}_2)$, and $\beta_3 \in (\mathcal{O}_1 \cap \mathcal{O}_2)$.

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 represent 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}.

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

	Removals	Axioms	Source
Ineffectual	Rewritten	α_3	$\{\beta_4, \beta_5\}$
Ineffectual	Strengthened	α_1	$\{\beta_2, \beta_3\}$
		α_4	$\{\beta_{11}\}, \{\beta_{12}\}$
Ineffectual	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	Source
Ineffectual	Rewritten	β_9	$\{\alpha_5, \alpha_7\}$
Ineffectual	Weakened	β_9	$\{\alpha_8\}$
		β_7	$\{\alpha_3\}$, $\{\alpha_4, \alpha_5\}$
Ineffectual	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	Strengthening	β_{11}	of α_4 ($\beta_{11} \models \alpha_4$)
Effectual	StrengtheningNT	β_{12}	of α_4 ($\beta_{12} \models \alpha_4$) and p is new in $\widetilde{\mathcal{O}}_1$
Effectual	NewDescription	β_{10}	K is defined in β_{10} and new in $\mathcal{O}_2$
Effectual	PureAddition	β_2	is a new axiom about shared terms (in $\widetilde{\mathcal{O}}_1 \cap \widetilde{\mathcal{O}}_2$)
Effectual	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 $\widetilde{\mathcal{O}}_2$

Generally speaking, with such a categorisation it becomes conceivably more intuitive to navigate and understand change sets. Even though ineffectual changes are often ignored by semantic diffs (ContentCVS [7], or CEX [10]), we gathered from their analysis useful information such as, e.g., that axiom α_4 is strengthened in two distinct, yet partially superfluous axioms (β_{11} and β_{12}). Similarly we discovered 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).

3.2 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 [8], and module extraction algorithms [2]. Ecco is implemented in Java, based on the OWL API [5] (v3.2.4).[1] The output of a diff is an XML file,[2] containing the axioms in the diff (in Manchester syntax)[3] and their respective categories. In order to present this output in a more sensible way, and allow user interaction with the output, we use XSLT[4] to transform the XML file into HTML, which, together with the supplied JavaScript[5] file, produces a hands-on front-end to the categorised change set.

The entry point to the diff on the web is shown in Figure 1, wherein users can supply URL's for the ontologies, or browse for ontology files in the local system. Using the command-line version of Ecco, users have further optional arguments:

-r Analyze the root ontologies only, not any of their imports
-s Save the categorical sets as OWL ontologies
-o Output the change sets to the specified directory
-n Normalize entity URIs. I.e. if two ontologies have the same entity names in a different namespace, this flag establishes a common namespace
-i Ignore Abox axioms
-v -version Print version information and exit
-h -help Print help message

The output of both implementations is the same; the XML change set with its transformation into HTML. The resulting webpage allows users to browse through the change set by focusing on general categories (e.g., additions or effectual changes only) or more specific ones (e.g., weakenings). The webpage derived from the diff example in Section 3.1 is shown in Figure 2. The basic layout displays removals on the left-hand side, in red, and additions on the right-hand side, in green. The top level links and buttons allow users to, accordingly, get the source XML file, show or hide all changes, and adjust entity rendering according to entity names, entity labels (rdfs:label), or gensyms. The latter can be used to mask entity names by replacing them with shorter symbols, which, in cases where entity names or labels are too big, reduces the amount of on-screen information, making pattern analysis easier. In the change summary we present a hierarchical structure of the categories, and the number of changes in each of them. Additionally there are help buttons to help the user understand what each category represents.

From this point onwards, users can select what kind of changes to focus on, having triggers available to inspect, and navigate to specific categories; e.g., in Figure 3 we have focused on strengthened axioms. In this figure we see on the left-hand side the removed axioms, while on the right-hand side, this being an ineffectual change, we present the justifications for each change (note that there

[1] `http://owlapi.sourceforge.net/`
[2] `http://www.w3.org/XML/`
[3] `http://www.w3.org/TR/owl2-manchester-syntax/`
[4] `http://www.w3.org/TR/xslt`
[5] `https://developer.mozilla.org/en/JavaScript`

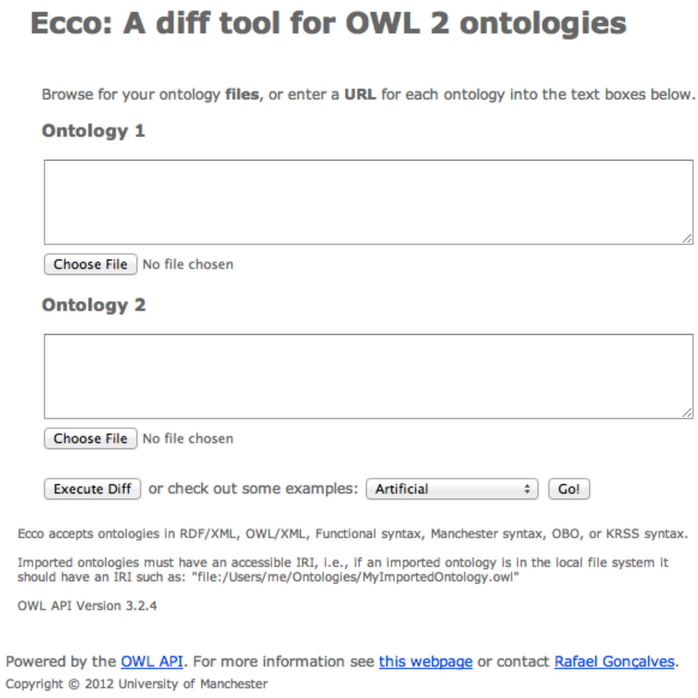

Fig. 1: Entry point to the diff tool on the web.

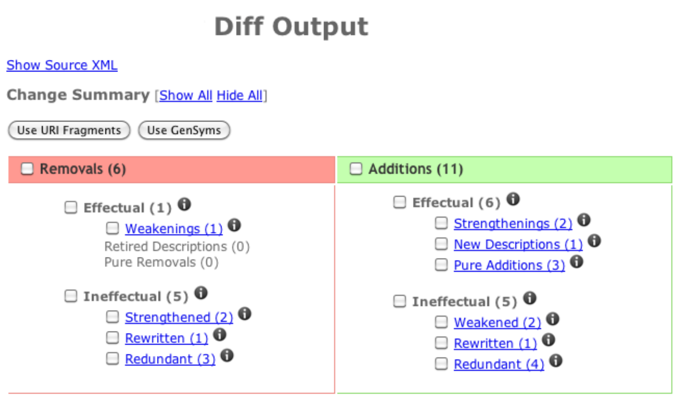

Fig. 2: Output of the diff between $\mathcal{O}_1$ and $\mathcal{O}_2$.

can be more than one justification, such as change with ID 6). Furthermore the tool flags those axioms that are shared between both ontologies.

Consider Figure 4, where we have focused on effectual additions, specifically strengthenings and new descriptions. In categories involving new or retired terms, such as new or retired descriptions, these are appropriately revealed below the axiom, as shown in Figure 4. In the case of strengthenings there is a one-to-one alignment between target and source axioms, i.e., the change and what it is a change of.

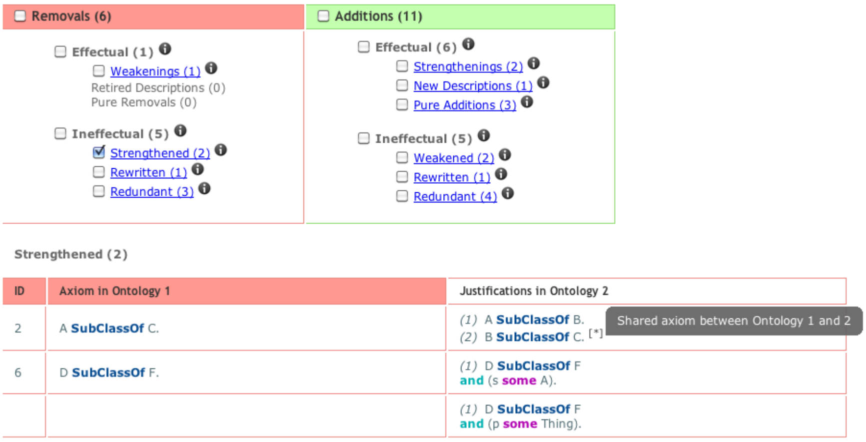

Fig. 3: Strengthened axioms between $\mathcal{O}_1$ and $\mathcal{O}_2$.

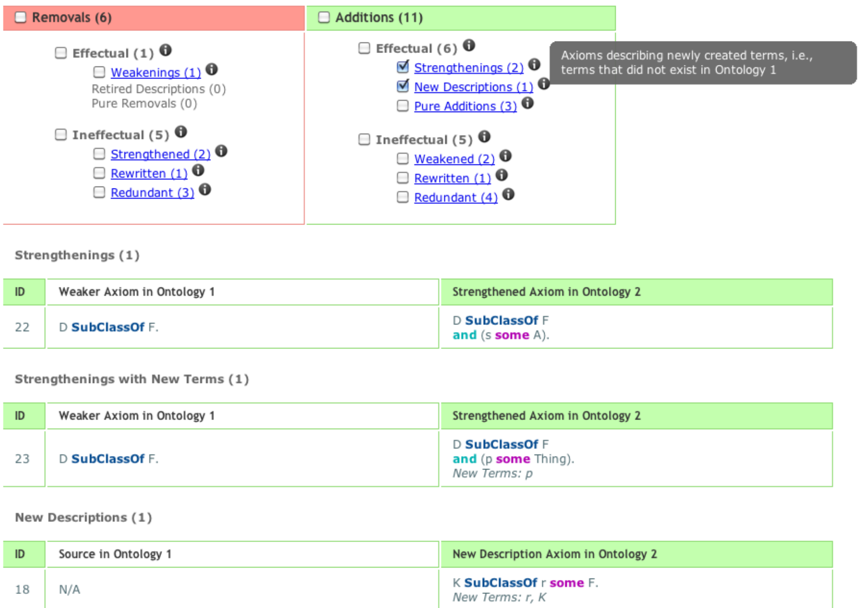

Fig. 4: Select categories of effectual additions between $\mathcal{O}_1$ and $\mathcal{O}_2$.

4 Discussion

We have presented the diff tool Ecco, and shown how its categorisation and presentation of change sets facilitates change analysis. 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 an unstructured change set while inspecting both ontologies. Moreover, with the alignment of changes between ontologies we can show the changed axioms and what they are a change of. Consequently, by analysing changes in this way, one can conceivably avoid examining the actual ontologies. Ecco is available as a web-based application, allowing users to compare ontologies with no installation necessary. When specific requirements come into play (e.g., increased Java heap space, or tool-specific options), the standalone version of Ecco would be most appropriate, enabling more advanced features.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2003)
2. Cuenca Grau, B., Horrocks, I., Kazakov, Y., Sattler, U.: Modular reuse of ontologies: Theory and practice. J. of Artificial Intelligence Research 31 (2008)
3. 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)
4. Gonçalves, R.S., Parsia, B., Sattler, U.: Categorising logical differences between OWL ontologies. In: Proc. of CIKM-11 (2011)
5. Horridge, M., Bechhofer, S.: The OWL API: A Java API for working with OWL 2 ontologies. In: Proc. of OWLED-09 (2009)
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) (2011)
8. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Proc. of ISWC/ASWC-07 (2007)
9. Kalyanpur, A., Parsia, B., Sirin, E., Cuenca Grau, B., Hendler, J.: Swoop: A Web ontology editing browser. J. of Web Semantics 4(2) (2006)
10. Konev, B., Walther, D., Wolter, F.: The logical difference problem for description logic terminologies. In: IJCAR-08. vol. 5195 (2008)
11. Křemen, P., Šmíd, M., Kouba, Z.: OWLDiff: A practical tool for comparison and merge of OWL ontologies. In: Proc. of DEXA-12 (2011)
12. 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)
13. Motik, B., Patel-Schneider, P., Parsia, B.: OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. W3C Cand. Rec. (11 June 2009), http://www.w3.org/TR/2009/CR-owl2-syntax-20090611/