SPARQL-based Detection of Antipatterns in OWL Ontologies

Oscar Corcho, Catherine Roussey, Ondrej Zamazal, Francois Scharffe

SPARQL-based Detection of Antipatterns in OWL Ontologies

[Poster]

Oscar Corcho
Ontology Engineering Group
Departamento de Inteligencia Artificial
Universidad Politécnica de Madrid, Spain
ocorcho@fi.upm.es

Catherine Roussey
Cemagref
Aubiére, France
Université de Lyon, CNRS
Université Lyon 1, LIRIS
UMR5205
Villeurbanne, France
catherine.roussey@cemagref.fr

Ondrej Svab-Zamazal
Knowledge Engineering Group
University of Economics
Prague, Czech Republic
ondrej.zamazal@vse.cz

François Scharffe
EXMO team, INRIA Grenoble
Rhône-Alpes
Saint-Ismier, France
francois.scharffe@inria.fr

ABSTRACT

Ontology antipatterns are structures that reflect ontology modeling problems because they lead to inconsistencies or to bad reasoning performance. Based on a collection of antipatterns coming from our experience in ontology engineering projects and bad modeling practices found in the literature, we propose to represent them as SPARQL queries and conduct an experiment to detect them in an ontology corpus obtained from the Watson ontology search portal.

Categories and Subject Descriptors

I.2.4 [**Computing Methodologies**]: Artificial Intelligence-Knowledge Representation Formalisms and Methods

General Terms

Experimentation

Keywords

Antipattern detection, SPARQL query, OWL ontology

1. INTRODUCTION

The focus of this poster is to understand, in the context of a larger experiment, how often antipatterns appear in existing publicly available ontologies and how these antipatterns are normally clustered in them, so as to understand better their characteristics and their importance in terms of addressing them later in the context of ontology debugging tools and tasks inside methodologies and methods. To achieve this objective, we have selected a set of ontologies from those available in the Watson semantic search engine, we have transformed the selected antipatterns from our catalogue into sets of SPARQL queries or have transformed the original ontologies into a form where simpler SPARQL queries can be run to detect antipatterns, and have run those queries against the selected ontologies or their transformations, switching on and off inferences, so as to see whether there is any impact in the detection and posterior repair of the ontologies.

EKAW '2010 Lisbon, Portugal

2. A CATALOGUE OF ANTIPATTERNS

In [1] we identified a set of patterns commonly used by domain experts in their implementation of OWL ontologies, and which normally resulted in unsatisfiable classes or modelling errors. These patterns are categorized into three groups:

- Detectable Logical AntiPatterns (DLAP). They represent errors that DL reasoners and debugging tools normally detect. Our experiment is focused on the detection of three examples of DLAP antipattern: AndIsOr (AIO), OnlynessIsLoneliness (OIL), UniversalExistence (UE).
- Cognitive Logical AntiPatterns (CLAP). They represent possible modelling errors that may be due to a misunderstanding of the logical consequences of the used expression. We try to detect two of them: SynonymOrEquivalence (SOE), SumOfSom (SOS).
- Guidelines (G). They represent complex expressions used in an ontology component definition that are correct from the logical and cognitive points of view, but for which the ontology developer could have used other simpler alternatives or more accurate ones for encoding the same knowledge. We choose to detect only one

antipattern of this type namely DisjointnessOfComplement (DOC).

3. ANTIPATTERN DETECTION METHODS

We have elaborated four different strategies in order to detect antipatterns in OWL ontologies by means of SPARQL queries, based on the usage of the PatOMat ontology pattern detection tool[1]. This tool is part of the PatOMat enables the processing of a set of SPARQL queries over a set of ontologies, producing a report in terms of numbers of patterns detected (queries) and details for each ontology.

Transforming antipatterns into SPARQL queries is not a trivial task. In some cases several SPARQL queries are needed to represent all the possible versions of an antipattern, and some of these transformations are difficult to generate.

Now we will show the different approaches that we have followed in order to detect antipatterns in the collection of ontologies that we have selected.

1. SP: Use of SPARQL Queries over Asserted OWL Ontology Axioms.

   In this approach, we take into account that SPARQL query engines per se do not consider inferences that can be done with OWL ontologies. However, our assumption is that there will be cases where this is the only solution that can be applied due to the fact that in some cases inconsistent ontologies can lead to difficulties in the detection of antipatterns due to some wrong inferences being done across the whole hierarchy. Hence we write in general a large number of queries for each antipattern, so that we embed in the syntax of those SPARQL queries some of the inferences that should be done in the OWL ontologies, which could be done with a reasoner.

2. SP+R: Use of SPARQL Queries over Materialised Inferences on OWL Ontologies.

   When it is possible to use a reasoner, we materialise all the inferences that can be done by an OWL reasoner on the selected ontologies and then run SPARQL queries over the resulting ontologies.

3. SP_Trans: Use of SPARQL Queries over Partially Transformed OWL Ontologies using SPARQL queries and no inference.

   Due to the uncertainty about some results from reasoners, to the complexity of creating a large number of SPARQL queries for an antipattern, and finally to the fact that different ontology developers may have different *implementation styles*, what leads to very different alternative implementations of the same sets of axioms, and makes it difficult to find all the possibilities in which queries should be generated in order to identify antipatterns, we propose to follow a two step process where we harmonise the implementation style of the ontology using transformations before proceeding to execute the queries.

   In this strategy, we have explored the behaviour of the SPARQL query checking method on the asserted ontology after transformation.

4. SP_Trans+R: Apply transformations on the original ontologies in order to harmonize the axioms, and search in the materialisation of these harmonised ontologies.

   In this strategy, we transformed the ontologies in order to harmonise the implementation style. Then, we have explored the behaviour of the SPARQL query checking method on the materialised ontology (also after transformation).

[1]The release used for this poster is at: `http://eso.vse.cz/~svabo/patomat/detectionTool.zip`

4. EXPERIMENTS

We have used the Watson API to retrieve publicly available ontologies that we could run experiments with, and we have always accessed these ontologies using the Watson cache. We searched for ontologies satisfying the following constraints: they should be implemented in OWL, they should have at least five classes, they should be classified as inconsistent by Pellet reasoner. So the corpus is finally composed of 66 incoherent ontologies.

The table 1 shows a summary of the total numbers of antipatterns detected on the selected ontologies using each of the methods that have been proposed.

method of detection	SOE	DOC	AIO	OIL	UE	SOS
SP	29	15	67	127	84	246
SP+R	24382	3268	12	0	1	12
SP_Trans			52	2	3	40
SP_Trans+R				0	1	40

Table 1: Total number of patterns detected

The results analysis shows that although antipatterns in the collection occur rarely, they tend to appear many times in the same ontology, reflecting a bad modeling practice.

5. CONCLUSION

In this poster, we present how antipatterns can be detected using different methods and we have tried to show which is the best detection process to be used for each type of antipattern. In many cases, these antipattern detection tools are very sensitive to the *implementation style* of the ontology developer.

6. ACKNOWLEDGMENTS

This work has been partially funded by CSF grant no. P202/10/1825, GeoBuddies, and sponsored by the European Commission under the grant number STSM-C21-04241.

7. REFERENCES

[1] O. Corcho, C. Roussey, L. M. Vilches Blázquez, and I. Pérez. Pattern-based OWL Ontology Debugging Guidelines. In *Workshop on Ontology Patterns (WOP 2009), collocated with the 8th International Semantic Web Conference (ISWC-2009).*, CEUR Workshop proceedings, pages 68–82, Oct. 2009.