Knowledge representation patterns for concepts defined in a Norm

Discussion document

Author: Massimo Coletti

Director Security and Control Systems

Banca Finnat Euramerica S.p.A.

Version draft 0.1 released on May 31, 2006

Abstract

This paper discuss some issues related to the knowledge acquisition process from a corpus of normative documents.

The specific point discussed is the best pattern usable to represent the concepts described and defined by the norm text. The context of this project requires that the resulting ontologies will be aligned to the DOLCE+ foundational ontology.

Two patterns are discussed, one leading to an OWL-Full coding, the other compliant with OWL-DL limitations, with a preliminary analysis of benefits deriving from the two alternatives.

The document is intended as a discussion basis, within the project team, and the community of ontology practitioners.

Copyright Notice

This work was developed by: Banca Finnat Euramerica S.p.A., OpenTech S.r.l. and Massimo Coletti

This work is licensed under the Creative Commons Attribution-NoDerivs 2.5 Italy License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nd/2.5/it/ or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.

The problem

Our need is to create a unified conceptual view of a corpus of norms.

Each norm was analyzed, and the relevant concepts defined or referenced in the norm are listed. Now we have the problem of representing the conceptual relations among concepts, and between concepts and the norm sections defining them.

Our goal is to capture this conceptual model in an ontology, coded with the OWL language. Furthermore, we would like to align this conceptual model with DOLCE Lite plus [DOLCE], a foundational ontology. This alignment is aimed at obtaining a well-founded conceptual model.

The Solution

I have developed two patterns to solve this problem. The first pattern require the adoption of OWL-Full, but is more flexible.

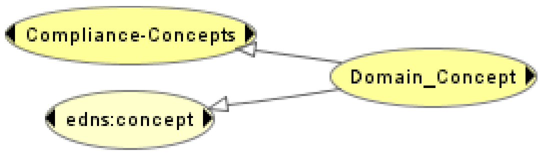
The second pattern allows an OWL-DL flavor, but gives to the knowledge engineer a rigid framework.

First pattern: Concepts as classes

This pattern was suggested by [Noy2005]. The concepts found in the normative corpus are coded as `owl:Class`, as subclasses of a generic class.

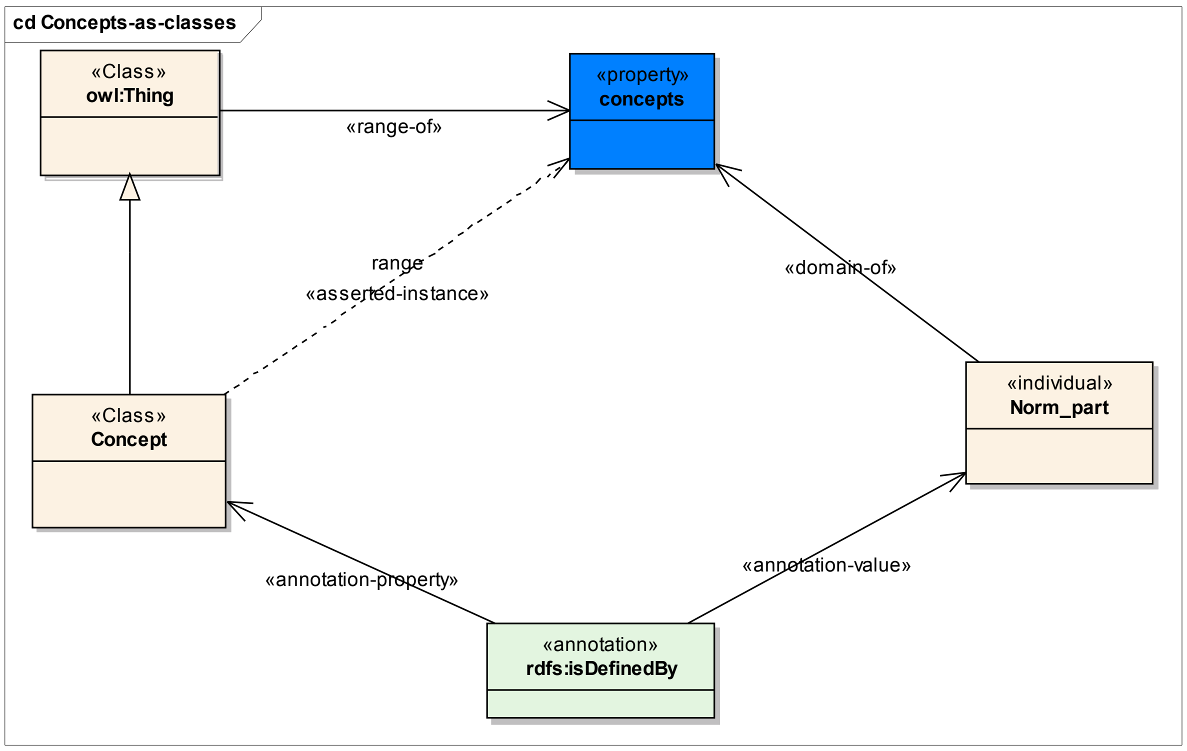
Furthermore – as an alignment to the DOLCE ontology – the concept is classified as a subclass of `edns:concept`.

The following diagram shows the class hierarchy:



The Compliance-Concepts class is an abstract superclass used to group all the concepts defined in our ontology.

The following UML diagram shows instead the pattern diagram:



In the diagram, you can see that the Concept is coded as a "Class" in the OWL terminology, while "Norm_part" (is the concepts that defines the sections from which a norm is composed) are coded as individuals.

The "concepts" property, which lists all the concepts that are defined within the text of a single section has the top class `owl:Thing` as range. This features allows the selection of an `owl:Class` as an asserted instance of the range class.

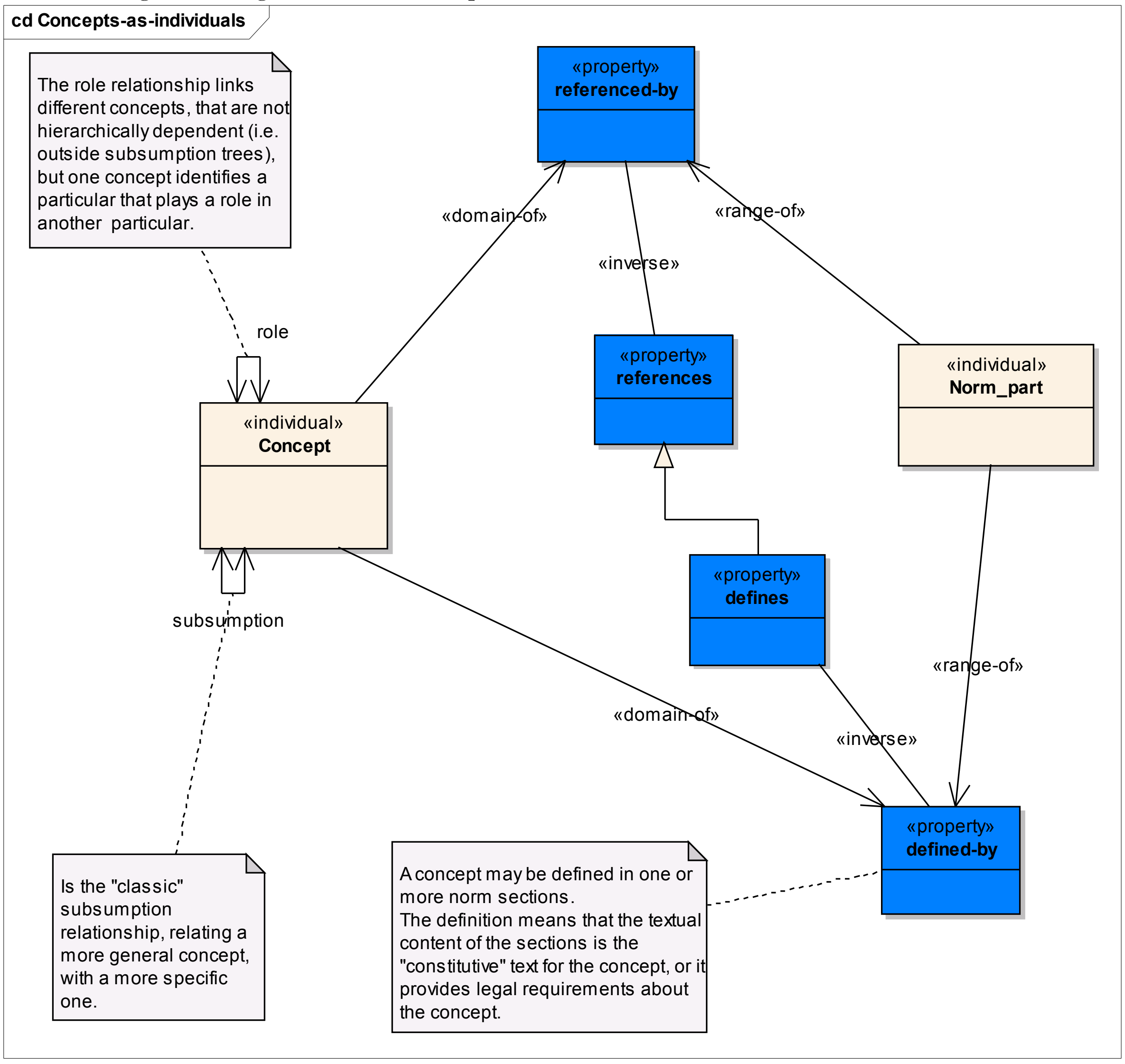
The inverse link is represented as an annotation property of the concept, using the `isDefinedBy` property.

The knowledge engineer is free to "configure" each single concept, adding properties and other elements that help capturing the doamain information.

Second pattern: concepts as individuals

This pattern sees the concepts defined as individual of a "Domain_Concept" class.

The following UML diagram illustrates the pattern:


cd Concepts-as-individuals
The role relationship links different concepts, that are not hierarchically dependent (i.e. outside subsumption trees), but one concept identifies a particular that plays a role in another particular.
«property» referenced-by
«domain-of»
«range-of»
«inverse»
role
«property» references
«individual» Norm_part
«individual» Concept
«property» defines
subsumption
«range-of»
«domain-of»
«inverse»
«property» defined-by
Is the "classic" subsumption relationship, relating a more general concept, with a more specific one.
A concept may be defined in one or more norm sections. The definition means that the textual content of the sections is the "constitutive" text for the concept, or it provides legal requirements about the concept.

The diagram shows that concepts are coded as individuals, as well as norm section elements.

Two properties link the two classes:

- `referenced-by`, classifies *weak* references of the concept in the text of the norm part;
- `defined-by`, states that the description of the concept can be found in the specified norm section.

The two properties have associated inverse properties, `references` and `defines`. There is a hierarchy between the two properties (represented with a generalization association), because the first subsumes the second.

The diagram shows also two relationships defined among individuals of the Concept class. The *subsumption* relationship links more generic concepts with more specific ones; it is the parallel of the rdfs:subClassOf relationship. The *role* relationship models many different kind of relationships where one concept plays some kind of role in another concept. Examples are:

- a "responsible" plays role in the organization that he manages;
- a "document" plays a role in the accomplishment of a norm prescription;
- a "policy" plays a role in the procedure followed during the execution of a process.

Solutions evaluations

Here I will try to develop a quality comparison of the two patterns, having in mind *ontology quality* measures and *usability* measures. My attempt is not methodologically correct; for a deeper analysis of ontology evaluation, you can see: [GANG05].

Criteria	***Pattern 1***	***Pattern 2***
Graph complexity	This pattern shows a simpler graph.	The increased complexity is a result of the complete expression of inverse relationships.
Reasoning complexity		This pattern is advantaged, having an OWL-DL profile. ✚
Verbosity		Is lower: disjoint predicates are not needed, as individuals are assumed to be different instances. ✚
Design flexibility	Is higher: concepts are expressed using the full OWL language features. ✚	If further binary predicates are required, the designer is however free to code specific subclasses, with the required properties.
User skill required		Is lower, as the architecture of the ontology is already defined. ✚
User interface	Dealing with class trees may allow a clearer comprehension of the conceptual model. ✚	

The result (of my evaluation) is a slight preference for pattern 2.

Transformation between patterns

The transformation from an OWL file coded according to the first pattern, to one coded according to the second (and vice versa) is straightforward.

An XSL stylesheet have been developed for this kind of transformation.

The availability of a simple transformation pattern is a guarantee for the developer that the switch from one pattern to the other is not so painful.

First pattern example

This is a concept coded according to the first pattern:

```
<owl:Class rdf:ID="Garante">
        <rdfs:subClassOf rdf:resource="&nof;Domain_Concept"/>
        <owl:disjointWith rdf:resource="#...."/>
        ...
        <rdfs:isDefinedBy rdf:resource="&l196;art153"/>
        <rdfs:isDefinedBy rdf:resource="&l196;art154"/>
</owl:Class>
```

Second pattern example

This is a concept coded according to the second pattern:

```
<nof:Domain_Concept rdf:ID="garante">
        <nof:defined-by rdf:resource="&l196;art153"/>
        <nof:defined-by rdf:resource="&l196;art154"/>
        <edns:specializes rdf:resource="#authority"/>
</nof:Domain_Concept>
```

References

[DOLCE]   " DOLCE : a Descriptive Ontology for Linguistic and Cognitive Engineering" 2006 Laboratory of Applied Ontology, Italian National research Council

[Noy2005] Natasha Noy, Michael Uschold, Chris Welty " Representing Classes As Property Values on the Semantic Web" 2005 http://www.w3.org/TR/swbp-classes-as-values/ ,