Pattern-based Mapping Refinement

Fayçal Hamdi, Chantal Reynaud, and Brigitte Safar

CNRS - University of Paris-Sud 11 (LRI) & INRIA Saclay Ile-de-France (LEO)
Parc Orsay Université 4 rue Jacques Monod 91893 Orsay France
{Faycal.Hamdi,Chantal.Reynaud,Brigitte.Safar}@lri.fr

Abstract. Semantic alignment between ontologies is a crucial task for information integration. There are many ongoing efforts to develop matching systems implementing various alignment techniques but it is impossible to predict what strategy is most successful for an application domain or a given pair of ontologies. Very often the quality of the results could be improved by considering the specificities of the ontologies to be aligned. In this paper, we propose a pattern-based approach implemented in the TaxoMap Framework helping an engineer to refine mappings to take into account specific conventions used in ontologies. Experiments in the topographic field within the *ANR* (The French National Research Agency) project *GéOnto* show the usefulness of such an environment both for a domain expert and an engineer, especially when the number of mappings is very large.

Key words: Ontology alignment, Mapping refinement

1 Introduction

The explosion of the number of data sources available on the web increases the need for techniques which allow their integration. The ontologies which provide definitions of domain concepts are essential elements in integration systems and the task of ontology alignment is particularly important for making different heterogeneous resources interoperable. The current alignment tools [4] do not have the same efficiency in all application domains or for all pairs of ontologies. They may be very good in some cases, worse in others. The quality of their results is not always guaranteed and could often be improved if the alignment process took more into account the specificities of the aligned ontologies.

Taking into account these specific aspects can be done in different ways: (1) during the alignment process itself or (2) by refining the results generated by the alignment, considered as preliminaries. In the first case, the adaptation of the handled ontologies is made possible by the modification of the alignment process parameters or by the definition of a particular combination of the alignment systems. No differentiation is thus made in the way the different elements of the ontologies are treated. Inversely, the refinement of mappings (the alignment results) extends the alignment process, applied in the same way to all the elements of the ontologies, and completes it. This second solution allows a finer adaptation of the alignment to the specificities of the handled ontologies. It also allows performing differentiated refinements according to the generated results. Our work follows this research direction.

Currently, there is no tool which helps to specify mapping refinement treatments to take into account specific conventions used in the ontologies. The TaxoMap Framework allows such specifications.

The paper is organized as follows. In the next section, we present the context of this work, in particular the ontology alignment tool TaxoMap and the goals of the conception of the TaxoMap Framework. In Section 3 we present our main contributions: a pattern-based approach to help refining mappings, the mapping refinement work-flow implemented in the Framework and MRPL(the Mapping Refinement Pattern Language), the language used in this environment to define mapping refinement patterns. In Section 4 we present some mapping refinement patterns built in the setting of the ANR project *GéOnto* [5]. Experiments in the topographic field which show the usefulness of this environment both for the domain expert and the engineer are described in Section 5. In Section 6 we present some related works. Finally we conclude and give some perspectives in Section 7.

2 Context

We describe the alignment tool TaxoMap [14][6] in Section 2.1 and the objectives of the approach in Section 2.2.

2.1 TaxoMap

TaxoMap has been designed to align owl ontologies $O = (C, H)$. C is a set of concepts where each concept is characterized by a set of labels and H is a subsumption hierarchy which contains a set of *isA* relationships between nodes corresponding to concepts. The alignment process is an oriented process which tries to connect the concepts of a source ontology O_S to the concepts of a target ontology O_T. The correspondences found are equivalence relations (*isEq*), subsumption relations (*isA*) and their inverse (*isMoreGnl*) or proximity relations (*isClose*).

To identify these correspondences, TaxoMap implements techniques which exploit the labels of the concepts and the subsumption links that connect the concepts in the hierarchy [6]. The morpho-syntactic analysis tool, *TreeTagger* [17], is used to classify the words of the labels of the concepts and to divide them into two classes, *full words* and *complementary words*, according to their category and their position in the labels. At first the repartition between *full* and *complementary words* is used by a similarity measure that compares the trigrams of the labels of the concepts [12] and gives more weight to the common *full words*. Then it is used by the alignment techniques. For example, one technique named t_2 generates an *isA* mapping between X and Y if (1) the concept Y is the concept of O_T having the highest similarity value with the concept X of O_S,

(2) one of the labels of Y is included in one of the labels of X, (3) all the words of the included label of Y are classified as *full words* by *TreeTagger*.

Mappings identified by TaxoMap are generated in the Alignment format [3] used as a standard in the OAEI campaign [9]. We added to this format the information about the names of the techniques that generated mappings. The aim is to facilitate the specification of treatments exploiting the mappings generated by those techniques. All these pieces of information are stored in a relational mappings database which can then be queried using *SQL* queries. This allows, in particular, to present the generated mappings to the expert in the validation phase, technique by technique.

2.2 Objectives

Many ontology alignment tools have been developed in these last years but as shown in the results of the OAEI campaigns [9] organized every year since 2004 [1], no tool reaches 100% of precision and recall, even though the results obtained by some of these tools are very good. This also applies to TaxoMap results, either in the OAEI competition in the two last years [7][6] or in the setting of the ANR project *GéOnto* [5]. The aim of this project is the construction of a topographic ontology and its enrichment with elements coming from other geographic ontologies using alignment techniques. In this setting, tests performed on taxonomies provided by the *COGIT-IGN* (project partner) have shown that TaxoMap gives very good results (precision $92,3\%$) but these results could still be improved.

A closer study showed that the improvements desired by the domain experts are rather specific to the aligned ontologies because they depend on the specific conventions used in the pair of ontologies. Our aim was not to turn TaxoMap into a tool dedicated to the alignment of such topographical taxonomies (the quality of the results would not be guaranteed when TaxoMap would be used to align ontologies coming from other domains). Therefore, we proposed to the experts of the *GéOnto* project an environment allowing to specify and perform refinement treatments applied on the prior obtained mappings. At first, this environment will be used to improve the quality of an alignment provided by TaxoMap. Subsequently, it will be used for other treatments based on mappings as enriching, restructuring or merging ontologies.

Such a mapping refinement environment must satisfy two main objectives. First, it must provide the domain experts with a tool helping them to detect and propose corrections for invalid mappings. The validation task is sometimes very difficult because the number of generated mappings can be enormous when the ontologies are very large. The expert may have difficulties to browse all the mappings and to have the global view he requires in order to propose the right modifications. In consequence, he may ask to modify some mappings without realizing that the requested modifications have an undesirable impact on other mappings. The observations of the consequences of the requested updates can be a means for the expert to clarify the right refinement treatments to be performed. Second, thanks to the iterative validation/correction process, such an environment must help the engineer to specify correct treatments. The validation phase performed by the expert allows to check whether the specification of a treatment intended to be applied to a given set of mappings is correct or not (i.e. if it does not also generate undesirable mappings).

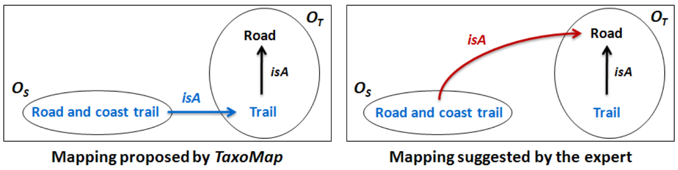
3 The approach

The approach implemented in the TaxoMap Framework has been designed to meet the objectives described in Section 2.2. We describe the approach and a diagram representing the mapping refinement work-flow respectively in Section 3.1 and 3.2. This work-flow allows the specification of treatments according to a pattern-based approach. The language MRPL used to define mapping refinement pattern is presented in Section 3.3.

3.1 Presentation of the approach

An important feature of the approach is to allow a declarative specification of treatments based on particular alignment results, concerning particular ontologies and using a predefined vocabulary. Treatments which can be specified depend on the characteristics of the concerned ontologies and on the task to be performed (at first mapping refinement and subsequently ontology merging, restructuring, enriching). These treatments are thus associated to independent specification modules, one for each task, each having their own vocabulary. The approach is extensible and a priori applicable to any treatment based on alignment results.

In the setting of mapping refinement, the approach should help to specify, for example, that the subsumption mapping *isA* generated between "Road and coast trail" and "Trail", as shown in Fig. 1 must be replaced by a mapping of the same type but between "Road and coast trail" and "Road". Indeed, "Trail" is defined in O_T as a kind of "Road" and the term "Road" itself appears in the label "Road and coast trail". The expert would thus prefer to establish a mapping directly between "Road and coast trail" and "Road".



Mapping proposed by *TaxoMap*

Mapping suggested by the expert

Fig. 1. Example of update asked by the expert

The specification of treatments must be as generic as possible. Thus, the specification of the treatment illustrated in Fig. 1 should not refer directly to the concepts denoted by "Road", "Trail" and "Road and coast trail". Instead, we provide the engineer with a vocabulary allowing to specify mapping refinement patterns. These patterns are generic specifications of mapping refinements which can then be instantiated and thus applied many times.

By analyzing the examples of mapping refinement delivered by the domain expert, the engineer will be able to identify groups requiring the same refinement treatment and to specify the appropriate pattern to apply to each of them. The specification will be declared in such a generic way, then instantiated on the alignment results and the concerned ontologies in order to perform the expected treatments. The patterns are stored and can be reused from one mapping refinement task to another.

3.2 The mapping refinement work-flow

Fig. 2 presents the mapping refinement work-flow implemented in the TaxoMap Framework. First, TaxoMap is performed on two ontologies, a source one and a target one (cf. 1). The alignment results, i.e. the mappings, are stored in a database (cf. 2) and have to be validated by a domain expert or an engineer (cf. 3). When the expert/engineer examines closely the built alignment, he may notice the existence of incorrect mappings or of mappings which are different from what he would have liked. These mappings are grouped by the engineer when they correspond to a similar case. The examples related to a similar case are generalized (cf. 4) and the corresponding pattern is described (cf. 5). The patterns are then applied to the whole mappings database, i.e. to the mappings cited by the expert as examples of mappings having to be refined but also to other ones that the expert has not seen but which are also instances of the patterns (cf. 6). Results of the mapping transformation process have then to be validated (cf. 3). The validation phase helps to check whether a treatment generates undesirable mappings. In case mappings are updated where they should not be, these mappings are a means to clarify the right treatments to be performed (the right patterns to be applied). Thus, the mapping refinement process must be viewed as an iterative validation/correction process needed by the great number of mappings to be examined. The validation, the generalization and the specification of patterns are manual treatments. The mapping transformation based on the use of patterns is automatic.

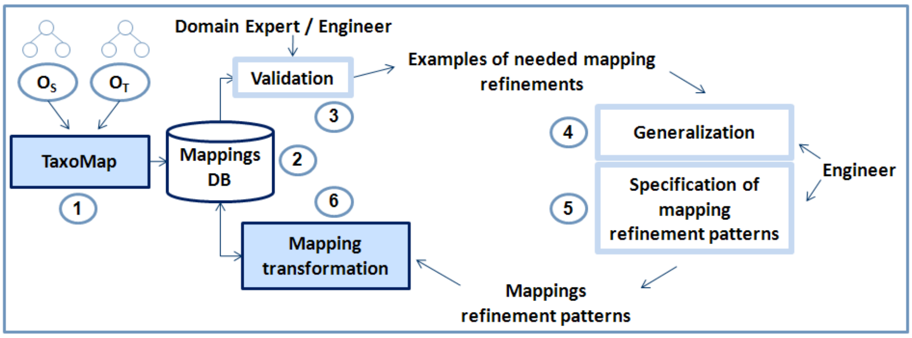


Fig. 2. The mapping refinement work-flow

3.3 MRPL, the Mapping Refinement Pattern Language

The language MRPL is used to specify mapping refinement pattern.This language differ from the one defined in [16]. It includes patterns which test the existence of mappings generated by a given technique. MRPL is defined as follows:

Definition 1 (Vocabulary).
The vocabulary of MRPL contains:

- a set of predicate constants. We distinguish three categories of predicate constants: the predicate constants relating to the type of techniques applied in the identification of a mapping by TaxoMap, the predicate constants expressing structural relations between concepts of a same ontology, the predicate constants expressing terminological relations between labels of concepts.
- a set of individual constants: $\{a, b, c, ...\}$
- a set of variables: $\{x, y, z, ..., \_\}$ where $\_$ is an unnamed variable used to represent parameters which do not need to be precised.
- a set of built-in predicates: $\{Add\_Mapping,\ Delete\_Mapping\}$
- a set of logical symbols: $\{\exists,\ \wedge,\ \neg\}$

MRPL allows the definition of a **context part** which must be satisfied to make the execution of a pattern possible, and of a **solution part** which expresses the process to achieve when the **context part** is satisfied. The **context part** is a logical formula defined as follows.

Definition 2 (Terms).
Variables and constants are terms.

Definition 3 (Syntax).
If α and β are terms and P is a predicate symbol with two places then $P(\alpha, \beta)$ is a formula.
If α, β and γ are terms and P is a predicate symbol with three places then $P(\alpha, \beta, \gamma)$ is a formula.
If ϕ and ψ are formulae then $[\phi \wedge \psi]$ is a formula.
If ϕ is a formula then $[\neg\ \phi]$ is a formula.
If ϕ is a formula and v is a variable then $\exists v \phi$ is a formula.

The **context part** tests (1) the technique used to identify the considered mapping, (2) the structural constraints on mapped elements, for example, the fact that they are related by a subsumption relation to concepts verifying or not some properties, or (3) the terminological constraints, for example, the fact that the labels of a concept are included in the labels of other concepts. These conditions are represented using formulae built from predicate symbols. So, we distinguish three kinds of formula according to the kind of predicate symbols used.

The formulae related to the type of techniques applied in the identification of a mapping by TaxoMap. By testing the existence in the mappings database of a particular relation generated by a given technique, we build formulae that implicitly test the conditions for the application of this technique. For example the formula $isAStrictInclusion(x, y)$ tests the existence of a mapping isA generated between two concepts x and y using the technique t_2. It validates implicitly at the same time all the conditions for the application of t_2, i.e. (1) the concept y is the concept of O_T having the highest similarity value with the concept x, (2) one of the labels of y is included in one of the labels of x, and (3) all the words of the labels of y are classified as *full words* by *TreeTagger*. TaxoMap includes several alignment techniques. Thus, several predicate symbols leading to formulae of that kind are needed. More formally, let:

$R_M = \{isEq, isA, isMoreGnl, isClose\}$, the set of correspondence relations used by TaxoMap,

$T = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7, t_8, t_9\}$, the set of techniques.

T_M, the table storing generated mappings in the form of *4-tuple* (x, y, r, t) where $x \in C_S, y \in C_T, r \in R_M, t \in T$. The pairs of variables (x, y) which can instantiate these formulae will take their values in the set $(x, y) \mid (x, y, r, t) \in T_M$. The predicate symbols necessary for the task of refinement presented in this paper are $isEquivalent$, $isAStrictInclusion$ and $isCloseCommonDescendant$ the semantics of which are the following:

- $isEquivalent(x, y)$ is true iff $\exists(x, y, isEq, t_1) \in T_M$
- $isAStrictInclusion(x, y)$ is true iff $\exists(x, y, isA, t_2) \in T_M$
- $isCloseCommonDescendant(x, y)$ is true iff $\exists(x, y, isClose, t_9) \in T_M$

The formulae expressing structural relations between concepts x and y of the same ontology $O = (C, H)$. Since the aim of TaxoMap is the alignment of taxonomies, the structural relations considered here are subsumption relations. If the approach was used with another alignment tool, other relations could be considered. Note that the instances of variables in these formulae will be constrained, either directly because they instantiate the previous formulae, related to the type of the applied techniques, or indirectly by having to be in relation with other instances.

- $isSubClassOf(x, y, O)$ is true $\Leftrightarrow isA(x, y) \in H$
- $isParentOf(x, y, O)$ is true $\Leftrightarrow isA(y, x) \in H$

The formulae expressing terminological relations between the labels of the concepts:

- $conceptsDifferent(x, y)$ is true $\Leftrightarrow ID(x) \neq ID(y)$ with $ID(x)$ is the identifier of the concept x.
- $appearInLabel(c, y)$ is true $\Leftrightarrow \exists$ a label L_1 of y such as $c \subset L_1$, where c is a string and $y \in C_S \cup C_T$.
- $strictInclusionLabel(x, y)$ is defined as follows:

Algorithm 1 strictInclusionLabel(x,y)

```
Require: $\{x, y\} \in C_S \cup C_T$
for each label $L_1$ of $x$ and each label $L_2$ of $y$ do
    if $L_1 \subseteq FullWords(L_2, L_1)$ then
        return $true$
    end if
end for
```

where $FullWords(L_2, L_1)$ is a function which calculates the common terms to L_1 and L_2 considered as *full words*.

- $extractFromLabel(x, c, y, r)$ is defined as follows:

Algorithm 2 extractFromLabel(x,c,y,r)

```
Require: $\{x, y\} \in C_S \cup C_T$ and $c \in \{$"and", "or"$\}$
for each label $L_1$ of $x$ do
    $SplitLabelPart(L_1, c, Part_1, Part_2)$
    if one label of $y = Part1$ then
        $r = Part_2$, return $true$
    else if one label of $y = Part_2$ then
        $r = Part_1$, return $true$
    else
        return $false$
    end if
end for
```

where $SplitLabelPart(L_1, c, Part_1, Part_2)$ is a function which extracts from the label L_1 two new labels $Part_1$ and $Part_2$, where $Part_1$ and $Part_2$ consist of words that appear respectively before and after c.

- $inclusionInLabel(x, c, y)$ is true $\Leftrightarrow$ $extractFromLabel(x, c, y, \_)$ is true.

A **context part** is associated to a **solution part** which is a set of actions to be performed. This set of actions is modeled by a conjunction of built-in predicates executed in a database. The built-in predicates are defined as follows:

- $Add\_Mapping(x, y, r)$ has the effect of adding a tuple to the table T_M which becomes $T_M \cup \{(x, y, r, t)\}$ where r and t are fixed in the treatment condition by instantiating the predicate corresponding to the type of technique associated with the considered mapping.
- $Delete\_Mapping(x, y, \_)$ has the effect of removing a tuple from the table T_M which becomes $T_M - \{(x, y, \_, \_)\}$.

4 Mapping Refinement Patterns

In this section, we present some mapping refinement patterns designed in the setting of the ANR project, *GéOnto* [5]. At first, TaxoMap performed an alignment between *Topo-Cogit* and *Carto-Cogit*, two taxonomies provided by the *COGIT-IGN* and containing respectively 600 and 495 concepts. 326 mappings have been generated and stored in the mappings database. 25 mappings (precision $92,3\%$) have been deemed as invalid by the domain expert. For other mappings, the expert proposed alternative mappings. We used the TaxoMap Framework to specify the changes to be done through mapping refinement patterns.

Pattern-1: This first pattern is illustrated in Fig. 3. It concerns mappings detected by the technique t_2, connecting by a subsumption relation isA a concept x of the source ontology O_S to a concept y of the target ontology O_T, such as one of the labels of y is included in one of the labels of x. If one of the labels of the concept z that subsumes y in O_T is also included in the label of x, the expert prefers to link x to z, the most general concept of O_T.

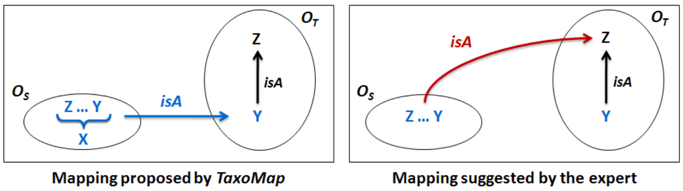
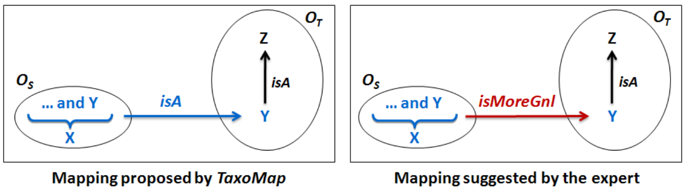

O_T
Z
isA
Y
O_S
Z ... Y
X
isA
Mapping proposed by *TaxoMap*
O_T
isA
Z
isA
Y
O_S
Z ... Y
Mapping suggested by the expert

Fig. 3. Illustration of Pattern-1

Context part of Pattern-1:
$\exists x \exists y \ (isAStrictInclusion(x, y)$
$\wedge \ \exists z \ (isSubClassOf(y, z, O_T) \wedge strictInclusionLabel(z, x)))$
Solution part of Pattern-1:
$Delete\_Mapping(x, y, \_) \wedge Add\_Mapping(x, z, isA)$

The application of this pattern on the example presented in Fig. 1 allows first to select the mapping (id_1, id_2, isA, t_2) where one of the labels of id_1 is “Road and coast trail”, one of the labels of id_2 is “Trail” and such as the formula $isAStrictInclusion$ (id_1, id_2) is satisfied in the mappings database. The variables x and y are instantiated by id_1 and id_2 respectively. The use of the formula $isSubClassOf(id_2, z, O_T)$ based on a structural predicate symbol leads to the instantiation of the variable z by id_3, where one of the labels of id_3 is “Road”, and to the verification of the formula $strictInclusionLabel(id_3, id_1)$. The mapping (id_1, id_2, isA, t_2) is then removed from the mappings database and replaced by the mapping (id_1, id_3, isA, t_2).

Pattern-2: This second pattern concerns also the mappings generated by the technique t_2. If none of the labels of the concept z that subsumes y in O_T is included in the labels of x (see the two last conditions of the pattern) but if instead it contains one of the connectors “and” or “or”, the expert considers that x is not a specialization of y but rather a generalization of it, that we represent by the relation “isMoreGnl” (see Fig. 4). An example of the application of the Pattern-2 is given in the Fig. 5.



Mapping proposed by *TaxoMap*

Mapping suggested by the expert

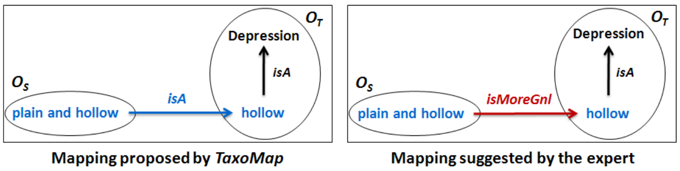
Fig. 4. Illustration of Pattern-2

Context part of Pattern-2:

$\exists x \exists y\ (isAStrictInclusion(x, y) \wedge inclusionInLabel(x, \text{``}and\text{''}, y)$
$\wedge\ \exists z\ (isSubClassOf(y, z, O_T) \wedge \neg strictInclusionLabel(z, x)))$

Solution part of Pattern-2:

$Delete\_Mapping(x, y, \_) \wedge Add\_Mapping(x, y, isMoreGnl)$



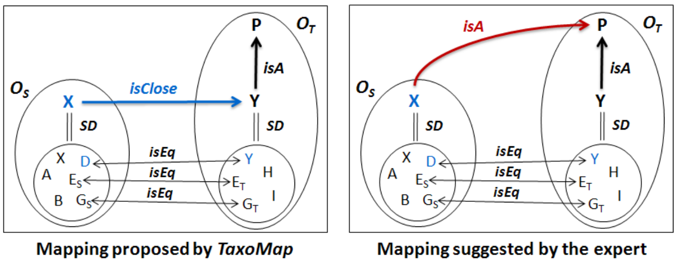
Mapping proposed by *TaxoMap*

Mapping suggested by the expert

Fig. 5. Example of the application of the Pattern-2

Pattern-3: Let the set $SD(c, O)$ be composed of c and of all its sub-concepts in O. The measure $M_{SD}(c_1, O_1, c_2, O_2)$ is defined as the ratio between the number of equivalence relations verified in the mapping table between concepts in $SD(c_1, O_1)$ and in $SD(c_2, O_2)$ and the total number of concepts belonging to the union of these two sets. The technique t_9 connects by a relation of proximity *isClose*, a concept x of O_S to a concept y of O_T, if y is the concept in O_T which has at least two descendants in common with x and which maximizes the M_{SD} for x.

If there is a concept $d \in O_S$ such that $isEquivalent(d, c_t)$ and $d \in SD(c_s, O_S)$, the expert prefers to connect x to the father of y in O_T by a subsumption relation. An illustration is given in Fig. 6.



Mapping proposed by *TaxoMap*

Mapping suggested by the expert

Fig. 6. Illustration of Pattern-3

Context part of Pattern-3:

$\exists x \exists y\ (isCloseCommonDescendant(x, y) \wedge \exists d\ isEquivalent(d, y)$

$\wedge\ isSubClassOf(d, x, O_S) \wedge \exists p\ isParentOf(p, y, O_T))$

Solution part of Pattern-3:

$Delete\_Mapping(x, y, \_) \wedge Add\_Mapping(x, p, isA)$

5 Experiments in the context of the *GéOnto* project

This section illustrates the mapping refinement work-flow presented in Section 3.2, the interactions between the expert, the engineer and our tool leading to the design of refinement patterns. The experimentation that we describe here is that guiding the expert and the engineer to refine mappings generated by the technique t_2 which constructs an isA mapping between c_s and c_{tmax} if (1) the concept c_{tmax} is the concept of O_T having the highest similarity value with the concept c_s of O_S, (2) one of the labels of c_{tmax} is included in one of the labels of c_s, (3) all the words of the included label are full words. $isAStrictInclusion$ is the corresponding predicate. We chose this experimentation because the technique t_2 generates an important number of mappings. The 3 iterations described below are needed to specify the right pattern operating the right modifications. Let us note that mapping produced by TaxoMap are presented technique by technique. This allows to easily validate mappings generated by a given technique.

Iteration 1:

The evaluation of the mappings produced by the technique t_2 leads the expert to identify 3 mappings as examples of what needs to be modified: "plain and hollow *isA* hollow" should become "plain and hollow *isMoreGnl* hollow", "wood and forest *isA* forest" should become "wood and forest *isMoreGnl* forest", "road or street *isA* street" should become "road or street *isMoreGnl* street".

These 3 examples are generalized by the engineer as follows: in the context of this alignment technique, when the label of the concept c_s in O_S contains a connector "and/or", c_s is not a specialization of c_{tmax} but rather a more general concept. This change is implemented in a pattern as follows:

Context part:

$\exists x \exists y\ (isAStrictInclusion(x, y) \wedge appearInLabel(\text{“}and\text{”}, x)$
$\wedge\ \exists z\ (isSubClassOf(y, z, OT) \wedge \neg strictInclusionLabel(z, x)))$

Solution part:

$Delete\_Mapping(x, y, \_) \wedge Add\_Mapping(x, y, isMoreGnl)$

The application of this pattern to the whole mappings database leads to the modification of 20 mappings. 3 of them are the examples proposed by the expert but 17 additional mappings have also been updated. For example, "rocks and sand *isMoreGnl* rock", "local or private museum *isMoreGnl* museum", "campanile and not adjacent belfry *isMoreGnl* belfry". Their evaluation is necessary.

That leads to a new cycle of mapping refinement.

Iteration 2:

For 5 additional mappings, the modifications are consistent with what the expert asks. The label of c_s contains a conjunction. So, c_s is considered as more general than c_{tmax} whose label is included in the label of c_s. For example "rocks and sand *isMoreGnl* rock". But it reveals also undesirable modifications, especially when the part of c_s containing the label of c_{tmax} denotes a more specific concept than c_{tmax} (for example the part of c_s "private museum" containing "museum" which is the label of c_{tmax}). In this case, c_S must not be considered as more general than c_{tmax}. Consequently, the only presence of a connector "and/or" is not enough to guarantee that c_s is more general than c_{tmax}. It is necessary to check that the connector separates effectively the exact label of c_{tmax} and something else (which we will called the remaining part), in the form "P_1 and/or P_2" where the label of c_{tmax} is exactly P_1 or P_2.

This leads the engineer to modify the previous pattern by using instead of $appearInLabel(\text{“}and\text{”}, x)$, the formula $inclusionInLabel(x, c, y)$, which allows to check if one of the two parts connected by the connector c is exactly the label of y: *InclusionInLabel("water treatment and pumping station", and, "pumping station")* is true, while *InclusionInLabel("local or private museum", or, "museum")* and *InclusionInLabel("campanile and not adjacent belfry", and, "belfry")* are false. The pattern becomes:

Context part:

$\exists x \exists y\ (isAStrictInclusion(x, y) \wedge inclusionInLabel(x, \text{“}and\text{”}, y)$
$\wedge\ \exists z\ (isSubClassOf(y, z) \wedge \neg strictInclusionLabel(z, x)))$

Solution part:

$Delete\_Mapping(x, y, \_) \wedge Add\_Mapping(x, y, isMoreGnl)$

The application of this new pattern to the original whole mappings database leads to the modification of 8 mappings. 3 of them are the examples proposed by the expert. Only 5 additional mappings (among 17 modified by the pattern in iteration 1) have been updated. This leads to a new iteration where the expert has to evaluate these 5 additional mappings and 12 mappings modified in iteration 1 but not in iteration 2, which are considered as counterexamples.

Iteration 3:

In this phase, the expert validates the modifications of the 5 additional mappings, as well as the preservation of 10 of the 12 mappings presented as counterexamples. Two mappings were not updated by the pattern in its final version but the expert would have wanted them to be modified: "campanile and not adjacent belfry *isA* belfry" "Highway or lane road with divided ways *isA* road with divided ways".

The analysis of these two counterexamples shows that in both cases, the label of c_s is in the form "P_1 and/or P_2" with the label of c_{tmax} included in

P_2 without being exactly equivalent. However the string P_1, which we call the remaining part, is the label of a domain concept ("campanile" in the first example, "highway" in the second). The concept identification would be simple to perform automatically in the second example because "highway" is a label of a concept in O_T. It is more difficult in the first example, since "campanile" is not a label of any concept, either in O_T or in O_S. So only one of the two new desired changes can be performed automatically by introducing an additionnal pattern. The pattern previously defined must not be modified. The expert has validated its results. The new pattern addresses a new case identified by the expert during iteration 3. Note that the results are unchanged regardless of the order of applying these 2 patterns (the pattern previously defined and the new one).

The whole experiment in the topographic field led to specify 6 refinement patterns related to 4 alignment techniques of TaxoMap. 24 mappings have been modified. 23 satisfy the wishes of the expert. One mapping is incorrect. Only two modifications have not been considered.

6 Related Works

Many alignment tools existing today generate good results in certain cases and not so good results in other cases. This observation should direct research to treat several problems [18] such as: the choice of the most adapted tool, the combination of the alignment techniques and the problem of the regulation of the parameters (thresholds, coefficient of formulas, etc.) used in the alignment tools. Our works are issued from the same observation but have been developed in a different direction, the alignment refinement, and subsequently the assistance to the specification of treatments based on mappings.

The closest work we know is the COMA++ system [2]. It aims to build powerful alignment tools by the combination of existing matchers then to refine the obtained alignment results considered as preliminary. The refinement process is here totally automatic. The COMA++ alignment process is re-applied on groups of elements whose proximity has been established by a first treatment applied to ontologies. The refinement of the alignment can also be seen as an adaptation of the alignment solutions to the context of an application. Thus, the system eTunes [11] adapts an alignment by looking automatically to the most adapted values for the parameters of the alignment system. Other works deal with alignment refinement or alignment transformation which are close but not similar activities. In [16] and [15], correspondences patterns are used to assist the design of precise and complex ontology alignments when parts of both ontologies represent the same conceptualizations but modeled in two different ways. This approach can be seen as a way to refine one-to-one correspondences which can then be used to transform an ontology into another as in [16]. Other works propose services to transform alignments. The Alignment API [3] generates transformations which are implementations for rendering the alignments, but the alignments are not modified.

Regarding our environment, another related work is PROMPT-Suite integrating the ontology merging tool IPROMPT [13], the alignment tool Anchor-PROMPT, versioning, comparison, translation functionalities. All these tools are interactive and semi-automatic. For example, in the fusion process the system makes suggestions. The expert can hold one of them or specify an operation to perform. The system then executes the operation, calculates the resulting changes, makes other suggestions and detects any inconsistencies.

All systems combining several alignment systems are very modular. The possibility of defining the strategy of combination makes them adaptable to a new field of application. This modularity and adaptability are strong points which also characterize our approach. The treatments which can be specified in the TaxoMap Framework are indeed modular and conceived to integrate the very particular characteristics of the treated ontologies. It goes beyond the possibilities of the tools previously mentioned. However, the TaxoMap Framework differs from existing tools such COMA++, eTunes or PROMPT-Suite by considering that the performance of an alignment tool implementing general alignment algorithms is necessarily limited (even if the values of parameters are optimal). Some improvements can be obtained only after taking into account the particularities of the aligned ontology which involves various improvements depending on the ontologies. Specifying such improvements needs to be familiar with the aligned ontologies. So this process cannot be automatic. Only an expert of the domain is able to suggest them. As in PROMPT-Suite, we offer an interactive environment to help an expert assisted by an engineer to carry out this task, but we do it differently. We allow the definition of particular generic treatments able to take into account specific conventions used in the ontologies. In PROMPT-Suite, this is not possible. The treatments are all pre-defined.

7 Conclusion and Future Work

In this paper, we have presented an environment for the specification of treatments based on alignment results generated by TaxoMap. We presented the context of this work, the approach, the mapping refinement work-flow and the Mapping Refinement Pattern Language MRPL. We described the use of our mapping refinement approach applied in the topographic field. This approach has been implemented in the TaxoMap Framework. We illustrated its use and the usefulness of the approach through experiments made in the setting of the ANR project *GéOnto*.

The engineer can select all the elements of the vocabulary of MRPL through an appropriate GUI accessible at the following Web address [19]. Note that the approach is based on the use of TaxoMap as an alignment tool, but it could be based on another tool. If the predicate symbols associated with this other tool have been defined, the specification of refinement treatments is simplified. If these predicates have not been defined, it will be necessary to further specify the conditions that must be satisfied in the context part of the pattern. Anyway, the method is usable for any alignment tool.

The TaxoMap Framework has also been designed to allow the specification of other treatments such as merging, restructuring and enriching ontologies based on alignment results. Future work will be devoted to the design and the implementation of the modules corresponding to these additional functionalities. It will be devoted also to the extension of the approach for refining the mappings between ontologies that have a more richer axiomatisation.

References

1. J. Euzenat, A. Ferrara, L. Hollink, A. Isaac, C. Joslyn, V. Malaisé, C. Meilicke, A. Nikolov, J. Pane, M. Sabou, F. Scharffe, P. Shvaiko, V. Spiliopoulos, H. Stuckenschmidt, O. Sváb-Zamazal, V. Svátek, C. Trojahn dos Santos, G. Vouros, S. Wang.: Results of the Ontology Alignment Evaluation Initiative 2009. Proc. 4th ISWC workshop on ontology matching (OM), Chantilly (VA US), pp. 73-126 (2009).
2. H.-H. Do, E. Rahm.: Matching large schemas: Approaches and Evaluation. Information Systems 32, pp. 857-885 (2007).
3. J. Euzenat.: An API for ontology alignment, ISWC, Hiroshima (JP), LNCS 3298:698-712 (2004).
4. J. Euzenat, P. Shvaiko.: Ontology Matching. Springer-Verlag, Heidelberg (DE) (2007).
5. *GéOnto* Project `http://geonto.lri.fr`
6. F. Hamdi, B. Safar, N. Niraula, C. Reynaud.: TaxoMap in the OAEI 2009 alignment contest. In ISWC Workshop on Ontology Matching, Westfields Conference center near Wahshington, DC, pp. 230-237 (2009).
7. F. Hamdi, H. Zargayouna, B. Safar, C. Reynaud.: TaxoMap in the OAEI 2008 alignment contest. In the 3rd ISWC Workshop on Ontology Matching, Karlsruhe (DE), pp. 206-213 (2008).
8. M. Kamel, N. Aussenac-Gilles.: Ontology Learning by Analysing XML Document Structure and Content, Knowledge Engineering and Ontology Development (KEOD), Madère, Portugal, pp. 159-165 (2009).
9. Ontology Alignment Evaluation Initiative.: `http://oaei.ontologymatching.org/`
10. E. Kergosien, M. Kamel, C. Sallaberry, M.-N. Bessagnet, N. Aussenac, M. Gaio.: Construction automatique d'ontologie et enrichissement à partir de ressources externes. In JFO proceedings, Poitiers, pp. 1-10 (2009).
11. Y. Lee, M. Sayyadian, A. Doan, A.S. Rosenthal.: eTuner: tuning schema matching software using synthetic scenarios. The VLDB Journal 16:97-122 (2007).
12. D. Lin.: An Information-Theoretic definition of Similarity. In proc. of ICML-98, Madison, pp. 296-304 (1998).
13. N. F. Noy, M.A. Musen.: The PROMPT Suite: Interactive Tools For Ontology Merging And Mapping. IJHCS, 59(6), pp. 983-1024 (2003).
14. C. Reynaud, B. Safar.: Techniques structurelles d'alignement pour portails Web. Revue RNTI W-3, Fouille du Web, Eds. Cépaduès, pp. 57-76 (2007).
15. D. Ritze, C. Meilicke, O. Svab-Zamazal, H. Stuckenschmidt.: A pattern-based Ontology Matching Approach for detecting Complex Correspondences, In ISWC Ontology Matching Workshop, Westfields Conference center near Washington, DC, pp. 25-36 (2009).
16. F. Scharffe, J. Euzenat, D. Fensel.: Towards Design patterns for Ontology Alignment, SAC, pp. 2321-2325 (2008).

17. H. Schmid.: Probabilistic Part-of-Speech Tagging Using Decision Trees. In Int. Conf. on New Methods in Language Processing (1994).
18. P. Shvaiko, J. Euzenat.: Ten Challenges for Ontology Matching. In proc. of the 7th International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE), Monterey (MX), pp. 1163-1181 (2008).
19. *TaxoMap FrameWork.*: `http://www.lri.fr/~hamdi/TaxoMap/TaxoMap.html`