Ontology Adaptation upon Updates

Alessandro Solimando, Giovanna Guerrini

Dipartimento di Informatica, Bioingegneria, Robotica e Ingegneria dei Sistemi
Università di Genova, Italy
name.surname@unige.it

Abstract. Ontologies, like any other model, change over time due to modifications of the modeled domain, a deeper understanding of the domain by the modeler, error corrections, simple refactoring or shift of modeling granularity level. Local changes usually impact the remainder of the ontology as well as any other data and metadata defined over it. The massive size of ontologies and their possible fast update rate (consider, *e.g.*, the daily updates of Gene Ontology, with $\sim 416K$ axioms and $\sim 40K$ entities) requires automatic adaptation methods for relieving ontology engineers from a manual intervention, in order to allow them to focus mainly on high-level inspection. This paper, in spirit of the *Principle of minimal change*, proposes a fully automatic ontology adaptation approach that reacts to ontology updates and computes sound reformulations of ontological axioms triggered by the presence of certain preconditions. The rule-based adaptation algorithm covers up to $\mathcal{SROIQ}$ DL.

1 Introduction and Motivations

Ontologies, like any other model, change over time and a revalidation of all data and metadata defined on top of the modified ontology is needed upon updates. The massive ontology size and possibly fast update rate[1] calls for automated support and adaptation algorithms. Despite the great attention devoted in the last ten years to ontology evolution (refer to [1] for a survey on the topic, and to [7] for an analysis of the difference between ontology evolution and schema evolution), to the best of our knowledge there are no proposals in the literature coping with ontology adaptations upon ontology updates. With similar motivations, an adaptation algorithm for a subset of *SPARQL* queries (with expressivity equivalent to Union of Conjunctive Queries) in response to ontology updates is proposed in [6]. *Protégé*[2], one of the most complete ontology frameworks, does not support any kind of adaptation w.r.t. ontology updates: when a concept or a role is deleted, all the axioms referring it are removed as well. Even if there are cases in which this behavior could be acceptable (*e.g.*, error corrections), there are others for which it is detrimental, for instance a modification of the

[1] An example is the *Gene Ontology* (http://www.geneontology.org/), with $\sim 416K$ axioms and $\sim 40K$ entities, daily updated (statistics for data-version 2013-02-22).
[2] Available here: http://protege.stanford.edu/

modeling granularity of the ontology. In this scenario, a valid reformulation of axioms by means of super/sub concepts or roles is not only desirable but usually manually performed by the modeler. Additionally, in Artificial Intelligence (*Belief Revision*), knowledge deletion usually follows the *Principle of Minimal Change* [5], which suggests that the amount of lost information should be as minimal as possible. Given that ontologies do not necessarily (explicitly) include all their logical consequences, also the implicit knowledge should be taken into account, as well as explicit one (that is, ontology axioms).

In this paper, we propose an algorithm that, given an ontology and an entity (concept or role) to delete, scans for an equivalent, a super and a sub-entity and tries to reformulate the axioms involving the entity in question, with a rule-based approach. Our reformulated axioms are a fraction of the *implicit knowledge* of the ontology under update that would be lost by deleting all of the axioms involving the removed entity. An alternative would be to compute the closure (that is, complete inference of implicit knowledge) for the ontology prior to entity deletion. Due to its high computational cost and possible non-finiteness of the result, a suboptimal but less expensive approach is preferable for our target scenario, that is interactive modeling.

While a set of basic ontology changes can be easily defined, it is impossible to identify a set of complex changes without fixing the granularity level, *i.e.*, updates expressed as arbitrarily complex graph patterns (see [10], Section 3.2.1). In this proposal we consider the basic updates proposed by [3]: addition, deletion and update of entities (concepts and roles). Given that adding or updating entities do not reduce knowledge, and that ontology consistency can be tested using ontology reasoners, our adaptation algorithm focuses only on entity deletions.

Even if the adaptation algorithm is completely automatic, it may not always be aligned with the modeler's intention. For this reason the present proposal has to be intended as an optional feature. When activated, a preview of the changes has to show the effects of the automatic adaptation, that could be selectively accepted or ignored by the modeler. In addition, a straightforward extension could be the possibility, for the modeler, to select the equivalent (resp. sub/super) entity for the reformulation, when different alternatives are available.

The contribution of the present paper can be summarized as follows: an automatic adaptation algorithm supporting up to $\mathcal{SROIQ}$ DL expressivity, correctness proof, and temporal complexity (Section 3), an experimental evaluation of the percentage of adaptable entities and axioms on a dataset of real ontologies (Section 4). First, DL basics are introduced (Section 2), and the paper concludes discussing future work (Section 5).

2 Preliminaries

Our proposal covers up to $\mathcal{SROIQ}$ Description Logic (DL), on top of which the *Ontology Web Language* (OWL2) [11] is defined. The notations and definitions used in this section are borrowed from [4]. An ontology is defined by a set of axioms and a set of entity names (signature), composed by three disjoint subsets:

	Precondition	Rule
a.1	$C \equiv C'$, $C \in signature(axiom)$	$axiom \to axiom[C/C']$
a.2	$C \sqsubseteq D$	$E \equiv \exists R.C \to E \sqsubseteq \exists R.D$
a.3	$C \sqsubseteq D$	$E \equiv\geq_n R.C \to E \sqsubseteq\geq_n R.D$
a.4	$C \sqsubseteq D$	$E \equiv C \sqcup F \to E \sqsubseteq D \sqcup F$
a.5	$C \sqsubseteq D$	$E \equiv C \sqcap F \to E \sqsubseteq D \sqcap F$
a.6	$C \sqsubseteq D$	$E \equiv \neg C \to \neg D \sqsubseteq E$
a.7	$C \sqsubseteq D$	$C(a) \to D(a)$
a.8	$C \sqsubseteq D$	$E \equiv \forall R.C \to E \sqsubseteq \forall R.D$
a.9	$B \sqsubseteq C$	$E \equiv\leq_n R.C \to E \sqsubseteq\leq_n R.B$
a.10	$B \sqsubseteq C$	$E \equiv C \sqcup F \to B \sqcup F \sqsubseteq E$
a.11	$B \sqsubseteq C$	$E \equiv C \sqcap F \to B \sqcap F \sqsubseteq E$
a.12	$B \sqsubseteq C$	$E \equiv \neg C \to E \sqsubseteq \neg B$
a.13	$B \sqsubseteq C$	$C \sqsubseteq E \to B \sqsubseteq E$

Table 1. Adaptation rules for concept deletion $DEL(C)$, where $B, C, C', D, E, F \in N_{\mathcal{C}}$, $R \in N_{\mathcal{R}}$ and $a \in N_{\mathcal{I}}$.

$N_{\mathcal{R}}$ for role names, $N_{\mathcal{I}}$ for individual names, $N_{\mathcal{C}}$ for concept name. These entities are defined by means of expressions. We have *Role expressions* $\boldsymbol{R} ::= U \mid N_{\mathcal{R}} \mid N_{\mathcal{R}}^{-}$, and *Concept expressions* $\boldsymbol{C} ::= N_{\mathcal{C}} \mid (\boldsymbol{C} \sqcup \boldsymbol{C}) \mid (\boldsymbol{C} \sqcap \boldsymbol{C}) \mid \neg\boldsymbol{C} \mid \top \mid \bot \mid \exists\boldsymbol{R}.\boldsymbol{C} \mid \forall\boldsymbol{R}.\boldsymbol{C} \mid \geq_n \boldsymbol{R}.\boldsymbol{C} \mid \leq_n \boldsymbol{R}.\boldsymbol{C} \mid \exists\boldsymbol{R}.Self \mid \{N_{\mathcal{I}}\}$, with $n \geq 0$. For the semantics associated with nominals, roles, and concept expressions the reader may refer to [4]. The set of axioms of an ontology, denoted with $Axioms$, is defined as $Axiom ::= ABox \cup RBox \cup TBox$. The reader may refer to [4] also for a detailed description of the different available axioms for $\mathcal{SROIQ}$ DL, and to [9] for the definitions of ontology *interpretation* and ontology *satisfiability*. W.l.o.g. we will consider normalized ontologies in *Negation Normal Form* (NNF), with an application of *Structural Reduction* (SR), as shown in [9] (Subsection 5.3). SR introduces fresh concept names for (complex) concept expressions, thus letting us to easily refer to each concept expression by means of its associated concept name. Neither the SR nor the NNF are required for the application of our method. NNF, however, may increase the ratio of adapted axioms.

3 Algorithm

This section introduces the adaptation rules (Section 3.1), the rule-based adaptation algorithm (Section 3.2), the correctness proof for the given rules (Section 3.3), and temporal complexity of the algorithm (Section 3.4).

3.1 Adaptation Rules

The adaptation rules are presented in Table 1 (rules for concepts) and Table 2 (rules for roles). We denote by $axiom[A/B]$ the alpha renaming of an axiom of entity A by entity B. A rule r is composed by a left hand side, $LHS(r)$, a right

	Precondition	Rule
b.1	$R \equiv R'$, $R \in signature(axiom)$	$axiom \rightarrow axiom[R/R']$
b.2	$Q \sqsubseteq R$	$T_0 \circ \ldots \circ T_o \circ R \circ T'_0 \circ \ldots \circ T'_p \sqsubseteq T$ $\rightarrow T_0 \circ \ldots \circ T_o \circ Q \circ T'_0 \circ \ldots \circ T'_p \sqsubseteq T$
b.3	$Q \sqsubseteq R$	$E \equiv \forall R.C \rightarrow E \sqsubseteq \forall Q.C$
b.4	$Q \sqsubseteq R$	$E \equiv \leq_n R.C \rightarrow E \sqsubseteq \leq_n Q.C$
b.5	$Q \sqsubseteq R$	$T \equiv R^- \rightarrow Q^- \sqsubseteq T$
b.6	$Q \sqsubseteq R$	$Disjoint(R, T) \rightarrow Disjoint(Q, T)$
b.7	$R \sqsubseteq S$	$R(a, b) \rightarrow S(a, b)$
b.8	$R \sqsubseteq S$	$E \equiv \exists R.C \rightarrow E \sqsubseteq \exists S.C$
b.9	$R \sqsubseteq S$	$E \equiv \exists R.Self \rightarrow E \sqsubseteq \exists S.Self$
b.10	$R \sqsubseteq S$	$E \equiv \geq_n R.C \rightarrow E \sqsubseteq \geq_n S.C$
b.11	$R \sqsubseteq S$	$T \equiv R^- \rightarrow T \sqsubseteq S^-$
b.12	$R \sqsubseteq S$	$T_0 \circ \ldots \circ T_q \sqsubseteq R \rightarrow T_0 \circ \ldots \circ T_q \sqsubseteq S$

Table 2. Adaptation rules for role deletion $DEL(R)$, where E, $C \in N_{\mathcal{C}}$, Q, R, R', S, T, T_i, $T'_j \in N_{\mathcal{R}}$, with $n, o, p, q \geq 0$, and $a, b \in N_{\mathcal{I}}$.

hand side, $RHS(r)$, and a precondition $precond(r)$. A rule is defined **applicable** iff $precond(r)$ is satisfied by at least one concept (resp. role). Given an ontology o and an entity e to delete, the LHS of a rule r is said to be in **matching** iff it exists an axiom in o that is equal to $LHS(r)$ modulo alpha renaming of C (resp. R) with e, denoted with $LHS(r)[e]$. The **application** of an applicable rule r w.r.t. o and e rewrites any axiom of o matching $LHS(r)[e]$ into $RHS(r)[e']$, where e' is the selected entity for reformulation. It is worth noting that if a DL less expressive than $\mathcal{SROIQ}$ is adapted, only a subset of the rules will be applicable, depending on the axioms and constructors available. For instance, for basic $\mathcal{ALC}$ DL with *General Concept Inclusion* (*i.e.*, $C \sqsubseteq D$), rules a.3, a.9, b.2, b.4, b.5, b.9, b.10, b.11, b.12 are not applicable.

3.2 Adaptation Algorithm

Algorithm 1 presents the adaptation algorithm for ontology updates. It takes as input the entity e to be deleted and the ontology o it belongs to. By means of function *computePrecond*, the set of axioms related to e is computed, as well as a triple p consisting of an equivalent, a sub and a super entity, if any (line 3). For each axiom a having e in its signature (line 4), it tests if the axiom matches the left hand side of the rule (line 5). At this point the function $satisfies$ (line 6) checks if the current axiom is compatible with rule r and if the required element of p is not null. The reformulated axiom is inserted in o (line 7). Finally, all the axioms involving entity e are removed from o (line 8). Even if a preliminar classification phase is not required, it may increase the algorhtm effectivity. In what follows we give a toy example of ontology update, comparing the result of adaptation to classical deletion approach:

Algorithm 1 Ontology Update Adaptation

```
function OntoUpdateAdapt(Entity e, Ontology o)
    axioms = ∅
    p := ⟨eq, sub, sup⟩ ← computePrecond(e, axioms, o)
    for a ∈ axioms do
        for r ∈ Rules . a = LHS(r)[e] do
            if satisfies(⟨a, e, e′⟩, precond(r)), e′ ∈ {eq, sub, sup} then
                Axioms(o) ← Axioms(o) ∪ {RHS(r)[e′]}
            end if
        end for
    end for
    Axioms(o) ← Axioms(o) \ axioms
end function
function computePrecond(Entity e, Set axioms, Ontology o)
    eq, sub, sup ← ϵ
    for a ∈ Axioms(o) . e ∈ signature(a) do
        axioms ← axioms ∪ {a}
        if eq, sub, sup ≠ ϵ then
            break
        end if
        if a = e ≡ e′ or a = e′ ≡ e then
            eq ← e′
        else if a = e ⊑ e′ then
            sup ← e′
        else if a = e ⊒ e′ then
            sub ← e′
        end if
    end for
    return ⟨eq, sub, sup⟩
end function
```

Example 1. Consider an ontology o consisting of these axioms and the natural associated signature: $Human \equiv \exists eats.Food$, $Food(cheese)$, $Eater \equiv \forall eats.Food$, $\bot \equiv Plastic \sqcap Food$, $Uneatable \equiv \neg Eatable$, $Pizza \sqsubseteq Food$, $Food \sqsubseteq Eatable$. Deleting $Food$ concept from o with adaptation we obtain: $Human \sqsubseteq \exists eats.Eatable$, $Eatable(cheese)$, $Eater \sqsubseteq \forall eats.Eatable$, $Plastic \sqcap Pizza \sqsubseteq \bot$, $Uneatable \equiv \neg Eatable$, $Pizza \sqsubseteq Eatable$ (using rule a.2, a.7, a.8, a.11 and a.13, respectively). Without adaptation, instead, only the last two axioms would be present in o after concept deletion.

3.3 Rules Correctness Proof

Before giving the proposition about the correctness of the adaptation rules we introduce some definitions and lemmata. For sake of brevity we will interchangeably refer to the axioms and their semantics, according to [4].

Definition 1. *An axiom A_1 **entails** an axiom A_2 iff, for any interpretation I, $I \models A_2 \implies I \models A_1$, that is ${A_2}^I \subseteq {A_1}^I$.*

Definition 2. *An adaptation rule r is **sound** iff $\{LHS(r), precond(r)\}$ entails $RHS(r)$.*

Lemma 1. $\forall C, D, F \in N_{\mathcal{C}} \; . \; C \sqsubseteq D \implies C \sqcup F \sqsubseteq D \sqcup F$

Proof. The semantics associated with Lemma 1 is $C^I \subseteq D^I \implies \underbrace{C^I \cup F^I}_{\alpha} \subseteq \underbrace{D^I \cup F^I}_{\beta}$. We assume that the preceding formula does not hold, $\alpha \nsubseteq \beta \implies \exists x \in \beta \,.\, x \notin \alpha$, this requires that at least one of the following conditions hold:

- $x \in F^I$, but this implies $x \in \alpha$, resulting in a contradiction,
- $x \in C^I$, but $C^I \subseteq D^I \implies x \in \alpha$, contradicting the hypothesis. □

Lemma 2. $\forall C, D, F \in N_{\mathcal{C}} \,.\, C \sqsubseteq D \implies C \sqcap F \sqsubseteq D \sqcap F$

Proof. The semantics associated with Lemma 2 is $C^I \subseteq D^I \implies \underbrace{C^I \cap F^I}_{\alpha} \subseteq \underbrace{D^I \cap F^I}_{\beta}$. Assume that $\alpha \nsubseteq \beta$. This requires that $\exists x \in \beta \,.\, x \notin \alpha$. $x \in \beta$ or, equivalently, that $x \in F^I \wedge x \in D^I$ holds. However, $x \in F^I \implies x \in \alpha$, and $x \in D^I$ contradicts the premise $C \sqsubseteq D$. □

Lemma 3. $\forall C, D \in N_{\mathcal{C}} \,.\, C \sqsubseteq D \implies \exists R.C \sqsubseteq \exists R.D$.

Proof. Assume that $\{x \mid \exists y \in C^I \,.\, \langle x, y\rangle \in R^I\} \nsubseteq \{x \mid \exists y \in D^I \,.\, \langle x, y\rangle \in R^I\}$ holds, that is, $\exists R.C \not\sqsubseteq \exists R.D$. This requires that the following condition holds: $\exists \langle x, y\rangle \in R^I \,.\, y \in C^I \wedge y \notin D^I$. But, if such condition holds, then $C \not\sqsubseteq D$, contradicting the premise. □

Lemma 4. $\forall C, D \in N_{\mathcal{C}} \,.\, C \sqsubseteq D \implies \forall R.C \sqsubseteq \forall R.D$.

Proof. Assume that $\{x \mid \forall \langle x, y\rangle \in R^I \implies y \in C^I\} \nsubseteq \{x \mid \forall \langle x, y\rangle \in R^I \implies y \in D^I\}$ holds, that is, $\forall R.C \not\sqsubseteq \forall R.D$. This requires that the following condition holds: $(\exists x \,.\, \forall \langle x, y\rangle \in R^I \implies y \in C^I) \wedge (\exists \bar{y} \,.\, \langle x, \bar{y}\rangle \in R^I \wedge \bar{y} \notin D^I)$. But, if this condition holds, then an $\bar{y}$ exists and R^I is not empty. Therefore, since the left operand of the implication holds, then right operand also does. From this, we obtain $C^I \nsubseteq D^I$, contradicting the premise. □

Proposition 1. *Adaptation rules application preserves ontology satisfiability.*

Proof. Ontology satisfiability is preserved because every adaptation rule is sound. We prove this for each rule separately:

a.1 The proof directly follows from the definition of *Concept Equivalence* axiom.

a.2 $E \equiv \exists R.C \rightarrow E \sqsubseteq \exists R.D$: $\exists R.C \sqsubseteq \exists R.D$ must hold: as a precondition for the application of the rule we have $C \sqsubseteq D$, so we can apply Lemma 3.

a.3 $E \equiv \geq_n R.C \rightarrow E \sqsubseteq \geq_n R.D$: $\geq_n R.C \sqsubseteq \geq_n R.D$ must hold, but it is sufficient that $\{x \mid \exists y \in C^I \,.\, \langle x, y\rangle \in R^I\} \subseteq \{x \mid \exists y \in D^I \,.\, \langle x, y\rangle \in R^I\}$ holds. As a precondition for the application of the rule we have $C \sqsubseteq D$, so we can apply Lemma 3.

a.4 $E \equiv C \sqcup F \rightarrow E \sqsubseteq D \sqcup F$: $C \sqcup F \sqsubseteq D \sqcup F$ holds for Lemma 1 because the rule precondition $C \sqsubseteq D$ holds.

a.5 $E \equiv C \sqcap F \to E \sqsubseteq D \sqcap F$: $C \sqcap F \sqsubseteq D \sqcap F$ holds for Lemma 2 because the rule precondition $C \sqsubseteq D$ holds.

a.6 $E \equiv \neg C \to \neg D \sqsubseteq E$: $\neg D \sqsubseteq \neg C$ must hold: the semantics is $\Delta^I \setminus D^I \subseteq \Delta^I \setminus C^I$, but this contradicts $C \sqsubseteq D$, the precondition for rule application.

a.7 $C(a) \to D(a)$: $C(a) \implies D(a)$, guaranteed by rule precondition.

a.8 $E \equiv \forall R.C \to E \sqsubseteq \forall R.D$: $E \equiv \forall R.C \implies E \sqsubseteq \forall R.D$ holds for Lemma 4 because the rule precondition $C \sqsubseteq D$ holds.

a.9 $E \equiv \leq_n R.C \to E \sqsubseteq \leq_n R.B$: $\leq_n R.B \sqsubseteq \leq_n R.C$, but it is sufficient that $\{x \mid \exists y \in B^I \,.\, \langle x, y\rangle \in R^I\} \subseteq \{x \mid \exists y \in C^I \,.\, \langle x, y\rangle \in R^I\}$. As a precondition for rule application we have $B \sqsubseteq C$, so we can apply Lemma 3.

a.10 $E \equiv C \sqcup F \to B \sqcup F \sqsubseteq E$: $E \equiv C \sqcup F \implies B \sqcup F \sqsubseteq E$ must hold. The semantics is $E^I = C^I \cup F^I \supseteq B^I \cup F^I$. As a precondition for rule application we have $B \sqsubseteq C$, so we can apply Lemma 1 for proving that $B^I \cup F^I \subseteq C^I \cup F^I$.

a.11 $E \equiv C \sqcap F \to B \sqcap F \sqsubseteq E$: $E \equiv C \sqcap F \implies B \sqcap F \sqsubseteq E$ must hold. The semantics is $E^I = C^I \cap F^I \supseteq B^I \cap F^I$. As a precondition for rule application we have $B \sqsubseteq C$, so we can apply Lemma 2 for proving that $B^I \cap F^I \subseteq C^I \cap F^I$.

a.12 $E \equiv \neg C \to E \sqsubseteq \neg B$: the proof showing that $\neg C \sqsubseteq \neg B$ holds is the dual of the one given in item (a.6).

a.13 $C \sqsubseteq E \to B \sqsubseteq E$: as a precondition for rule application we have $B \sqsubseteq C$, that by transitivity implies that $B \sqsubseteq E$.

b.1 The proof directly follows from the definition of *Role Equivalence* axiom.

b.2 $\underbrace{T_0 \circ \ldots \circ T_o \circ R \circ T'_0 \circ \ldots \circ T'_p}_{\alpha} \sqsubseteq T \to \underbrace{T_0 \circ \ldots \circ T_o \circ Q \circ T'_0 \circ \ldots \circ T'_p}_{\beta} \sqsubseteq T$: assume that $\beta^I \not\subseteq \alpha^I$ holds. This requires that $\exists x_0, \ldots, x_{o+p+3} \,.\, \langle x_0, x_1\rangle \in {T_0}^I \wedge \ldots \wedge \langle x_{o+1}, x_{o+2}\rangle \in Q^I \wedge \langle x_{o+p+2}, w_{o+p+3}\rangle \in {T'_p}^I \wedge \langle x_{o+1}, x_{o+2}\rangle \notin R^I$, but this contradicts the rule precondition $Q \sqsubseteq R$.

b.3 $E \equiv \underbrace{\forall R.C}_{\alpha} \to E \sqsubseteq \underbrace{\forall Q.C}_{\beta}$: assume that $\alpha^I \not\subseteq \beta^I$ holds. This requires that $\exists x \,.\, x \in \alpha^I \wedge x \notin \beta^I$, that is equals to $\exists x.((\forall y \,.\, \langle x, y\rangle \in R^I \implies y \in C^I) \wedge (\exists y' \,.\, \langle x, y'\rangle \in Q^I \wedge y' \notin C^I))$. Given that $Q \sqsubseteq R$, if such y' exists, α cannot hold, leading to a contradiction.

b.4 $T \equiv \underbrace{\leq_n R.C}_{\alpha} \to T \sqsubseteq \underbrace{\leq_n Q.C}_{\beta}$: assume that $\alpha^I \not\subseteq \beta^I$. This requires that $\exists x \,.\, |\{y \mid y \in C^I \wedge \langle x, y\rangle \in R^I\}| \leq n \wedge |\{y \mid y \in C^I \wedge \langle x, y\rangle \in Q^I\}| > n$. This implies $|Q^I| > |R^I|$ contradicting the rule precondition $Q \sqsubseteq R$.

b.5 $T \equiv R^- \to Q^- \sqsubseteq T$: assume that ${Q^-}^I \not\subseteq {R^-}^I$. This requires that $\exists \langle x, y\rangle \,.\, \langle y, x\rangle \in Q^I \wedge \langle y, x\rangle \notin R^I$, but this contradicts the rule precondition $Q^I \subseteq R^I$.

b.6 $Disjoint(R, T) \to Disjoint(Q, T)$: assume that $R^I \cap T^I = \emptyset \implies Q^I \cap T^I = \emptyset$ does not hold. This requires that $\exists \langle x, y\rangle \in Q^I \wedge \langle x, y\rangle \in T^I \wedge \langle x, y\rangle \notin R^I$ holds, but $\langle x, y\rangle \in Q^I \wedge \langle x, y\rangle \notin R^I$ contradicts $Q \sqsubseteq R$.

b.7 $R(a,b) \rightarrow S(a,b)$: from the rule precondition $R \sqsubseteq S$ we have that $\forall \langle x,y \rangle . \langle x,y \rangle \in R \implies \langle x,y \rangle \in S$.

b.8 $E \equiv \underbrace{\exists R.C}_{\alpha} \rightarrow E \sqsubseteq \underbrace{\exists S.C}_{\beta}$: assume that $\alpha^I \not\subseteq \beta^I$. This requires that $\exists x . \exists y \in C^I . \langle x,y \rangle \in S^I \wedge \langle x,y \rangle \notin R^I$ holds, but this contradicts the rule precondition $R \sqsubseteq S$.

b.9 $E \equiv \underbrace{\exists R.Self}_{\alpha} \rightarrow E \sqsubseteq \underbrace{\exists S.Self}_{\beta}$: assume that $\alpha \not\subseteq \beta$. This requires that $\exists x . \langle x,x \rangle \in S^I \wedge \langle x,x \rangle \notin R^I$ holds, but this contradicts $R \sqsubseteq S$.

b.10 $E \equiv \underbrace{\geq_n R.C}_{\alpha} \rightarrow E \sqsubseteq \underbrace{\geq_n S.C}_{\beta}$: assume that $\alpha^I \not\subseteq \beta^I$. This requires that $\exists x . |\{y \mid y \in C^I \wedge \langle x,y \rangle \in R^I\}| \geq n \wedge |\{y \mid y \in C^I \wedge \langle x,y \rangle \in S^I\}| < n$. This implies $|R^I| > |S^I|$ contradicting the rule precondition $R \sqsubseteq S$.

b.11 $T \equiv R^- \rightarrow T \sqsubseteq S^-$: assume that ${R^-}^I \not\subseteq {S^-}^I$. This requires that $\exists \langle x,y \rangle . \langle y,x \rangle \in R^I \wedge \langle y,x \rangle \notin S^I$, thus contradicting the rule precondition $R \sqsubseteq S$.

b.12 $T_0 \circ \ldots \circ T_q \sqsubseteq R \rightarrow T_0 \circ \ldots \circ T_q \sqsubseteq S$: this immediately follows, by transitivity, from the rule precondition $R \sqsubseteq S$. $\square$

3.4 Temporal Complexity

In this section we analyze the temporal complexity of the algorithm.

Proposition 2. *The adaptation algorithm that takes as input the ontology o has time complexity in $\mathcal{O}(n)$, where n is equal to the number of axioms of o.*

Proof. *computePrecond* scans all the axioms of ontology o, for each of them it performs some comparison having a total cost of c_1, so it has a cost of $n \cdot c_1$. The *for* statement of line 4 in Algorithm 1 is executed n times in the worst case (each axiom of the ontology refers to the entity in question). The *for* statement of line 5 is executed $c_2 = |Rules|$ times, where *Rules* is the set of adaptation rules. *satisfies* test requires a constant (c_3) time for checking the required conditions. Axiom rewriting and its insertion requires constant (c_4) time. The removal of old axioms requires constant time c_5 too. The overall complexity is therefore equal to $n \cdot c_1 + c_2 \cdot c_3 \cdot c_4 \cdot n + c_5$, that belongs to $\mathcal{O}(n)$. $\square$

4 Experiments

In order to evaluate the practical feasability of our proposal we implemented a Java prototype based on the OWL API library[3]. In addition to correctness we also experimentally evaluated the coverage of OWL2 axioms and constructors of our set of rules. The dataset is presented in Table 3 (manual selection on the Web based on ontology size and DL expressivity).

[3] Available here: `http://owlapi.sourceforge.net/`

Implementation Overview The library does not natively support any modification to the axioms, they are intended as immutable objects. Only the ontology can change, by axiom addition and removal. Whenever possible, the rule application has been simulated with a pair of add and delete changes, otherwise we employed Java Reflection[4] for directly modifying the involved axiom.

Correctness The developed proof-of-concept prototype has been used for testing correctness of our adaptation rules, the experimental counterpart of the proofs given in Section 3.3. More precisely, the test consists in taking as input a satisfiable ontology composed by the precondition and an axiom corresponding to the LHS of a rule r (modulo alpha renaming of the entity to delete). At this point, using *Hermit* reasoner (v1.3.7)[5], we check the entailment of $RHS(r)[e']$.

Evaluation An entity e is **adaptable** iff it satisfies at least one rule precondition, while an axiom a is **adaptable** iff it exists at least one rule r s.t. $LHS(r)[e] = a$. As an estimation of the practical effectiveness of our algorithm, we considered, for each ontology in our dataset, the following scenario: we simulate the deletion of each single entity, in isolation, and we take into account the percentage of adaptable ones (there exists another entity suitable for reformulation). For each of these adaptable entities, we also inspect how many axioms involving them would be adapted instead of simply deleted. For this reference scenario we defined *Coverage* measure as: **(C.1)** the percentage of adaptable concepts (resp. roles **(C.3)**) out of the total number of concepts (resp. roles), and **(C.2)**, the percentage of adaptable axioms w.r.t. the deleted concept (resp. role, **(C.4)**), out of the number of axioms to be deleted (that is, presenting the deleted entity in their signature). In Table 3 the coverage for each ontology in isolation is reported (computed from the raw data of Table 4), while the result considering the dataset as a whole ontology is the following: (C.1) 93.247%, (C.2) 41.757%, (C.2*) 44.185%, (C.3) 73.647%, (C.4) 79.63%, (C.4*) 80.847%. (C._)* substracts from the total number of adaptable axioms the ones not adaptable because the deleted entity do not match their precondition. The result taking these axioms into account represents the percentage of adaptable axioms, while the version excluding them evaluates the completeness of our adaptation rules (the complement of the rewritten axioms is not supported by our rules). Table 3 shows that 10 out of 12 of the worst performing ontologies w.r.t. role coverage ((C.3), (C.4) and (C.4)*) are expressed in a DL missing role hierarchy constructs (identified by letter H in the DL name). Without role hierarchy constructs only role equality can be used for adaptation, thus reducing the number of adaptable roles. Concept coverage (C.1) presents, instead, high values (above 60%) for all the considered ontologies, independently from the DL they are expressed with. This is not surprising because concept hierarchy constructs are available for DLs at least as expressive as $\mathcal{AL}$ DL. On the contrary, coverage results for concept rules w.r.t. OWL2 axioms and constructors seem to be

[4] For a quick summary refer to: `http://docs.oracle.com/javase/tutorial/reflect/index.html`

[5] Hermit and related information are available at `http://hermit-reasoner.com/`

unrelated to either the underpinning DL or the ontology size (in terms of number of axioms and/or entities). For instance, the ontologies with worst values for (C.2)* are 2. ($\mathcal{SHIN}(\mathcal{D})$ DL), 8. ($\mathcal{SHOIN}(\mathcal{D})$ DL) 11. ($\mathcal{SROIF}$ DL) 3. ($\mathcal{ALEHI}+(\mathcal{D})$ DL) and 15. ($\mathcal{ALE}$ DL), with very different number of concepts and axioms (Table 4). Similarly, among the best results for (C.2)* the expressivity ranges from $\mathcal{AL}(\mathcal{D})$ DL to $\mathcal{SROIN}(\mathcal{D})$ DL, again with varying number of axioms and concepts. Ideally the proposal should adapt all the axioms: (C.2)*, in particular, is far from this result, but it is well known that OWL2, despite being based on $\mathcal{SROIQ}$ DL, adds new constructors and axioms, that are derivable from $\mathcal{SROIQ}$ ones (they do not add expressive power). For example, *Concept Disjointness* axiom (*i.e., Disjoint*(C, D), with $C, D \in \mathcal{N}_C$) is only a shortcut for $C \sqcap D \sqsubseteq \bot$[6]. Our prototype strictly applies the rules of Table 1 and Table 2, so it cannot directly process the axioms and constructors not available in $\mathcal{SROIQ}$ DL, thus diminishing the number of adaptable axioms. As a future work we plan to extend the prototype to a full support of OWL2.

5 Future Work

The paper represents, to the best of our knowledge, the first adaptation proposal upon ontology updates. In addition, the algorithm is totally automatic and supports ontology expressivity up to $\mathcal{SROIQ}$ DL, on top of which OWL2 is defined. Both formal and experimental correctness proofs of the adaptation rules have been provided as well as complexity analysis.

The present paper could be extended in several directions. The set of adaptation rules is a preliminary proposal, we plan to further enrich it in order to increase the coverage rate reported in Section 4 and to consider reasonable alternatives for each single rule (*e.g.*, sound alternatives for a.8 could be $C \sqsubseteq D, E \equiv \forall R.C \rightarrow \forall R.D \sqsubseteq T$ or $B_0 \dots B_n \sqsubseteq C, E \equiv \forall R.C \rightarrow E \sqsubseteq \forall R.\bigsqcup_{i=0}^{n} B)$. We also plan to consider the integration of anonymous entities (*e.g.*, using $\top$ as superclass). Another line of research is the integration of a complex update (*e.g.*, concept merge and split) proposals, such as [2]. The relationship between DL updates and *Belief Revision* has been investigated [8], we plan to further investigate it w.r.t. our proposal. Beside theoretical inquiries, we also intend to improve our prototype up to a full support of OWL2. Our final goal will be a *Protégé* plugin, from which we hope to receive feedbacks from the community of ontology engineers and practitioners. The experimental evaluation will also be strengthened with an extended ontology dataset and temporal profiling of the prototype.

References

1. Flouris, G., Manakanatas, D., Kondylakis, H., Plexousakis, D., Antoniou, G.: Ontology change: Classification and survey. Knowl. Eng. Rev. 23(2), 117–152 (Jun 2008)

[6] Refer to [9], Chapter 9, for further examples and details.

2. Hartung, M., Groß, A., Rahm, E.: COnto–Diff: Generation of Complex Evolution Mappings for Life Science Ontologies. Journal of Biomedical Informatics (2012)
3. Hartung, M., Gross, A., Rahm, E.: Rule-based Generation of Diff Evolution Mappings between Ontology Versions. CoRR abs/1010.0122 (2010)
4. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible SROIQ. In: Principles of Knowledge Representation and Reasoning – KR 2006. pp. 57–67 (2006)
5. Katsuno, H., Mendelzon, A.O.: Propositional knowledge base revision and minimal change. Artificial Intelligence 52(3), 263–294 (1991)
6. Kondylakis, H., Plexousakis, D.: Ontology Evolution: Assisting Query Migration. Conceptual Modeling – ER 2012 pp. 331–344 (2012)
7. Noy, N., Klein, M.: Ontology Evolution: Not the same as Schema Evolution. Knowledge and Information Systems 6(4), 428–440 (2004)
8. Ribeiro, M.M., Wassermann, R., Antoniou, G., Flouris, G., Pan, J.: Belief Contraction in Web-Ontology Languages. In: Proceedings of the 3rd International Workshop on Ontology Dynamics, IWOD (2009)
9. Rudolph, S.: Foundations of Description Logics. Reasoning Web. Semantic Technologies for the Web of Data pp. 76–136 (2011)
10. Stojanovic, L.: Methods and Tools for Ontology Evolution. Ph.D. thesis, University of Karlsruhe (2004)
11. W3C as Hitzler, P. and Krötzsch, M. and Parsia, B. Patel-Schneider, P.F. and Rudolph, S.: OWL 2 Web Ontology Language Primer. `http://www.w3.org/TR/owl2-primer/` (2009)

ID	DL	URI	Axioms	Logical Axioms	(C.1)	(C.2)	(C.3)	(C.4)	(C.2)*	(C.4)*
1.	ALUQ(D)	http://omv.ontoware.org/2009/09/OWLChanges	186	100	100.00	49.49	0.00	N/A	49.49	N/A
2.	SHIN(D)	http://ccdb.ucsd.edu/SAO/1.2	7767	2712	100.00	17.92	97.22	76.17	17.95	76.17
3.	ALEHI+(D)	http://swat.cse.lehigh.edu/onto/univ-bench.owl	243	93	97.67	35.11	36.00	45.16	37.10	45.16
4.	SHOIN(D)	http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine	747	657	94.81	67.88	84.62	66.96	88.29	66.96
5.	ALCO	http://purl.obolibrary.org/obo/ogms.owl	576	84	100.00	50.00	N/A	N/A	50.00	N/A
6.	ALQ(D)	http://owlodm.ontoware.org/OWL2	353	215	90.80	45.50	0.00	N/A	46.91	N/A
7.	SRIQ(D)	http://semanticscience.org/ontology/sio-core.owl	5043	1747	100.00	48.07	100.00	93.55	48.43	93.55
8.	SHOIN	http://www.co-ode.org/ontologies/pizza/pizza.owl	939	712	100.00	22.14	75.00	100.00	22.15	100.00
9.	SHOIN(D)	http://sweet.jpl.nasa.gov/2.1/reprSciUnits.owl	503	475	100.00	8.20	33.33	0.00	42.86	0.00
10.	SR	http://purl.obolibrary.org/obo/hao/2011-11-03/hao.owl	20027	8493	96.22	38.53	0.00	N/A	50.47	N/A
11.	SROIF	http://purl.obolibrary.org/obo/ido.owl	3499	1025	100.00	34.67	86.67	67.02	36.92	67.73
12.	SROIF	http://ontology.neuinfo.org/NIF/Dysfunction/NIF-Dysfunction.owl	5649	350	100.00	50.00	N/A	N/A	50.00	N/A
13.	SHIN(D)	http://aims.fao.org/aos/geopolitical.owl	23527	22834	100.00	100.00	100.00	100.00	100.00	100.00
14.	S	http://human.owl	30364	11545	99.82	49.84	0.00	N/A	49.84	N/A
15.	ALE	http://mouse.owl	11043	4838	67.02	36.99	0.00	N/A	38.93	N/A
16.	ALCHOIN	http://owl.man.ac.uk/2005/07/sssw/people.owl	396	108	100.00	69.94	71.43	81.25	71.18	81.25
17.	SROIF	http://ontology.neuinfo.org/NIF/BiomaterialEntities/NIF-GrossAnatomy.owl	19849	2930	99.75	39.62	18.18	100.00	39.62	100.00
18.	SROIN(D)	http://purl.obolibrary.org/obo/flu/dev/flu.owl	874	204	100.00	100.00	100.00	100.00	100.00	100.00
19.	ALCOIF	http://www.owl-ontologies.com/generations.owl	60	38	94.44	77.14	75.00	100.00	100.00	100.00
20.	AL(D)	http://protege.stanford.edu/plugins/owl/owl-library/ka.owl	404	216	100.00	57.03	0.00	N/A	57.18	N/A
21.	SROIF	http://ontology.neuinfo.org/NIF/BiomaterialEntities/NIF-Cell.owl	3508	398	100.00	46.73	0.00	N/A	46.73	N/A
22.	SOIN(D)	http://www.owl-ontologies.com/travel.owl	145	93	100.00	49.61	33.33	100.00	50.00	100.00
23.	ALUHN	http://www.hozo.jp/owl/YAMATO20120714.owl	4428	2444	100.00	51.69	95.63	11.67	51.69	11.67
24.	ALHIF(D)	http://purl.org/net/ontology/beer.owl	165	81	84.48	42.86	22.22	100.00	42.86	100.00
25.	SH(D)	http://msi-ontology.sourceforge.net/ontology/NMR.owl	1096	290	100.00	50.00	N/A	N/A	50.00	N/A
26.	SHIF(D)	http://www.nada.kth.se/~mehrana/Delegation.owl	103	63	63.16	40.91	80.00	81.25	40.91	81.25
27.	ALCRIQ(D)	http://www.biomodels.net/kisao/KISAO	2044	647	100.00	42.70	100.00	92.23	42.82	92.47
28.	ALCO	http://purl.obolibrary.org/obo/omrse.owl	99	25	100.00	50.00	0.00	N/A	50.00	N/A

Table 3. Dataset and coverage results presented in Section 4. Ontologies 14 and 15 are part of the dataset used by *Ontology Alignment Evaluation Initiative*, and are available at http://oaei.ontologymatching.org/. Coverage results are unaggregated and based on the data of Table 4. N/A means the ontology contains no roles/adaptable roles.

ID	Concepts	Adaptable Concepts	Concept Axioms	Adaptable Concept Axioms	Roles	Adaptable Roles	Role Axioms	Adaptable Role Axioms	Unsatisfiable Concept Axioms	Unsatisfiable Role Axioms
1.	100	100	198	98	2	0	0	0	0	0
2.	736	736	5129	919	36	35	256	195	8	0
3.	43	42	131	46	25	9	31	14	7	0
4.	77	73	411	279	13	11	345	231	95	0
5.	92	92	168	84	0	0	0	0	0	0
6.	87	79	367	167	44	0	0	0	11	0
7.	1021	1021	2823	1357	184	184	946	885	21	0
8.	100	100	1477	327	8	6	180	180	1	0
9.	12	12	183	15	6	2	137	0	148	136
10.	1930	1857	9809	3779	4	0	0	0	2321	0
11.	509	509	2189	759	30	26	285	191	133	3
12.	352	352	700	350	0	0	0	0	0	0
13.	12	12	460	460	6	6	4888	4888	0	0
14.	3304	3298	10880	5423	2	0	0	0	0	0
15.	2744	1839	6256	2314	3	0	0	0	312	0
16.	60	60	173	121	14	10	48	39	3	0
17.	1628	1624	6116	2423	11	2	86	86	0	0
18.	213	213	447	447	19	19	83	83	0	0
19.	18	17	35	27	4	3	20	20	8	0
20.	96	96	377	215	60	0	0	0	1	0
21.	373	373	794	371	2	0	0	0	0	0
22.	35	35	129	64	6	2	11	11	1	0
23.	925	925	4589	2372	183	175	1542	180	0	0
24.	58	49	105	45	9	2	6	6	0	0
25.	301	301	580	290	0	0	0	0	0	0
26.	19	12	22	9	20	16	48	39	0	0
27.	202	202	1335	570	9	9	386	356	4	1
28.	27	27	50	25	2	0	0	0	0	0

Table 4. Raw data used for coverage analysis of Section 4. Unsatisfied axioms stands for axioms matching the LHS of a rule but the entity under deletion failed to satisfy rule precondition.