Negative Property Assertion Pattern (NPAs)

http://
ontologydesignpatterns.org/wiki/Submissions:NegativePropertyAssertions

Olaf Noppens

Inst. of Artificial Intelligence
Ulm University
Germany
olaf.noppens@uni-ulm.de

1 Introduction

The basic building blocks of Description Logics (DLs) in general are concept and role constructors (e. g. , intersection, union, nominals, negation etc.), terminological as well as individual axioms. Combining axioms, meaningful statements can be expressed. Additional axiom constructors (so-called syntactical sugar) has been introduced in most ontology language (such as OWL and OWL 2 [1]) in order to simplify ontology modeling, understanding and maintenance. These syntactical sugar axioms can be reduced to basic axioms. A simple example is disjointness of concepts that can be reduced to GCI axioms. However, even experts have often difficulties to understand the reduced forms in comparison to syntactical sugar forms. From the modeling perspective, patterns can support the user in modeling logical meaning that is not directly supported in an ontology language. For instance, OWL 2 provides a syntactical form for modeling *negative property assertions* whereas using the predecessor OWL 1 one have to model it by a more or less complicated inclusion axiom (or, alternatively, disjoint axiom).

The motivation of this pattern is to model *negative property assertions* (NPAs) in ontology languages such as OWL 1 that do not provide a special construct for it. It is worth mentioning that not all knowledge base systems can be migrated to OWL 2 for several reasons. On the other hand, NPAs modeled according to this pattern can be migrated to OWL 2 using the newly introduced constructor. A negative property assertion as defined in the upcoming OWL 2 states that a given individual i is *never* connected to a given individual j by a given property expression P. In other words, asserting that i is connected to j by P results in an inconsistent ontology. In this sense this assertion can be considered as a constraint that should not be violated. In contrast, considering an ontology where it cannot be inferred that i is connected to j by p does not necessarily mean that there cannot be such a connection – in fact, it is merely not modeled.

2 Pattern

In this section we describe the Negative Property Assertion Pattern in detail.

2.1 Problem

Prior to OWL 2 it is difficult to model *negative property assertions* and, if they are contained in an ontology, difficult to understand because OWL 1 does not provide a specialized constructor for it. In addition, with help of this pattern one can also migrate OWL 1 ontologies containing axioms describing NPAs into the NPA constructor of OWL 2.

Intent This patterns allows to express NPAs in ontologies written in an ontology language such as OWL 1 that does not allow NPA directly. The pattern can also be understood as a transformation rule for transforming axioms modeling NPAs into the direct NPA axiom constructor of OWL 2.

2.2 Solution

A negative property assertion that states that an individual $Individual_1$ is not connected to an individual $Individual_2$ by a property *property* can be modeled as the following inclusion axiom: $\{Individual_1\} \sqsubseteq \neg\ (\exists property\ \{Individual_2\})$ [1]. Here we are using the German DL syntax where $\{x\}$ denotes an enumeration class with a single individual x (aka. 'one-of'). A graphical representation of this pattern is given in Figure 1.

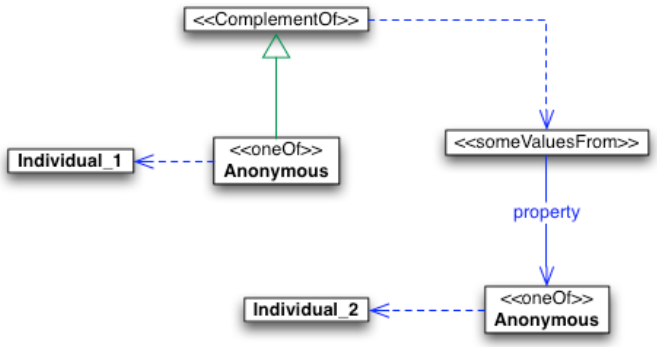

Fig. 1. Graphical representation of the pattern using the UML-based notation proposed in [2].

In the following we proof the equivalence between the axiom produced by the pattern and NPA as defined in OWL 2.

[1] Note that this can also be modeled with help of a disjointness axiom. However, not all languages provide a special disjointness constructor.

Proof Let C and D be concepts. Then C and D are disjoint if, and only if, C is subsumed by the complement of D, i. e. , $(C \sqsubseteq \neg D)$. The equivalence is correct because of the duality of disjointness, equivalence, and unsatisfiability: C is subsumed by D if, and only if, $C \sqcup \neg D$ is unsatisfiable, and the intersection of C and D is unsatisfiable if, and only if, C and D are disjoint.
One also reminds that the extension of the concept $\exists\, prop.C$ is the set of individuals i which are connected to an individual j that is in the extension of the concept C, by the property *prop*.
Let $NPA\,(p\,a\,b)$ be a negative property assertion axiom, i. e. , the individual a is not related to b by the property p. Then the extension of $\exists\, p.\,\{b\}$ which contains all individuals that are connected to b by p must not contain a. This is true, if, and only if, $\{a\}$ is disjoint to $\exists\, p.\,\{b\}$.

Consequences Applying this pattern to an ontology will add the logical meaning of a NPA to the ontology. There are not any restrictions or limitations of the solution besides that nominals and unrestricted existential quantification must be supported in the target ontology language.

2.3 Example

Consider a social network containing facts about people and their relationships. Let `Adam` and `Eve` be two persons and `like` a property ('A likes B'). Furthermore we know that `Adam` does not like `Eve` but we have no dislike relationship. Moreover, our language (such as OWL 1) does not have any NPA axiom constructor.

The sample ontology is interpreted with respect to the open-world semantics, i. e. , one can not infer the dislike merely from the lack of a property assertion axiom `Adam like Eve`. Then this fact can be expressed with the following axiom in OWL 1: $\{Adam\} \sqsubseteq\neg\ (\exists likes\ \{Eve\})$.

3 Pattern Usage

The NPA pattern is useful in all situations where one has to model a negative property assertion but the language does not allow it directly. One real-world example here is the one mentioned in Section 2.3 where we had modeled a social network structure and need a possibility to express that one person does not like another and to ensure that one person does not know another person. With help of NPAs we have the possibility to distinguish between 'Person A does not know Person" and "We do not know whether Person A knows Person B or not".

4 Summary

The *Negative Property Assertion Pattern* allows to express negative property assertions in languages such as OWL 1 that do not a build-in constructor for this kind of assertions. In addition, when migrating OWL 1 ontologies to OWL 2 ontologies this pattern can be used to find negative property assertions and to transform them into the special NPA constructor of OWL 2.

References

1. Motik, B., Patel-Schneider, P.F., Parsia, B.: OWL 2 Structural Specification and Functional-Style Syntax. W3C Candidate Recommendation 11 June 2009 (June 2009)
2. Borckmans, S., Volz, R., Eberhart, A., Löffler, P.: Visual Modeling of OWL DL Ontologies Using UML. 198–213