Concept Partition Pattern

http:// ontologydesignpatterns.org/wiki/Submission:Partition

Olaf Noppens

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

1 Introduction

The *Partition Pattern* is a logical pattern that introduces axioms which model a partition of concepts. A partition is a general structure which is divided into several disjoint parts. With respect to ontologies the structure is a concept which is divided into several pair-wise disjoint concepts. This pattern reflects the simplest case where a named concept is defined as a partition of concepts.

2 Pattern

2.1 Problem

The *Partition Pattern* describes how to model a partition, i. e., a named concept which is divided into several disjoint concepts. Applying this pattern to an ontology will introduce the necessary axioms.

2.2 Solution

Let P be a named concept that is the partition which is divided into several concepts C_i. Then the partition is defined by introducing the following axioms (expressed in the standard DL syntax) as also be shown in Figure 1:
$P \equiv C_0 \sqcup C_1 \sqcup \cdots \sqcup C_n$ and $C_i \sqcap C_j = \bot$ for $0 \leq i, j \leq n, i \neq j$.
Note that some ontology languages such as OWL 2 [1] provides disjointness axioms as syntactical sugar to make the definition of pair-wise disjointness easier. In OWL 2 the pattern can be translated into the following axioms (expressed in OWL 2 abstract syntax):

```
EquivalentClasses(P ObjectUnionOf(C0, C1, ..., Cn))
DisjointClasses(C0, C1, ..., Cn)
```

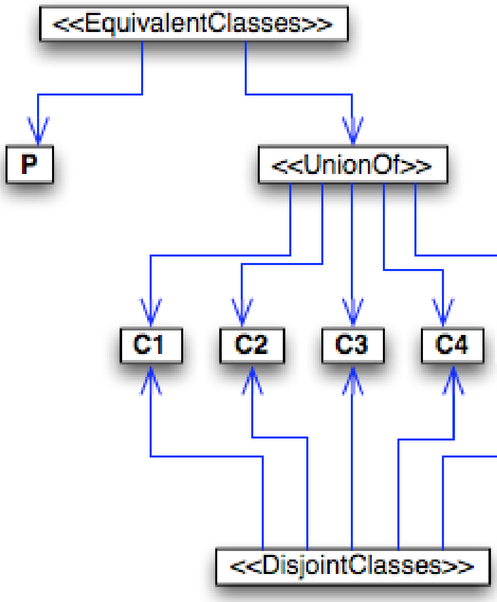

Fig. 1. Graphical representation of the *Partition Pattern*. Here, *EquivalentClasses* and *DisjointClasses* are axioms and *ObjectUnionOf* is a class constructor building a disjunction over an arbitrary number of concepts C_0, ..., C_n.

2.3 Example

Consider a world where only humans and animals live. Then the inhabitants of this world are partitioned into humans and animals. The following two axioms

```
EquivalentClasses(Inhabitant ObjectUnionOf(Human, Animal))
DisjointClasses(Human, Animal)
```

describe the partitions of inhabitants into human and animals.

3 Pattern Usage

In an ontology about family relationship we defined concepts such as `Person`, `Aunt` and `ParentOfSon` which are characterized by a relationships such as `hasChild` (resp. the inverse relationship `hasParent`), `hasSibling`, `married-with` as well as by the gender of people (`Male` respectively `Female`). There are a lot of similar ontologies about family relationships. Our version can be downloaded at http://www.informatik.uni-ulm.de/ki/Noppens/generation.owl.

```
EquivalentClasses(Parent-Of-Son
     ObjectSomeValuesFrom (has-Child Male) )
EquivalentClasses(Parent-Of-Daughter
     ObjectSomeValuesFrom(has-Child Female) )
EquivalentClasses(Aunt ObjectIntersectionOf(Uncle-Or-Aunt Female))
EquivalentClasses(Uncle-Or-Aunt
     ObjectIntersectionOf(Person
         ObjectSomeValuesFrom(has-Sibling Parent)))
```

The concept `Gender` is partitioned in `Male` and `Female`. Applying this pattern results in the following axioms:

```
EquivalentClasses(Gender, ObjectUnionOf(Male  Female))
DisjointClasses(Male Female)
```

4 Summary and Future Work

The *Partition Pattern* describes how to model a partition. The pattern reflects the simplest case where a *named* concept is the partition of (arbitrary) concepts. Future work will be concerned with a more general variant of this pattern: in some cases, the partition concept is not explicitly named (i. e., is not a named concept) but implicitly used, for instance, as value range of quantifications. In other words, no equivalent class axiom will be used but the value range is the union of the pair-wise disjoint parts of the partition.

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)