

# Syntax Proposal for Nominal Schemas

David Carral Martínez, Adila A. Krisnadhi, Pascal Hitzler

Kno.e.sis Center, Wright State University  
Technical Report, June 2011

## Abstract

This paper proposes a syntax representation for nominal schemas, a new description-logic style extension of OWL 2 [2, 3]. Necessary and minimal modifications to Functional and Manchester grammars are included as well as mappings from these two syntaxes to Turtle. An appendix with two examples is added in order to clarify the proposal.

## 1 Introduction

Nominal schemas [6, 7] are a new description-logic style extension of OWL 2 [11] which can be used like "variable nominal classes" within axioms. Nominal Schemas have been introduced in description logics (DL) syntax.

While the semantic intuition behind nominal schemas is the same as the one behind DL-safe variables presented in [9], the difference lies in the fact that DL-safe variables are tied to rule languages, while nominal schemas integrate seamlessly with DL syntax. The proposed extension encompasses DL-safe variable SWRL [5, 10, 6] while staying within the DL/OWL language paradigm and without employing hybrid approaches.

Nominal schemas have been introduced as a new general constructor for DL, denoted by the letter  $\mathcal{V}$  in the DL nomenclature (define DL  $\mathcal{SROIQV}$  as an extension of the description logic  $\mathcal{SROIQ}$ ). Worst-case complexity remains N2EXPTIME-complete after the addition of this new DL extension.

We present an example of nominal schemas.

Rules such as (1) are not expressible in current OWL 2.

$$\text{hasFather}(x,y) \wedge \text{hasBrother}(y,z) \wedge \text{hasTeacher}(x,z) \rightarrow \text{ChildTaughtByUncle}(x) \quad (1)$$

In contrast, using nominal schemas, rule (1) can be expressed as (2).

$$\exists \text{hasTeacher}.\{z\} \sqcap \exists \text{hasFather}.\exists \text{hasBrother}.\{z\} \sqsubseteq \text{ChildTaughtByUncle}. \quad (2)$$

The expression  $\{z\}$  is a nominal schema, which is to be read as a variable nominal that can only represent nominals (i.e.,  $z$  binds to known individuals), where the binding is the same for all occurrences of the nominal schema in an axiom. Variables  $x$  and  $y$  can still take arbitrary values and be hidden in the DL axiom,  $z$  needs to be restricted to be DL-safe to retain the conclusion.

For a more detailed description of nominal schemas including their formal semantics see [7].

This document proposes different ways to represent nominal schemas in the main syntaxes of OWL: Functional, Manchester, Turtle and RDF/XML. For an introduction of the OWL syntaxes consult [11]. Mapping from Turtle triples to RDF/XML is a well defined and automatized process so the RDF/XML based syntax will not be directly addressed in this document, it is assumed that it can be easily derived from the Turtle Syntax.

New reserved words are presented to mark the appearance of nominal schemas in the different syntaxes (Functional, Manchester and Turtle) as well as the necessary modifications to their grammars (Functional and Manchester). The representation of nominal schemas in Turtle syntax is defined by the mappings from Functional and Manchester.

Several approaches were considered for the representation and storage of nominal schemas, such as the use of entities with the ontology namespace, but this paper proposes the use of string literals. Using this approach we prevent the possible overlap that could be produced by giving the same name to two different nominal schemas. If these are declared as entities and, by error, two of them share the same name they will end up pointing to the same node in an RDF graph when they most likely refer to different individuals.

The selected approach, the use of a `xsd:string` datatype, is also considered by the RIF XML format [12]. Note that the same nominal schema can never appear in two different statements of an ontology. A nominal schema will only be related with one single axiom. By using a string type the occurrence of the nominal schema is exclusively bound to the axiom where it appears and the same string could be repeated in different axioms along the ontology safely. Even if two nominal schemas use the same string it will be considered as different occurrences of a datatype and therefore, they will be two separated nodes in an RDF graph.

Using the underscore to mark the appearance of a nominal schema, as it is done for Turtle blank nodes, was also considered. This approach was rejected because it could induce errors. Although in some cases both nominal schemas and blank nodes can represent individuals in an RDF graph they are completely different concepts. Using the underscore to mark both could be tricky and would make mappings from and to Turtle syntax difficult to define. With such a similar syntax the mapping may produce errors confusing nominal schemas with blank nodes and problems may arise when we want to move from the Turtle syntax to an RDF Graph.

The document is structured as follows. Section 2 contains the necessary modifications that have to be made to the Manchester and Functional Syntax grammars in order to include nominal schemas. Section 3 refers to the mappings from these syntaxes to Turtle and. Section 4 includes the conclusions of the proposal. An appendix A is provided with two examples using nominal schemas in the different syntaxes that have been discussed in the document.

## 2 Grammar Modifications

We propose several changes to the grammars of the different OWL syntaxes in order to include nominal schemas. The presented changes are designed to be minimal and imply very small modifications to the formal definitions of these grammars.

### Functional Syntax Grammar Modifications

We define in this section the required modifications we propose for the Functional Syntax grammar [1]. The reserved word `ObjectVariable` will be used to mark the appearance of the nominal schema. The nominal schema will be in parentheses and will always be followed by the expression `'^xsd:string'`. The proposed

changes are as follows

Add the last line, (**ObjectVariable**), to the **ClassExpression** production rule:

```
ClassExpression :=  
Class |  
ObjectIntersectionOf | ObjectUnionOf | ObjectComplementOf | ObjectOneOf |  
ObjectSomeValuesFrom | ObjectAllValuesFrom | ObjectHasValue | ObjectHasSelf |  
ObjectMinCardinality | ObjectMaxCardinality | ObjectExactCardinality |  
DataSomeValuesFrom | DataAllValuesFrom | DataHasValue |  
DataMinCardinality | DataMaxCardinality | DataExactCardinality |  
ObjectVariable
```

Add the next production rule to the grammar:

```
ObjectVariable := 'ObjectVariable (' quotedString'^^xsd:string)'
```

Although nominal schemas are not conceptually class expressions, their addition in this part of the grammar has been chosen in order to keep the modifications as small as possible.

## Manchester Syntax Grammar Modifications

Again, the reserved word **ObjectVariable** will be used to mark the appearance of the nominal schemas in the Manchester Syntax [4]. As in the Functional Syntax, the nominal will be in parentheses and followed by <sup>^^xsd:string</sup>. The needed changes to this grammar are:

Add the last line, (**ObjectVariable**), to **atomic** production rule:

```
atomic :=  
classIRI |  
'{individualList}' |  
'(description)' |  
ObjectVariable
```

Add the next production rule to the grammar:

```
ObjectVariable := 'ObjectVariable (' quotedString'^^xsd:string)'
```

## 3 Mapping FS and MS to Turtle

We define the syntax of nominal schemas in Turtle through the mapping from Functional and Manchester Syntaxes to the triple-notation. We assume that from this notation the process to move to RDF/XML is already formalized so, as said before, the XML syntax will not be directly addressed in this document.

### Functional Syntax to and from Turtle

The W3C document containing the formal mapping from FS to Turtle can be found in [8]. To add nominal schemas syntax to the mappings add the next row to the mapping from FS to Turtle:

Functional-Style Syntax	S Triples Generated in an Invocation of T(S)	Main Node of T(S)
ObjectVariable("v1"^^xsd:string)	_:x rdf:type owl:ObjectVariable _:x owl:variableId "v1"	_:x

And add the next row to the mapping from Turtle to FS:

RDF/XML Triples	Functional Syntax
_:x rdf:type owl:ObjectVariable _:x owl:variableId "v1"	ObjectVariable("v1"^^xsd:string)

## Manchester Syntax to and from Turtle

The mapping through Manchester Syntax and Turtle is defined in a similar way as the one from the Functional Syntax. To include the nominal schema in this mapping we need to add the next row to the table from MS to Turtle:

Manchester-Style Syntax	S Triples Generated in an Invocation of T(S)	Main Node of T(S)
Variable "v1"^^xsd:string	_:x rdf:type owl:ObjectVariable _:x owl:variableId "v1"	_:x

And add the next row to the mapping from Turtle to FS:

RDF/XML Triples	Manchester Syntax
_:x rdf:type owl:ObjectVariable _:x owl:variableId "v1"	Variable "v1"^^xsd:string

## 4 Conclusions

In this document we propose ways for representing nominal schemas in the different syntaxes of the OWL language. Reserved words have been provided for Functional, Manchester, Turtle and RDF/XML syntaxes, along with the consistent modifications to their grammars and mapping functions. Nominal schemas will be stored as string values in the OWL syntaxes to prevent overlapping errors. In the appendix of this document two examples are presented showing nominal schemas across the different covered syntaxes of OWL.

*Acknowledgements* This work was partially supported by the National Science Foundation under award 1017225 “III: Small: TROn—Tractable Reasoning with Ontologies” and by EPSRC in project “Hermit: Reasoning with Large Ontologies” (EP/F065841/1). The first author acknowledges support from Programa de Intercambio de la Universidad Pontificia de Salamanca 2010/11. The third author acknowledges support by a Fulbright Indonesia Presidential Scholarship PhD Grant 2010.

## References

- [1] Peter F. Patel-Schneider Boris Motik and Bijan Parsia, editors. *OWL 2 Web Ontology Language: Structural Specification and Functional-Style*. W3C Recommendation 27 October 2009, 2009. <http://www.w3.org/TR/owl2-syntax/>.
- [2] Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, and Sebastian Rudolph, editors. *OWL 2 Web Ontology Language: Primer*. W3C Recommendation 27 October 2009, 2009. Available from <http://www.w3.org/TR/owl2-primer/>.

- [3] Pascal Hitzler, Markus Krötzsch, and Sebastian Rudolph. *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC, 2009.
- [4] Matthew Horridge and Peter F. Patel-Schneider. *OWL 2 Web Ontology Language: Manchester Syntax*. W3C Working Group Note 27 October 2009, 2009. <http://www.w3.org/TR/owl2-manchester-syntax/>.
- [5] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosf, and Mike Dean. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*. W3C Member Submission 21 May 2004, 2004. Available from <http://www.w3.org/Submission/SWRL/>.
- [6] Adila A. Krisnadhi, Frederick Maier, and Pascal Hitzler. OWL and Rules. In *Reasoning Web 2011*, Lecture Notes in Computer Science. Springer, 2011. To appear.
- [7] Markus Krötzsch, Frederick Maier, Adila A. Krisnadhi, and Pascal Hitzler. A Better Uncle for OWL: Nominal Schemas for Integrating Rules and Ontologies. In S. Sadagopan, Krithi Ramamritham, Arun Kumar, M.P. Ravindra, Elisa Bertino, and Ravi Kumar, editors, *Proc. of the 20th International World Wide Web Conference (WWW'11)*, pages 645–654, Hyderabad, India, March/April 2011. ACM, New York.
- [8] Boris Motik and Peter F. Patel-Schneider, editors. *OWL 2 Web Ontology Language: Mapping to RDF Graphs*. W3C Recommendation 27 October 2009, 2009. <http://www.w3.org/TR/owl2-mapping-to-rdf/>.
- [9] Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for OWL-DL with rules. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 3(1):41–60, 2005.
- [10] Boris Motik, Ulrike Sattler, and Rudi Studer. Query answering for owl-dl with rules. *Web Semant.*, 3:41–60, July 2005.
- [11] W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview*. W3C Recommendation, 27 October 2009. Available at <http://www.w3.org/TR/owl2-overview/>.
- [12] Harold Boley Gary Hallmark Michael Kifer Adrian Paschke Axel Polleres Dave Reynolds, editor. *OWL 2 Web Ontology Language: Manchester Syntax*. W3C Recommendation 22 June 2010, 2010. <http://www.w3.org/TR/rif-core>.

## A Syntax Examples

### A.1 Example 1

#### Rule Syntax

hasFather( $x, y$ )  $\wedge$  hasBrother( $y, z$ )  $\wedge$  hasTeacher( $x, z$ )  $\wedge$   
 $\rightarrow$  ChildTaughtByUncle( $x$ )

#### DL Syntax

$\exists$ hasFather.( $\exists$ hasBrother. $\{z\}$ )  $\sqcap$   $\exists$ hasTeacher. $\{z\}$   $\sqsubseteq$  ChildTaughtByUncle

#### Functional Syntax

SubClassOf(

ObjectIntersectionOf(  
   ObjectSomeValuesFrom( :hasFather  
   ObjectSomeValuesFrom( :has Brother ObjectVariable("v1"^^xsd:string) ) )

```

    ObjectSomeValuesFrom( :hasTeacher ObjectVariable("v1"^^xsd:string) )
  )
  :ChildTaughtByUncle
)

```

### RDF/XML Syntax

```

.:x1 rdfs:subClassOf :ChildTaughtByUncle

.:x1 rdf:type owl:Class
.:x1 owl:intersectionOf ( .:x2 .:x3)

    .:x2 rdf:type owl:Restriction
    .:x2 owl:onProperty :hasFather
    .:x2 owl:someValuesFrom .:x5

    .:x3 rdf:type owl:Restriction
    .:x3 owl:onProperty :hasTeacher
    .:x3 owl:someValuesFrom .:x4

    .:x4 rdf:type owl:Restriction
    .:x4 owl:onProperty :hasBrother
    .:x4 owl:someValuesFrom :x6

    .:x5 rdf:type owl:ObjectVariable
    .:x5 owl:variableId "v1"

```

### Manchester Syntax

```

Class: ChildTaughtByUncle
SubClassOf:
  ( hasTeacher some (Variable "v1"^^xsd:string) )
  and
  ( hasSubmittedPaper some
    (hasFather some (hasBrother some (Variable "v1"^^xsd:string))) )

```

## A.2 Example 2

### Rule Syntax

```

hasReviewAssignment( $v, x$ )  $\wedge$  hasAuthor( $x, y$ )  $\wedge$  atVenue( $x, z$ )  $\wedge$ 
hasSubmittedPaper( $v, u$ )  $\wedge$  hasAuthor( $u, y$ )  $\wedge$  atVenue( $u, z$ )
 $\rightarrow$  ReviewerWithConflictingAssignment( $v$ )

```

### DL Syntax

```

 $\exists$ hasReviewAssignment. ( $\exists$ hasAuthor. $\{a\}$   $\sqcap$   $\exists$ atVenue. $\{b\}$ )  $\sqcap$ 
 $\exists$ hasSubmittedPaper. ( $\exists$ hasAuthor. $\{a\}$   $\sqcap$   $\exists$ atVenue. $\{b\}$ )
 $\sqsubseteq$  ReviewerWithConflictingAssignment

```

### Functional Syntax

```

SubClassOf(
  ObjectIntersectionOf(

```

```

ObjectSomeValuesFrom ( :hasReviewAssign ObjectIntersectionOf (
    ObjectSomeValuesFrom (:hasAuthor ObjectVariable("v1"^^xsd:string))
    ObjectSomeValuesFrom (:atVenue ObjectVariable("v2"^^xsd:string)) )
)
ObjectSomeValuesFrom ( :hasSubmittedPaper ObjectIntersectionOf (
    ObjectSomeValuesFrom (:hasAuthor ObjectVariable("v1"^^xsd:string))
    ObjectSomeValuesFrom (:atVenue ObjectVariable("v2"^^xsd:string)) )
)
)
)
:ReviewerWithConflictingAssign
)

```

### RDF/XML Syntax

```
_:x1 rdfs:subClassOf :ReviewerWithConflictingAssignment
```

```
_:x1 rdf:type owl:Class
```

```
_:x1 owl:intersectionOf ( _:x2 _:x3)
```

```
_:x2 rdf:type owl:Restriction
```

```
_:x3 rdf:type owl:Restriction
```

```
_:x2 owl:onProperty :hasReviewAssign
```

```
_:x3 owl:onProperty :hasSubmittedPaper
```

```
_:x2 owl:intersectionOf ( _:x4 _:x5)
```

```
_:x3 owl:intersectionOf ( _:x8 _:x9)
```

```
_:x4 rdf:type owl:Restriction
```

```
_:x8 rdf:type owl:Restriction
```

```
_:x4 owl:onProperty :hasAuthor
```

```
_:x8 owl:onProperty :hasAuthor
```

```
_:x4 owl:someValuesFrom _:x6
```

```
_:x8 owl:someValuesFrom :x10
```

```
_:x6 rdf:type owl:ObjectVariable
```

```
_:x10 rdf:type owl:ObjectVariable
```

```
_:x6 owl:variableId "v1"
```

```
_:x10 owl:variableId "v1"
```

```
_:x5 rdf:type owl:Restriction
```

```
_:x9 rdf:type owl:Restriction
```

```
_:x5 owl:onProperty :atVenue
```

```
_:x9 owl:onProperty :atVenue
```

```
_:x5 owl:someValuesFrom _:x7
```

```
_:x9 owl:someValuesFrom :x11
```

```
_:x7 rdf:type owl:ObjectVariable
```

```
_:x11 rdf:type owl:ObjectVariable
```

```
_:x7 owl:variableId "v2"
```

```
_:x11 owl:variableId "v2"
```

### Manchester Syntax

```
Class: ReviewerWithConflictingAssignment
```

```
SubClassOf:
```

```
( hasReviewAssign some
```

```
( (hasAuthor some (Variable "v1"^^xsd:string)) and (atVenue some (Variable "v2"^^xsd:string)) ) )
```

```
and
```

```
( :hasSubmittedPaper some
```

```
( (hasAuthor some (Variable "v1"^^xsd:string)) and (atVenue some (Variable "v2"^^xsd:string)) ) )
```