Mapping Adaptation Actions for the Automatic Reconciliation of Dynamic Ontologies

Julio Cesar Dos Reis
CR SANTEC, CRP Henri Tudor
L-4362, Luxembourg
LRI, University of Paris-Sud XI
julio.dosreis@tudor.lu

Duy Dinh
CR SANTEC, CRP Henri Tudor
L-4362, Luxembourg
duy.dinh@tudor.lu

Cédric Pruski
CR SANTEC, CRP Henri Tudor
L-4362, Luxembourg
cedric.pruski@tudor.lu

Marcos Da Silveira
CR SANTEC, CRP Henri Tudor
L-4362, Luxembourg
marcos.dasilveira@tudor.lu

Chantal Reynaud-Delaître
LRI, University of Paris-Sud XI
F-91405, France
chantal.reynaud@lri.fr

ABSTRACT
The highly dynamic nature of domain ontologies has a direct impact on semantic mappings established between concepts from different ontologies. Mappings must therefore be maintained according to ongoing ontology changes. Since many software applications exploit mappings for managing information and knowledge, it is important to define appropriate adaptation strategies to apply to existing mappings in order to keep their validity over time. In this article, we propose a set of mapping adaptation actions and present how they are used to maintain mappings up-to-date based on ontology change operations of different nature. We conduct an experimental evaluation using life sciences ontologies and mappings. We measure the evolution of mappings based on the proposed approach to mapping adaptation. The results confirm that mappings must be individually adapted according to the different types of ontology change.

Categories and Subject Descriptors
D.2.12 [Interoperability]; Semantic Techniques

General Terms
Formalism, Design, Experimentation

Keywords
Ontology mappings, mapping evolution, mapping adaptation, mapping maintenance, ontology evolution

1. INTRODUCTION
Ontologies offer means to make the semantics of information explicit. Different software applications from various domains use this formalism for supporting distinct tasks such as automatic reasoning, information retrieval \footnote{12}, semantic data annotation \footnote{1}, as well as data interpretation \footnote{13}. An ontology is normally constructed to cover a specific domain, and interconnecting different ontologies is necessary if the underlying information is described by several ontologies. Mappings are the materialization of the semantic relations between elements of interrelated ontologies \footnote{6}.

Creating mappings between ontologies is a complex task, and especially due to the increasing size of ontologies. To this end, alignment techniques have been proposed \footnote{6}, but significant manual efforts of validation are still demanded if a certain level of quality is required \footnote{17}. In addition to the laborious work to create mappings, the dynamic nature of knowledge induces continuous changes in existing ontologies. Ontology Changes Operations (OCO) include revision, deletion and addition of ontology elements (e.g., attributes, concepts and relationships between concepts), which directly influence the reliability of mappings making them invalid.

This problem prevents software applications relying on mappings to fully exploit them. In order to avoid the costly ontology re-alignment process, it is crucial to have efficient strategies of mapping adaptation to keep mappings semantically valid (i.e., up-to-date). Manual maintenance of mappings is an alternative only if modifications are applied to a restricted number of mappings. Otherwise supporting methods and automatic tools are required for large and highly dynamic ontologies. For example, biomedical ontologies usually contain hundred of thousands of concepts interconnected via mappings. Therefore, tackling (semi-) automatically the reconciliation of dynamic ontologies may decrease the time and human efforts to maintain the semantic validity of mappings.

Coping with the mapping maintenance problem in a (semi) automatic way entails many research challenges. First, it is difficult to evaluate the real impact of the ontology evolution on mappings. For instance, changing an attribute value may lead to invalidate a mapping in some cases. In these situations, the challenging issue is to identify and classify the different cases. Second, several types of OCO can be applied to an ontology, but it is unknown how these different types of operations should be duly taken into consideration for mapping adaptation. Thus, generally only the removal of concepts is addressed \footnote{15}. Third, the design of
possible strategies of mapping adaptation according to the different types of OCO is preliminary coped within existing approaches. Fourth, the adaptation of mappings with several types of semantic relations need deeper studies.

In this article, we propose an approach to select the appropriate strategy for the (semi-)automatic adaptation of mappings according to the different types of OCO. For this purpose, we formalize Mapping Adaptation Actions (MAA) expressing different behaviours of mapping adaptation that are used in the mapping maintenance process. The proposed approach faces a particular challenge in determining the most adequate concept of the context to be considered in the adaptation of each impacted mapping. In summary, we make the following contributions:

- We propose a set of MAA and we formalize them for the purpose of maintaining mappings automatically. Aiming at adequately applying these actions for each individual mapping affected by ontology evolution, we investigate the most suited action(s) according to the different types of OCO.
- We conduct a series of experiments using real-world life science ontologies and their associated mappings, for which we observe the evolution of mappings according to the MAA for the most representative types of OCO. Results reveal the relevance of applying the suggested MAA individually for each mapping associated to concepts affected by ontology changes.

The remainder of this article is structured as follows: Section 2 presents the preliminaries. Section 3 presents the proposed approach to mapping adaptation based on OCO. Section 4 presents the experiments. Section 5 describes the proposed approach to mapping adaptation based on OCO. The remainder of this article is structured as follows: Section 2 presents the preliminaries. Section 3 presents the proposed approach to mapping adaptation based on OCO.

2. PRELIMINARIES

The definitions presented in this section are used to detail the addressed problem and to describe our approach.

Ontology. An ontology $O$ specifies a conceptualization of a domain in terms of concepts, attributes and relationships. Formally, an ontology $O = (C, R, A)$ consists of a set of concepts $C$ interrelated by directed relationships $R$. A leaf concept has no sub concept. One concept may have one or more super concepts and/or sibling concepts. Each concept $c \in C$ has a unique identifier and is associated with a set of attributes $A(c) = \{a_1, a_2, \ldots, a_p\}$ (e.g., label, definition, synonym, ...). Furthermore, each attribute is defined for a particular objective, e.g., “label” for denoting concept names or “definition” for giving the meaning in the context where the concept is used. Each relationship $r \in R$ is typically a triple $(c_1, r, c_2) = r(c_1, c_2)$ where $r$ is the relationship symbol (e.g., “isa”, “part_of”, “advised_by” ...) used to interconnect both concepts $c_1, c_2$.

Context of a concept. We define the context of a particular concept $c_i \in O$ as a set of super concepts, sub concepts and sib (sibling concepts) of $c_i$, as following:

$$CT(c_i) = \text{sup}(c_i) \cup \text{sub}(c_i) \cup \text{sib}(c_i)$$

where

$$\text{sup}(c_i) = \{c_j | c_j \in O, c_j \sqsupseteq c_i \land c_i \neq c_j\}$$

$$\text{sub}(c_i) = \{c_j | c_j \in O, c_j \sqsubseteq c_i \land c_i \neq c_j\}$$

$$\text{sib}(c_i) = \{c_j | c_j \in O, \text{sup}(c_j) = \text{sup}(c_i) \land c_i \neq c_j\}$$

where $c_i \sqsupseteq c_j$ stands for “$c_i$ is a sub concept of $c_j$”. This definition of $CT(c_i)$ is specially designed as the relevant concepts to be taken into account in the settings of this investigation on mapping adaptation.

Mapping. Given two concepts $c_s$ and $c_t$ from two different ontologies, a mapping $m_{st}$ can be defined as:

$$m_{st} = (c_s, c_t, \text{semType}, \text{conf}, \text{status})$$

where semType is the semantic relation connecting $c_s$ and $c_t$. In this article, we differentiate relation from relationship, where the former belongs to a mapping and the later to an ontology. The following types of semantic relation are considered: unmappable $[\perp]$, equivalent $[\equiv]$, narrow-to-broad $[\leq]$, broad-to-narrow $[\geq]$ and overlapped $[\approx]$. For example, concepts can be equivalent (e.g., “head”“=“head”), one concept can be less or more general than the other (e.g., “thumb”“≤”“finger”) or concepts can be somehow semantically related ($\approx$). The conf is the semantic similarity between $c_s$ and $c_t$ indicating the confidence of their relation [6]. A status is useful for describing the state of a mapping during the adaptation process.

Ontology change operations (OCO). We consider the ontology change operations presented in Table 1. They are classified into two main categories: atomic and complex changes. Each operation in the former can not be divided into smaller operations while each one of the latter is composed of more than one atomic operations. For instance, the operation $\text{chgA}(c, a, v)$ is composed of two atomic operations $\text{delA}(a, c)$ and $\text{addA}(a, c)$.

Mapping adaptation problem. Given two versions of the same source ontology, namely $O_S^0$ at time $t_0$ and $O_S^1$ at time $t_1$, one target ontology $O_T^1$, and an initial set of valid mappings $M_{ST}^0$ between $O_S^0$ and $O_T^0$ at time $t_0$. Suppose that the frequency of new releases of $O_S$ and $O_T$ are different and at time $t_1$ only $O_S$ evolves. Since this evolution is likely to impact the mappings $M_{ST}^0$, the necessary mapping adaptation actions are applied to $M_{ST}^0$ to guarantee the mapping validity, generating $M_{ST}^1$. The validity stands for the logical consistency of the mappings. For instance, mappings are not established with removed concepts. The mapping adaptation problem, therefore, refers to the process that existing mappings are modified according to changes affecting KOS entities in order to keep mappings valid and complete over time. Figure 1 illustrates the general scenario of this work where we consider that $O_T^0$ and $O_T^1$ are the same (i.e., $OCO_T = \emptyset$).

![Figure 1: Mapping adaptation based on ontology changes](image-url)
3. ADAPTING MAPPINGS ACCORDING TO ONTOLOGY CHANGES

We present an approach to adapting semantic mappings based on different types of ontology changes (Table 1). The proposal explores the CT of the mappings’ source concept (CT(c₀)) for adapting mappings individually. Given a mapping mₛₜ associated to a concept cᵢ affected by changes in the ontology, the challenging issue is to determine an exact and suited action of adaptation to apply to mₛₜ in order to keep such mapping up-to-date. To address this challenge, we define and formalize a set of mapping adaptation actions (MAA). We aim at using MAA as pre-defined behaviours of mapping adaptation into algorithms designed to maintain the semantic validity of mappings according to ontology changes. Based on the nature of different types of OCO, we propose algorithms to determine the most accurate MAA for each mapping impacted by an ontology change. Regarding the adaptation process, the algorithms consider different delimitations of the CT(cᵢ) according to the types of change. The necessary instances of OCO are identified from one ontology version at time t₀ to another at time t₁ with a diff computation [10]. It generates a diff which is basically a set of changes identified between two versions of ontologies. This article considers only the changes affecting Oₛ, i.e., diffₒₛ.

We have previously designed and evaluated an algorithm for identifying the most relevant concept’s attributes based on which mappings are defined [3]. Given a mapping mₛₜ between two concepts cₛ ∈ Oₛ and cᵢ ∈ Oᵢ, the proposed algorithm, namely getTopA(mₛₜ), retrieves the minimum set of source concept’s attributes that are the most similar to the ones in the target concept. Attributes are retrieved directly from cₛ or its context CT(cₛ). For this purpose, the edit-distance similarity measure [14] is used to quantify the semantic relatedness between concepts’ attributes. The proposition of selecting top attributes is to identify key elements used to define existing mappings. The top source concept’s attributes selected are then exploited for supporting the adaptation of mappings according to ontology changes.

In what follows, we describe and formalize the proposed MAA (Section 3.1). Afterwards, each section is devoted to a set of OCO. For each section, we propose an algorithm illustrating the adaptation of mappings which explores getTopA, MAA and CT. Particularly, Section 3.2 concerns the adaptation of mappings based on OCO regarding revision of concepts. Section 3.3 accounts for the changes of removal of attributes and concepts, and Section 3.4 represents the approach to tackling the addition of attributes and concepts.

### 3.1 Mapping Adaptation Actions

We propose five distinct actions that represent different possibilities for adapting mappings: remove, addition, move, derivation and modification of mappings. In the following, we formally describe each action. To this end, let mᵢₛ⁺ ∈ Mᵢₛ (resp. mᵢₛ⁻ ∈ Mᵢₛ⁻) be the mapping between two particular concepts cᵢ⁺ ∈ Oᵢ⁺ (resp. cᵢ⁻ ∈ Oᵢ⁻) and cₛ⁺ ∈ Oₛ⁺ (resp. cₛ⁻ ∈ Oₛ⁻) at time t₀ (resp. t₁). Moreover, we suppose that cₛ does not change while the concept cᵢ evolves from one ontology version to another.

**Remove of mapping.** This is an atomic action through which a mapping mₛₜ is deleted from Mₛₜ:

\[ \text{removeM} (mₛₜ) \rightarrow mₛₜ⁺ \in Mₛₜ⁺ \land mₛₜ⁻ \notin Mₛₜ⁻ \]

**Addition of mapping.** This is an atomic action through which a new mapping mₛₜ is added to Mₛₜ:

\[ \text{addM} (mₛₜ) \rightarrow mₛₜ⁺ \notin Mₛₜ⁺ \land mₛₜ⁻ \in Mₛₜ⁻ \]

**Move of mapping.** This is a composed action for which an existing mapping from Mₛₜ is re-allocated in Mₛₜ⁺, thus the source concept is different. This action is important for adapting mappings by reusing an existing mapping which can be considered invalid in Mₛₜ due to OCO affecting cₛ.

<table>
<thead>
<tr>
<th>Change operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Addition of a new concept c ∈ Oₛ⁺</td>
</tr>
<tr>
<td>t</td>
<td>Deletion of an existing concept c ∈ Oₛ⁻</td>
</tr>
<tr>
<td>o</td>
<td>Addition of a new attribute a to a concept c ∈ Oₛ⁺</td>
</tr>
<tr>
<td>m</td>
<td>Deletion of an attribute a from a concept c ∈ Oₛ⁻</td>
</tr>
<tr>
<td>i</td>
<td>Addition of a new relationship r between two concepts c₁ and c₂ which belongs to Oₛ⁺</td>
</tr>
<tr>
<td>c</td>
<td>Deletion of an existing relationship r between two concepts c₁ and c₂ which belongs to Oₛ⁻</td>
</tr>
<tr>
<td>chgA(c, a, v)</td>
<td>Change of attribute a in concept c with the new value v</td>
</tr>
<tr>
<td>moveC(c, p₁, p₂)</td>
<td>Moving of concept c (and its subtree) from concept p₁ to concept p₂</td>
</tr>
<tr>
<td>C</td>
<td>Replacement of concept cᵢ ∈ Oᵢ by concept cⱼ ∈ Oᵢ</td>
</tr>
<tr>
<td>o</td>
<td>Fusion of a set of multiple concepts Cₖ ∈ Oᵢ into concept cᵢ ∈ Oᵢ</td>
</tr>
<tr>
<td>m</td>
<td>Split of concept cᵢ ∈ Oᵢ into a set of resulting concepts Cᵣ ⊂ Oᵢ</td>
</tr>
<tr>
<td>p</td>
<td>Sets status of concept c to obsolete (c is no longer available)</td>
</tr>
<tr>
<td>l</td>
<td>Deletion of concept cᵢ where pᵢ ∈ sup(cᵢ) and sub(cᵢ) ≠ ∅ from ontology Oᵢ</td>
</tr>
<tr>
<td>e</td>
<td>Deletion of leaf concept cᵢ where pᵢ ∈ sup(cᵢ) and sub(cᵢ) = ∅ from ontology Oᵢ</td>
</tr>
<tr>
<td>x</td>
<td>Addition of a new concept cᵢ under the concept pᵢ ∈ sup(cᵢ) to the ontology Oᵢ</td>
</tr>
<tr>
<td>addLeafC(cᵢ, pᵢ)</td>
<td>Addition of leaf concept cᵢ where pᵢ ∈ sup(cᵢ) and sub(cᵢ) = ∅ to the ontology Oᵢ</td>
</tr>
<tr>
<td>revokeObsolete(c)</td>
<td>Revokes obsolete status of concept c (i.e., c becomes active)</td>
</tr>
</tbody>
</table>

Table 1: Ontology change operations [10]
The mapping is thus adapted considering its CT(c_i).

\[
\text{moveM}(m_{st}, c_i^0) \rightarrow m_{st}^0 \in M_{ST}^0 \land m_{st}^0 \notin M_{ST}^1 \land
(\exists c_i^0 \in CT(c_i), m_{st}^0 \in M_{ST}^0 \land sim(c_i^0, c_i^0) \geq \sigma) \lor
(\exists c_i^0 \in CT(c_i), m_{st}^0 \in M_{ST}^0 \land sim(c_i^0, c_i^0) \geq \sigma)
\]

where \(sim(c_i, c_j)\) denotes the similarity between \(c_i\) and \(c_j\).

**Derivation of mapping.** This is a composed action for which an existing mapping in \(M_{ST}^0\) has a modified copy in \(M_{ST}^1\) with a different source concept. This action is important for reusing an existing mapping, which is still considered as valid in \(M_{ST}^1\).

\[
\text{deriveM}(m_{st}, c_i^0) \rightarrow m_{st}^0 \in M_{ST}^0 \land m_{st}^0 \in M_{ST}^1 \land
(\exists c_i^0 \in CT(c_i), m_{st}^0 \in M_{ST}^0 \land sim(c_i^0, c_i^0) \geq \sigma)
\]

**Modification of semantic relation.** This is a composed action in which the type of the semantic relation of a given mapping is modified. This action is designed for supporting the adaptation of mappings with different types of semantic relations rather than only considering the type of equivalence relation (\(\equiv\)).

\[
\text{modSemTypeM}(m_{st}, \text{new}_{\text{SemType}}) \rightarrow m_{st}^0 \in M_{ST}^0 \land
\text{new}_{\text{SemType}} = \text{new}_{\text{SemType}} \land \text{sim}_{\text{new}} \geq \text{sim}_{\text{old}}
\]

The action for the modification of semantic relation can be applied in conjunction with the actions of move or derivation of mapping. That is when moving/deriving a mapping, it is also possible to modify the type of the semantic relation of such mapping.

### 3.2 Mapping adaptation according to the revision of knowledge in ontology

The revision of knowledge in ontology is performed by the following OCOs: chgA(c_i, a, v), sub(c_i, c_j), merge(C_i, c_j), and split(c_i). The main characteristic of these operations is that the involved concepts have some semantic similarity. There is normally a flow of information between concepts belonging to the same complex change that should be explored when adapting mappings. Algorithm 1 presents the proposed strategy for adapting mappings associated to concepts affected by revision OCOs. These mappings are adapted considering a delimited context, which is represented by the concepts resulting from the complex change. The resulting set of concepts in these operations is \(C_r \subseteq CT(c_i)\) such that \(c_j \in C_r\) (for substitute and merge). Note that \(C_r \subseteq CT(c_i)\) represents a delimited context for adapting mappings. Algorithm 1 is applied for all instances of revision OCO retrieved from the \(\text{diff}_{O_0}O_0\).

Concepts in these operations are the input of Algorithm 1. Indeed, it requires two different set of concepts: \(C_k \subseteq O_0\) representing the original concepts before evolution and \(C_r \subseteq O_0\) the resulting concepts in the complex change. The input concepts of these sets are based on the type of OCO as well as the context where mappings are adapted. Mappings associated with concepts affected by chgA operation are not adapted considering other concepts of the context. In the substitute and merge operations, an initial concept \(c_i \in O_0\) or a set of concepts \(C_k \subseteq O_0\) (in case of merge) gives place to a concept \(c_j \in O_1\). The input data of the algorithm by these two types of operations are \(c_i \in C_k\) for substitute and \(C_k\) for merge while \(c_j \in C_r\) is used in both operations, which is the delimited context considered. In the split(c_i, C_r), an initial concept \(c_i \in O_0\) gives place to a set of (new) related ones in \(C_r \subseteq O_1\). In this case, the input is slightly different from substitute and merge. In split, \(c_i \in C_k\) with many resulting concepts in \(C_r\).

Algorithm 1: Adaptation of mappings according to revision of knowledge in ontology

**Require:** \(C_k \subseteq O_0\); \(C_r \subseteq O_0\)

1: for all \(c_i \in C_k\) do
2: \(M_{\text{ci}} \leftarrow \text{getAssociatedMappings}(c_i)\)
3: for all \(m_{ij} \in M_{\text{ci}}\) do
4: \(S \leftarrow \emptyset\); \{Initialize the result set for each mapping to be adapted\}
5: \(\text{TopAm}_{m_{ij}} \leftarrow \text{getTopA}(m_{ij})\)
6: for all \(s_n \in \text{TopAm}_{m_{ij}}\) do
7: if delA(a_n) \in diff\ O_0\ then
8: mapImpacted \leftarrow TRUE
9: for all \(c_i \in C_r\) do
10: for all \(a_r \in A(c_i)\) do
11: \(s_n \leftarrow \text{sim}(a_n, a_r)\);
12: \(S \leftarrow S \cup \{a_r, s_n, c_i\}\);
13: end for
14: end for
15: end if
16: end for
17: if \(S \neq \emptyset\) then
18: \(S \leftarrow \text{sort}(S);\{\text{sort results}\}\)
19: \(c_f \leftarrow \text{getFirst}(S)\);
20: if \(c_f \in O_1\) then
21: \(\text{moveM}(m_{ij}, c_f)\);
22: \(\text{modSemTypeM}(m_{ij}, \text{semType})\);
23: end if
24: else
25: if mapImpacted = \text{TRUE} \lor c_i \in O_0\ then
26: \(\text{removeM}(m_{ij})\);
27: end if
28: end if
29: end for
30: end for

According to Algorithm 1, given a concept \(c_i \in C_k\) (line 1), it first selects the mappings associated with the concept \(c_i\) (line 2). These mappings are adapted individually and for each of them, the algorithm applies the getTopA(m_{ij}) to identify the relevant attributes of \(c_i\). Based on this result, it checks whether the selected attributes are affected by a delA(a) change operation (line 7). In this case, the mapping is assigned as an impacted mapping since a relevant information defining mapping \(m_{ij}\) was removed. If a selected
attribute is affected, the mapping is likely to be adapted. We assume that mappings associated with unchanged attribute values remain valid. When a mapping \( m_{ij} \) is impacted, Algorithm 1 explores \( C_r \) (line 9) by finding a candidate concept to perform a move of mapping. If concept \( c_r \) remains in \( O_1^S \), i.e., \( c_i = c_r \in C_1^r \), a mapping \( m_{ij} \) can remain attached to this concept whether the most similar attribute value is still found in this concept. However, it is necessary to guarantee that concept \( c_r \) remains active in \( O_0^S \). Similarly, in merge operation, if \( c_r = c_b \in C_b \), then the associated mapping can be moved to the same source concept (i.e., it is still associated with the same source concept identifier at time \( t_0 \) and \( t_1 \)). The evaluation in Section 4 considers these mappings as unchanged.

Differently, the mapping can be moved to another resulting concept \( c_r \), e.g., \( c_r \in sib(c_i) \), such that \( c_r \in C_r \) and \( c_r \neq c_i \). The most adequate candidate is calculated based on the semantic similarity between the value of affected attributes selected for a mapping and the attributes’ value of each concept in \( C_r \) (line 11). The best candidate contains the highest similarity value by sorting \( S \) (line 18) which is used to apply a moveM action (line 21). If no candidate is found, there is no concept \( c_r \in C_r \) that contains explicitly a piece of conceptual information the mapping \( m_{ij} \) was related to, or if the mapping was assigned to impacted, since a relevant attribute identified was deleted, the algorithm proposes a removeM action (line 26).

Figure 2 presents the general idea of adapting mappings according to the revision of knowledge in ontologies. Given the evolution of the concept \( c_{i1} \in O_0^S \), three mappings were associated with this source concept at time \( t_0 \). After evolution, there is a split of the concept \( c_{i1} \) with two sibling concepts on which a mapping \( m_1 \) remains attached to \( c_{i1} \), and two other mappings are moved to resulting concepts accordingly:

![Figure 2: Mapping adaptation according to the revision of attributes and concepts](image)

### 3.3 Mapping adaptation according to the removal of knowledge in ontology

The removal of knowledge in ontology includes the OCOs delA \((a)\), delInnerC \((c)\), delLeafC \((c)\) and toObsolete \((c)\). Normally, these operations are less frequent compared to the others, since there is a natural tendency to expand the knowledge in the ontology rather than the removal of knowledge.

They are usually applied when concepts turn out of scope. Anyway, they can have a great impact on the validity of existing mappings.

Algorithm 2 presents the proposed strategy for adapting mappings associated with concepts affected by removal-based OCOs. The input of the algorithm is the set of concepts from the \( \text{diff} \ O_0^S \), where attributes are deleted from concept(s). We assume that the deletion of a concept (or its assignment to obsolete) is the consequence of deleting all attributes (or assigning them to obsolete). Since our approach tackles the ontology mapping adaptation based on the information at the level of attributes, the designed algorithm is suited for handling deletion changes affecting either the entire concept and/or the attributes as a part of the evolving concept. In Algorithm 2, the moveM action is proposed to adapt an impacted mapping \( m_{ij} \) to one of the concepts in the \( \text{CT}(c_r) \), instead of directly removing the mapping. For this purpose, only attributes identified by the algorithm getTopA are considered.

#### Algorithm 2: Adaptation of mappings according to the removal of knowledge in ontology

**Require:** \( C_{rem} \subseteq O_0^S \); \{concept having at least one deleted attribute\)

1: for all \( c_i \in C_{rem} \) do
2: \( A_{remc_i} \leftarrow \text{getRemovedAttributes}(c_i) \)
3: \( M_{ci} \leftarrow \text{getAssociatedMappings}(c_i) \)
4: for all \( m_{ij} \in M_{ci} \) do
5: \( S \leftarrow \emptyset \); \{Initialize the result set for each mapping to be adapted\}
6: \( \text{TopA}_{m_{ij}} \leftarrow \text{getTopA}(m_{ij}) \)
7: for all \( a_n \in \text{TopA}_{m_{ij}} \) do
8: if \( a_n \in A_{remc_i} \) then
9: \( \text{mapImpacted} \leftarrow \text{TRUE} \)
10: for all \( c_w \in \text{CT}(c_i) \) do
11: for all \( a_w \in A(c_w) \) do
12: \( s_{nw} \leftarrow \text{sim}(a_n, a_w) \); \( S \leftarrow S \cup \{ (a_w, s_{nw}, c_w) \} \); end for
end for
14: end if
15: end for
16: end if
17: end for
18: if \( S \neq \emptyset \) then
19: \( S \leftarrow \text{sort}(S) \); \{sort results\}
20: \( c_f \leftarrow \text{getFirst}(S) \);
21: if \( c_f \in O_0^S \) then
22: \( \text{moveM}(m_{ij}, c_f) \)
23: \( \text{modSemTypeM}(m_{ij}, \text{semType}) \)
24: end if
25: else
26: if \( \text{mapImpacted} = \text{TRUE} \) then
27: \( \text{removeM}(m_{ij}) \)
28: end if
29: end if
30: end for
31: end for

According to Algorithm 2, given a concept \( c_i \) affected by a removal change operation (line 1), it first selects all deleted attributes of concept \( c_i \) and the mappings associated with the concept (lines 2-3). For each mapping, it applies the algorithm getTopA to identify the relevant attributes. Then,
it checks whether the selected attributes belong to a deleted attribute of the concept (line 8). In case a selected attribute is deleted, the mapping must be adapted. When a mapping is impacted, Algorithm 2 explores the CT(c_i) (line 10) by finding a candidate to perform a move of the mapping. Different from the revision of knowledge, there is not a well-delimited context of resulting concepts of a complex change, and thus the algorithm searches for the best candidates among all super, sub and sibling concepts. To this end, it calculates the semantic similarity, among the value of the affected attributes selected for a mapping with the value of attributes of each concept of the CT(c_i) (line 12), in order to select possible candidates for applying a moveM action. The best candidate considered contains the highest similarity value by sorting S (line 19-20). If the candidate is found, a move of mapping is applied towards the selected concept of the CT(c_i) (line 22). If no candidate is found, and the mapping was impacted by a change in at least one attribute of the corresponding source concept among the ones identified as top related attributes for defining the mapping, then the algorithm proposes a removeM action (line 27).

Figure 3 presents the general idea of adapting mappings according to the removal of knowledge in ontologies. Given the evolution of the concept c_s ∈ O_S, three mappings were associated with this concept at time t0. After evolution, some attributes belonging to c_s are deleted or the whole concept is removed, and as a consequence one mapping is removed while two others are moved to concepts in the context of c_s.

**3.4 Mapping adaptation according to the addition of knowledge in ontology**

Addition of knowledge in ontology is performed by addA(a), addInnerC(c), addLeafC(c) and revokeObsolete(c). When handling these OCOs, the proposed strategy of mapping adaptation explores the action of derivation of mappings. In order to generate a complete M_{CFP} as a result of mapping adaptation, we aim to reuse mappings associated with concepts, which are in the context of a concept affected by an addition OCO, benefiting of already valid established mappings.

Algorithm 3 presents the approach to adapting mappings by deriving them according to addition OCO. The input of the algorithm is the set of concepts obtained from the diffO_S which are affected by some addition of attributes(s). Note that when an addition of concept occurs (even inner or a leaf concept) or a concept is assigned to revokeObsolete, we assume that the whole set of attributes is also added. Similarly, as proposed in Algorithm 1 and 2 the approach here also tackles the ontology mapping adaptation based on the information in the level of attributes. Hence, the designed algorithm is suited for handling addition changes affecting the entire concept or in the level of attributes.

**Algorithm 3: Adaptation of mappings according to the addition of knowledge in ontology**

Require: C_{add} ⊂ O_S: {concepts having at least one added attribute and concepts identified as revokeObsolete}

1: for all c_i ∈ C_{add} do
2: if hasNoMappingAssociated(c_i) then
3: A_{addA(i)} ← getAddedAttributes(c_i)
4: CT_{c_i} ← CT(c_i)
5: for all c_k ∈ CT_{c_i} do
6: M_{c_k} ← getAssociatedMappings(c_k)
7: for all m_{kj} ∈ M_{c_k} do
8: S ← ∅;
9: TopA_{m_{kj}} ← getTopA(m_{kj})
10: for all a_w ∈ TopA_{m_{kj}} do
11: if delA(a_w) ∉ diffO_S then
12: for all a_w ∈ A_{addA(i)} do
13: s_{nw} ← sim(a_n, a_w);
14: S ← S ∪ {s_{nw}, m_{kj}};
15: end if
16: end for
17: end for
18: end for
19: end for
20: if S ≠ ∅ then
21: S ← sort(S); {sort results}
22: m_f ← getFirst(S);
23: deriveM(m_f, c_i);
24: modSemTypeM(m_f, semType);
25: end if
26: end if
27: end for

The reasoning behind Algorithm 3 is that existing mappings associated with concepts of the context of the new (attribute) concept are considered candidate mappings to be adapted for putting in correspondence the concept affected by an addition OCO. Given a concept c_i affected by an addition OCO (line 1), Algorithm 3 first verifies if the concept c_i has still no mapping associated. Concepts already mapped are not taken into account. Afterwards, the algorithm selects the added attributes and the concepts of the context of c_i (lines 3-4). For each one of these concepts, the associated mappings are selected. For each mapping m_{kj}, Algorithm 3 applies the getTopA(m_{kj}) for identifying the relevant attributes. Based on this result, the algorithm checks whether the set of attributes identified belongs to a deleted attribute of the concept (line 11). If the attribute is not affected by a change, the mapping m_{kj} remains valid. In this case, it can be considered as an adequate candidate for the derivation. Using the method of calculating the semantic similarity between attributes’ value, a selection of candidate mappings...
is performed. The best candidate in sorted $S$ contains the maximum similarity value (line 21-22). If a candidate mapping is found, the derivation $M$ action is applied using the selected mapping associated with a concept of the CT($c_i$) (line 23).

Figure 4 presents the general idea of adapting mappings according to the addition of knowledge in ontologies. Consider the ontology at time $t_0$ including concepts containing mappings ($m_1, m_2, m_3$). After evolution, a new concept $c_{x+1}$ is added and a mapping related to a super concept is derived considering $c_{x+1}$ and the same target concept $c_i$ of the original mapping $m_1$.

Figure 4: Mapping adaptation according to the addition of attributes and concepts

4. EXPERIMENTS

In order to evaluate our approach we select a set of ontologies and mappings (Section 4.1). We then conduct a series of experiments to evaluate the three proposed mapping adaptation strategies (Section 4.2). We present the obtained results in Section 4.3.

4.1 Ontologies and mappings

We use three large life sciences ontologies: SNOMED-CT (SCT), NCI Thesaurus (NCI) and ICD-9-CM (ICD). Table 2 depicts some statistics regarding the number of concepts, attributes or terms denoting concepts, and the number of relationships between concepts. In this study, we focus on exploiting the hierarchical structure of ontologies. Therefore, relationships in Table 2 correspond to the number of direct subsumption relationships between concepts. Furthermore, we consider the interval of three years from 2009 to 2012 to demonstrate the evolution of ontologies over time when studying the different strategies of mapping adaptation. Table 3 shows the quantity of mappings established between various releases of SCT and NCI, SCT and ICD, respectively from 2009 to 2012. Mappings between SCT and NCI are established between versions in the same year, while the ones between SCT and ICD are established between SCT of version released in year $x$ and ICD of version released in year $x - 1$. Table 4 shows the quantity of changes in ontology per categories of ontology changes studied in the experiments as well as the total number of analyzed mappings associated with evolving concepts.

4.2 Methodology

The objectives of the conducted evaluation are two-fold. First, we aim to evaluate the utility to adapt each mapping individually based on different types of OCO according to the proposed strategies of mapping adaptation. Second, we assess the effectiveness of the proposed strategies. More specifically, our objective is to determine to which extent the different MAAs suggested can be applied individually for each mapping where the source concept suffers at least from one OCO.

We conduct the following procedure in the evaluation. First, we identify OCO for each pair of released versions starting from 2009 until 2012 of SCT, ICD and NCI using the COnto-Diff [10] tool. This generates a set of diffs for each ontology. For each diff we keep only the most relevant types of operations studied in this evaluation. Indeed, we focus on changes in attributes of concepts; therefore, we retain nine OCOs at the level of concepts as presented in Table 4. Then, for all affected concepts of OCO identified in the $diff_{OCont}$, we select the impacted mappings in $M^o_{ST}$. It produces for each type of change a particular subset of mappings which are further examined. In order to analyze the results within a global view, for each type of OCO we group all instances of changes identified in the calculated diffs of all ontologies ICD, SCT and NCI. Here, mappings are studied independently from the ontology structure. With this assumption the subset of associated mappings is also grouped accordingly and the total number of mappings are shown in Table 4. Based on the proposed approach to adapting mappings as described in the algorithms, we evaluate the behaviour of each individual mapping according to the proposed MAAs for each type of OCO. To this end, given a mapping $m_i^o \in M^o_{ST}$ and based on the type of OCO affecting the source concept, we verify that $m_i^o \in M^o_{ST}$ calculating the applied MAA. For each subset of mappings under analysis according to the OCO, we measure the proportion of each type of MAA correctly identified in the mapping evolution from one release to another.

4.3 Results

The results are classified according to the nature of the ontology changes. Figure 5 considers OCO related to revision, removal and addition of knowledge in ontologies. It depicts the percentage of MAAs applied for the studied mappings.
<table>
<thead>
<tr>
<th>OCO</th>
<th>Nb. of ontology changes</th>
<th>Nb. of mappings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Revision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>substitute((a, c_j))</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>merge((C_k, c_j))</td>
<td>0</td>
<td>25</td>
</tr>
<tr>
<td>split((c_i, C_r))</td>
<td>45</td>
<td>225</td>
</tr>
<tr>
<td>Removal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>delInnerC((c_i))</td>
<td>0</td>
<td>99</td>
</tr>
<tr>
<td>delLeafC((c_i))</td>
<td>0</td>
<td>286</td>
</tr>
<tr>
<td>toObsolete((c_i))</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Addition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>addInnerC((c_i))</td>
<td>26</td>
<td>2611</td>
</tr>
<tr>
<td>addLeafC((c_i))</td>
<td>140</td>
<td>17056</td>
</tr>
</tbody>
</table>

Table 4: Number of OCOs vs. number of analyzed mappings associated with evolving concepts

Figure 5: MAAs applied for mappings associated with OCO belonging to one of the three categories of OCO

Figure 6: MAAs applied for the mappings associated with OCO split, toObsolete and addLeafC by the different ontologies. The MAAs applied are denoted using characters from A to S (cf. Figure 5)
For the revision of knowledge, we notice that the MAA moveM(C_r) has been applied with a percentage of 64.69% for the split operation in source concepts. We observe that in some cases, two MAAs moveM(C_r) and modSemTypeM can be simultaneously applied with a percentage of 10.26%. For merge operation, mappings are adapted by applying moveM(C_r) (22.04%) or removeM (18.64%). In the case of substitution of concepts, mappings are more frequently removed (66.67%). The latter can be explained by the fact that whenever the relevant attributes identified in the source concept, which is replaced by another one, are not found in the latter then the mapping is removed. This highlights the relevance of the proposed technique of retrieving the top attributes identified from existing mappings impacted by a change in the resulting concepts of the change.

For the removal of knowledge, mappings can be adapted by removing or moving the source concept to a concept in its hierarchy. When a concept is assigned to toObsolete, several MAAs are applied, but at different frequencies: moveM(sup)—moving the source concept to a super concept (65.26%), moveM(sib)—moving the source concept to a sibling concept (22.71%) and removeM—removing the mapping (9.27%). Observe that mappings are mostly moved to a super concept rather than removed. Mappings are mainly moved to sibling concepts (moveM(sib)) when occurring OCO delLeafC (50%). This MAA is also applied for a significant percentage of mappings (37%) where the source concept is affected by the OCO delInnerC, but mappings are more frequently removed (48%) when inner concepts are deleted. This shows the importance of considering the context with different hierarchical relationships for mapping adaptation.

For the addition of knowledge, most of the time, if there is a new concept added, then a new mapping is established for this concept (i.e., the addM action is applied) while the derivation of mappings (deriveM) is applied only for a small number of mappings. It stresses the importance of other factors for supporting the mapping adaptation particularly for the addition of knowledge.

At this level, we aim to evaluate the behaviours of mappings under evolving ontologies considering each ontology separately. Figure 4 presents a closer analysis examining the three most important OCOs containing the higher number of associated mappings (see Table 1): split, toObsolete and addLeafC. The results show that MAAs are applied depending on the characteristics of the source ontologies and the way that they evolve over time, i.e., the OCOs observed during their evolution. For example, when the source concepts are split into several ones, we observe that the MAA moveM(C_r) is applied at a percentage of 83.94% for ICD while this action has not frequently been applied for SCT and NCI. The removeM action is also applied but with a very low percentage. This highlights the necessity of carefully searching for the relevant conceptual content related to mappings in the context of source concepts and adequate hierarchical relationships between evolving concepts in order to reach a high quality of mapping adaptation. The obtained results reveal that considering the delimited context of complex changes for handling revision OCO and the CT for removal OCO is a relevant approach to adapting mappings. The proposed mapping adaptation strategies based on the MAA according to the OCO can be applied mainly in cases of revision and removal of knowledge in ontologies. A refined strategy in the addition of knowledge requires further studies, but it is not so severe since addition of knowledge is relevant to have a more complete final set of adapted mappings and the impact on the invalidation of existing mappings is low. Results allow to point out the importance of mapping maintenance considering the adaptation of individual mappings. The candidate MAA shall be carefully analyzed for each mapping. Furthermore, there have been evidences that mappings associated with the same type of OCO behave differently, and the designed algorithms are suited to adapt mappings individually based on the conceptual context where they are.

5. RELATED WORK

Different approaches have been proposed to ontology reconciliation under evolution. Three main categories of work can be distinguished. The first one is based on a revision of mappings by identifying and repairing invalid mappings. Melicke et al. [12] propose an automatic debugging of mappings between expressive ontologies eliminating inconsistencies, caused by erroneous mappings, by means of logical diagnostic reasoning. Similarly, Castano et al. [2] suggest a probabilistic reasoning approach for performing the validation of mappings with regard to the semantics of the ontologies involved. These techniques can be applied at ontology evolution time to detect the invalid mappings. However, it demands logically expressive ontologies requiring a high level of formalization.

The second category is based on a full or a partial re-calculation of mappings. While the former does not consider any information from either ontology evolution nor existing mappings, the latter aims at exploiting this information for recreating only mappings that are associated with changed concepts in ontologies. In addition, if large ontologies are frequently released, the full re-calculation of mappings becomes less flexible than a partial re-calculation approach because the costs in terms of processing time for re-aligning ontologies is too expensive. A partial re-calculation approach is proposed by Khattak et al. [11], re-creating only those mappings associated with concepts whose elements have changed. Matching algorithms are used to perform a new alignment between changed concepts issued from source ontology and the whole target ontology. However, large ontologies, as in the life sciences, so far represent a big challenge for methods of mapping calculation [17].

The third category concerns approaches that attempt to adapt mappings in response to ontology evolution. Ontology changes are usually used to support mapping adaptation avoiding to perform calculations for re-aligning ontologies. The first propositions appeared in the context of database schema mappings [19] based on primitive schema changes. Composition of mappings [20] is an approach for adapting mappings in which mappings between different schema versions are explored. Concerning ontologies, Tang & Tang [15] propose a method for ontology evolution whose objective is to find the minimal impact of ontology change propagation. Martins & Silva [14] propose that evolution of mappings should behave similarly with the strategies applied for ontology evolution, but mappings are only adapted when concepts are removed. Aiming at better understanding the mapping evolution, Gross et al. [8] empirically investigated the evolution of life science ontology mappings.

We have conceptualized a framework [5] for supporting the adaptation of semantic mappings highlighting different as-
pects such as: the role of different types of ontology changes, the importance in considering the conceptual information which established mappings are related to, as well as the relevance of the different types of semantic relation of mappings. We also have investigated semi-automatic approaches to adapting ontology mappings when at least one of the mapped ontologies evolves [7]. However, the selection of the most appropriate strategies of mapping adaptation considering each existing mapping individually still remains a research challenge. In order to address it, this article proposed actions suited to adapt mappings considering particular decisions of adaptation. We presented how these actions are correlated to different types of OCO. We further analyzed the evolution of mappings based on the proposed approach.

6. CONCLUSION
The continuous evolution of domain ontologies requires the maintenance of mappings established between them to keep the validity of mappings over time. In this article, we proposed an approach to mapping adaptation based on a categorization of ontology change operations. More specifically, our approach aimed at determining the most relevant actions to perform on existing mappings based on both conceptual information identified for mappings and ontology changes calculated between different versions of ontologies. In order to correctly adapt mappings, we needed to look inside the content of evolving source concepts. The conducted experiments, analysing large life sciences ontologies and their associated mappings, highlighted important facts with respect to mapping adaptation showing evidences of the significance of the proposed approach. The experimental results mostly revealed the relevance of applying particular actions for adapting individual mappings in the maintenance process.

In future work, we plan to refine the MAA defined in this study and investigate additional factors to which MAA shall be applied in order to better support the automatic maintenance of mappings under evolving ontologies.

7. ACKNOWLEDGMENTS
This work is entirely supported by the National Research Fund (FNR) of Luxembourg (grant C10/IS/786147) under the DynaMO research project.

8. REFERENCES