Towards Correspondence Patterns for Ontology Mediation*

François Scharffe, Ying Ding, and Dieter Fensel

University of Innsbruck, Austria
firstname.lastname@uibk.ac.at

Abstract. We introduce in this paper correspondence patterns as a tool to design ontology alignments. Based on existing research on patterns in the fields of software and ontology engineering, we define a pattern template and use it to develop a correspondence patterns library. This library is published in RDF following a structured vocabulary. It is meant to be used in ontology alignment systems, in order to support the user or improve matching algorithms to refine ontology alignments.

1 Introduction

The semantic web contains many ontologies, and is expected to contain more and more as it will develop. When two ontologies overlap, they can be linked together in order to enable exchange of their underlying knowledge. An *alignment* between two ontologies specifies a set of *correspondences*, and each correspondence models a bridge between a set of ontologies entities. Designing ontology alignments is a tedious task. There are many ongoing efforts to develop tools such as graphical user interfaces and matching algorithms, in order to make it easier.

Most ontology alignment systems [1] are limited to detect simple equivalence or subsumption correspondences between single entities, and research concentrates on improving their quality on various datasets more than on finding more complex matches[1]. This can be explained as the problem of detecting complex matches is not a trivial one, maybe also because no standard semantic-web language is expressive enough to represent such correspondences.

However, simple correspondences are often not sufficient to correctly represent the relation between the aligned entities. The two ontologies in the following example deal with wines. In the "Wine Ontology"[2], which main class is $Wine$, the class $BordeauxWine$ represents instances of a popular french wine grown up in the region around Bordeaux. In the "Ontologie du Vin"[3] a similar wine

* This work was partially funded by the European Commission under the Knowledge Web NoE (FP6-507482) and SEKT project

[1] See the Ontology Alignment Evaluation Initiative 2007 evaluation criteria. http://oaei.ontologymatching.org/2007/

[2] http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine#

[3] http://www.scharffe.fr/ontologies/OntologyDuVin.wsml

is expressed as an instance of Vin with an attribute $terroir$[4] indicating the wine provenance. Classical systems are able to detect the two correspondences $Wine \equiv Vin$ stating equivalence between two wines and $BordeauxWine \subseteq Vin$ stating that $BordeauxWine$ is a narrower concept than Vin . In this case, a more precise correspondence would be $BordeauxWine \equiv Vin \wedge terroir = Bordeaux$, restricting the scope of Vin to only those instances having $Bordeaux$ as value of the $terroir$ attribute.

Going from the initial two basic correspondences to the refined one involving a condition would be easier using a pattern showing the structure of the correspondence. Inspired from design patterns in software engineering, this paper introduces correspondence patterns as helpers that facilitate the ontology alignment process. They improve graphical tools by assisting the user when creating complex correspondences, and we believe they are a first step towards matching algorithms able to detect complex matches.

2 Correspondence Pattern Template

A pattern template provides a standard way to represent patterns. Following [2], we introduce a correspondence pattern template divided in two parts that roughly correspond to the two lowest levels of Blomqvist pattern classification [3]. The first part, core of the correspondence pattern, defines the pattern using classical elements from design patterns literature. The second part presents a grounding for that pattern in a knowledge representation formalism. Many groundings can be defined given a pattern.

A Correspondence Pattern represents a correspondence between two ontologies. The core part of the pattern template contains common pattern elements found in the literature. It follows the high level classification of the GoF [4].

The grounding part represents the grounding of the pattern into a knowledge representation formalism used by a mediator executing pattern instances.

The pattern template defined above give all the elements to describe a correspondence pattern. We give next an example of pattern based on this template. We then see in Section 3 how correspondence patterns can be encoded in a machine readable format in order to facilitate their use in semantic web applications.

The following pattern can be used to model the correspondence presented in Section 1.

Name: Class By Attribute Correspondence
Also Known As: classByAttributeCorrespondence
classByAttributeMapping
Problem:

[4] Terroir is a French word for a particular agricultural region

A class in one ontology is related to a class in the other ontology. The scope of the class of the first ontology is restricted to only those instances having a particular value for a a given attribute.

Context: This pattern is used in case two classes have a similar but not completely overlapping extension and the intent of the first class is expressed by a particular attribute value in the target class.

Solution:
Solution description:
This pattern establishes a correspondence between a class / attribute / attribute value combination in one ontology and a class in another.

Syntax: Here comes a correspondence described using the Alignment Ontology

Example: Relate a Human to a Blue-Eyed-Person by restricting those instances of Human whose eye-color attribute is equal to "blue"

Related Patterns: Equivalent Class Correspondence, Subclass Correspondence

Degenerated Pattern: Subclass Correspondence

The patterns library defined Section 3.2 provide generic entities used to be replaced. A grounding of the pattern is given in the following in SPARQL.

Name of the target language/system SPARQL

Applicability Applicable

Purpose Instance Transformation

Example Grounding

```
CONSTRUCT { ?X rdf:type target:BlueEyedPerson }
WHERE { ?X rdf:type source:Human.
        ?X source:eyeColor xsd:String^^"Blue" }
```

The wine example Section 1 can be built by instantiating this pattern. We will show in the next section how is it done in our implementation.

The syntax element of the solution shows a correspondence described in RDF, as an instance of the Alignment Ontology [5]. Purpose of this ontology is the abstract representation of ontology alignments. Section 3 briefly introduces this ontology, and presents its extension: a correspondence patterns library.

3 Correspondence Patterns on the Semantic Web

Now that correspondence patterns are given a template, we need to consider how to implement and store them. Mediation systems need an organized library of patterns in order to retrieve them efficiently. A hierarchical organization seems preferable, following patterns specialization: from general purpose patterns such

as between any two classes to very specific patterns using conditions and transformations of attributes values.

Correspondence patterns are used to align ontologies, which are themselves used to reference and reason about RDF resources on the semantic web. On the other way around, publishing patterns and correspondences as ontology instances presents many advantages. The class hierarchy automatically organizes patterns according to their specificity. The degenerated version of a pattern is thus obtained by taking its parent in the class hierarchy. In addition, patterns assert additional facts about correspondences that can be used as background information in order to construct new alignments. Let us consider the wine example shown in Section 1 solved using the `ClassByAttribute` pattern. A system trying to align $BordeauxWine$ with a third ontology can then try applying the same pattern as a similar structure is likely to appear.

Based on the template given in Section 2 we present in this section a library organizing common ontology correspondences patterns. This library is provided as an extension of the Alignment Ontology [5].

3.1 The Alignment Ontology

The Alignment Ontology is an OWL-DL ontology that models ontology alignments as sets of correspondences between ontological entities. It results from efforts [6, 7] to create an ontology alignment format, abstract from the ontology language, that could serve as an interchange format between matching algorithms, mediators and graphical user interfaces.

The ontology is available at http://www.omwg.org/TR/d7/.

3.2 A Correspondence Patterns Library

The patterns library extends the Alignment Ontology under the `Cell` class. The library contains 35 correspondence patterns at the time of writing[5]. Properties corresponding to the pattern template elements are modeled as OWL annotation properties. A pattern in the library is modeled as a class with restrictions on its possible instantiations and an example instance is given for each pattern. In other words, the deepest one goes into the pattern hierarchy, the more specific are the patterns, making retrieval of patterns easier.

Besides patterns themselves, the library provides a set of generic ontological entities used to compose patterns. These entities serve as placeholders to be filled with concrete entity names when using the pattern to make a concrete correspondence. They provide a reference structure for modeling entities in the same way patterns provide a reference structure for modeling correspondences. For example, the `ClassConditionInstance` models a class with a condition restricting its scope.

The pattern library is available at http://www.omwg.org/TR/d7/.

[5] This figure does not take into account all possible value transformations patterns.

We ground correspondences using transformations of the Alignment Ontology instances into the target formalism. The Alignment API [6] and the Mapping API [7] propose groundings into various semantic-web languages such as OWL, SWRL, WSML and SKOS. We are currently working on extending the implementation in order to provide SPARQL++ [8] grounding.

4 Conclusion

We introduced in this paper ontology correspondences patterns as helper to model ontology alignments. We defined a template based on the literature on design patterns and gave a library of elementary correspondence patterns. Patterns instances are described using the Alignment Ontology, a formalism to represent ontology alignments abstract from the underlying knowledge representation formalism. In order to be executed, pattern instances are grounded to the desired formalism. We are constantly refining the library given experience in research projects. We currently investigate on using correspondence patterns in order to develop a patterns based ontology matching algorithm.

5 Acknowledgements

We would like to thanks Jos de Bruijn for useful discussions about correspondence patterns.

References

1. Shvaiko, P., Euzenat, J.: A survey of schema-based matching approaches. (2005) 146–171
2. Staab, S., Erdmann, M., Maedche, A.: Engineering ontologies using semantic patterns. In O'Leary, A.P..D., ed.: Proceedings of the IJCAI-01 Workshop on E-Business & the Intelligent Web, Seattle, WA, USA, August 5, 2001. (2001)
3. Blomqvist, E., Sandkuhl, K.: Patterns in ontology engineering: Classification of ontology patterns. In: ICEIS (3). (2005) 413–416
4. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns. Addison-Wesley Pub. (1995)
5. Scharffe, F.: Omwg d7: Ontology mapping language. http://www.omwg.org/TR/d7/ (2007)
6. Euzenat, J.: An API for ontology alignment. In: Proc. 3rd international semantic web conference, Hiroshima (JP). (2004) 698–712
7. Scharffe, F., de Bruijn, J.: A language to specify mappings between ontologies. In: Proc. of the Internet Based Systems IEEE Conference (SITIS05). (2005)
8. Polleres, A., Scharffe, F., Schindlauer, R.: Sparql++ for mapping between rdf vocabularies. In: ODBASE, On the Move Federated Conferences (to appear). (2007)