MANCHESTER 1824

The University of Manchester

OWL 1.0 Implementation Experience

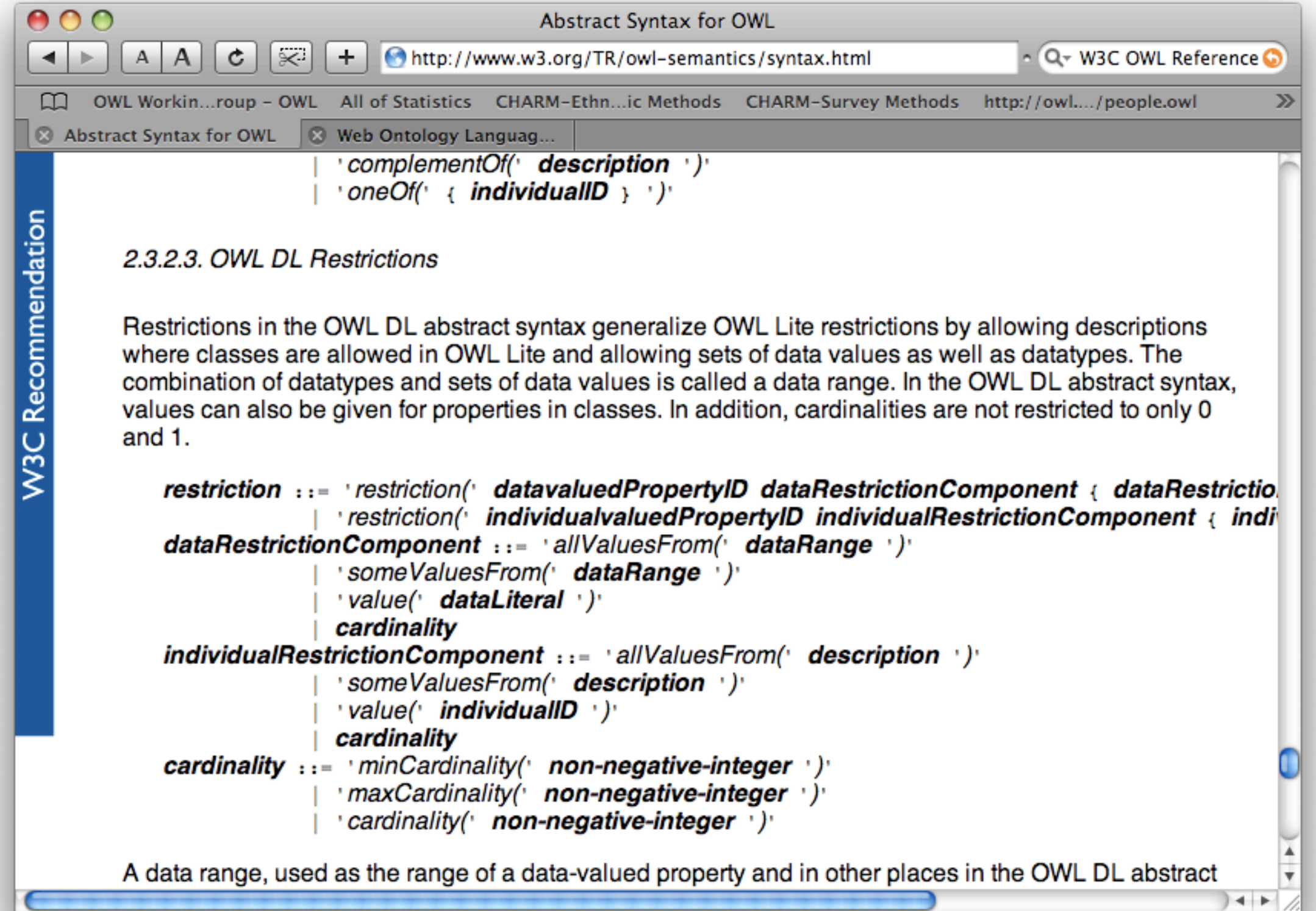
Sean Bechhofer and Matthew Horridge

RDF

Imports

The University of Manchester

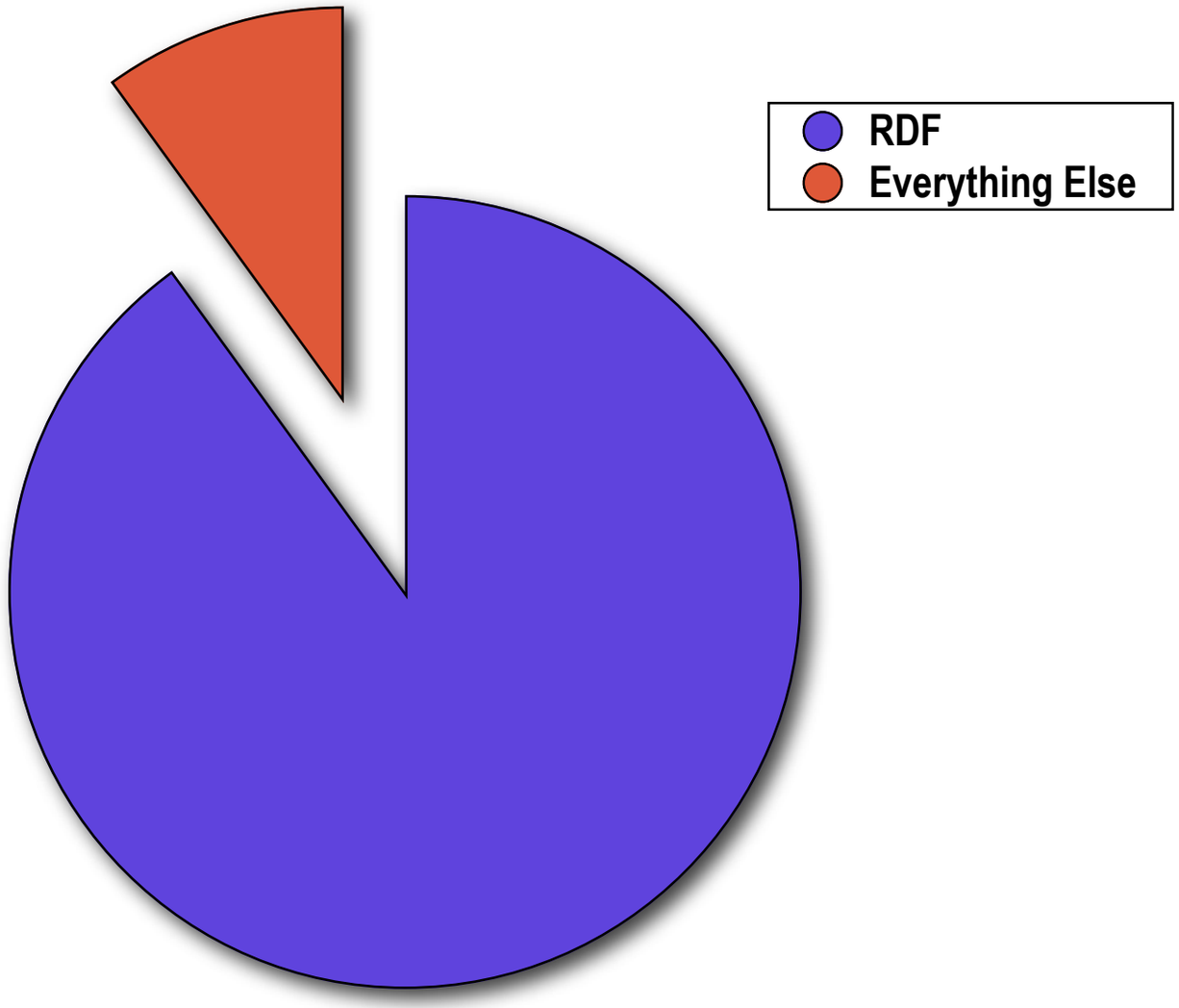
The OWL API



MANCHESTER 1824

The University of Manchester

Distribution of Effort



MANCHESTER 1824

The University of Manchester

RDF Mapping

Use as little time and memory as possible

Streaming vs. Non-Streaming

The University of Manchester

RDF Mapping

OWL Web Ontology Language Parsing OWL in RDF/XML

http://www.w3.org/TR/owl-parsing/

W3C OWL

OWL Workin...roup - OWL | All of Statistics | CHARM-Ethn...ic Methods | CHARM-Survey Methods | http://owl..../people.owl | owl1-1 - Google Code

OWL Web Ontology Lan... | Web Ontology Languag...

W3C Working Group Note

OWL Web Onto
Parsing OWL in

W3C Working Grou

This version:
http://www.w3.org/TR
Latest version:
http://www.w3.org/TR
Author:
Sean Bechhofer (sea

Copyright © 2004 W3C® (MIT, E

Abstract

An OWL-RDF parser takes
represented in the RDF. Th
intended as a complete spe
parser that will deal with a

For example, we do not dis
concerned with spotting so

Status of This Docu

OWL 1.1 Web Ontology Language: Mapping to RDF Graphs

http://www.webont.org/owl/1.1/rdf_mapping.html

OWL 1.1

OWL Workin...roup - OWL | All of Statistics | CHARM-Ethn...ic Methods | CHARM-Survey Methods | http://owl..../people.owl | owl1-1 - Google Code

Negative object and data property assertions are already reified so only the following triples are added if an assertion contains an annotation:

$_:x\ T(apID_i)\ T(ct_i) \quad 1 \leq i \leq n$

3 Translation from RDF Graphs to Functional-Style Syntax

This section specifies how to translate a set of RDF triples G into an OWL 1.1 ontology in functional-style syntax O, if possible. The function `Type(x)` assigns a set of types to each resource node x in G (in this and all other definitions, the graph G is implicitly understood and is not specified explicitly) and is defined as the smallest set satisfying the conditions from Table 3.

Table 3. Types of Nodes in a Graph

If G contains a triple of this form...	...then `Type(x)` must contain this URI.
`x rdf:type owl:Class`	owl:Class
`x rdf:type owl:Restriction`	owl:Class
`x rdf:type owl11:ObjectRestriction`	owl:Class
`x rdf:type owl11:DataRestriction`	owl:Class
`x rdf:type owl:DataRange`	owl:DataRange
`x rdf:type owl:Datatype`	owl:DataRange
`x rdf:type owl:ObjectProperty`	owl:ObjectProperty
`x rdf:type owl:TransitiveProperty`	owl:ObjectProperty
`x rdf:type owl:SymmetricProperty`	owl:ObjectProperty
`x rdf:type owl11:AsymmetricProperty`	owl:ObjectProperty
`x rdf:type owl11:ReflexiveProperty`	owl:ObjectProperty
`x rdf:type owl11:IrreflexiveProperty`	owl:ObjectProperty

RDF Mapping

```
<A rdfs:subClassOf B>
```

```
Class(A partial B)
```

```
SubClassOf(A B)
```

The University of Manchester

RDF Mapping

```
axiom ::= 'DisjointClasses(' description description { description } ')'
        | 'EquivalentClasses(' description { description } ')'
        | 'SubClassOf(' description description ')'
```

DisjointClasses(A B C D)

Pairwise disjoint statements

MANCHESTER 1824

The University of Manchester

RDF Mapping

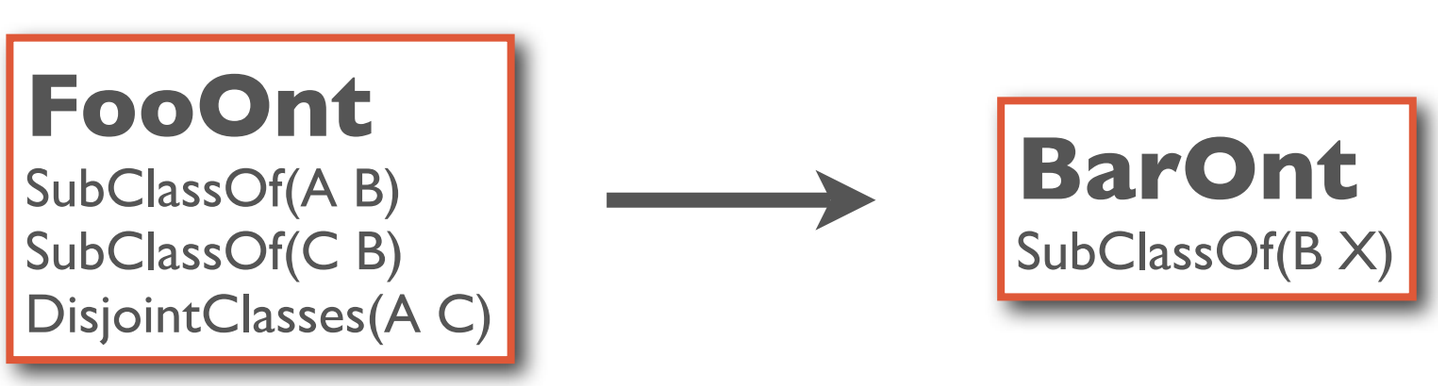
```
A rdfs:subClassOf B

_x owl:onProperty p
_x owl:someValuesFrom C

_x owl:onObjectProperty p
_x owl:someValuesFrom C
```

Imports

http://foo.com/fooOnt ⟶ http://bar.com/barOnt



MANCHESTER 1824

The University of Manchester

Imports

3.4. Interpreting Ontologies

From Section 2, an OWL ontology can have annotations, which need their own semantic conditions. Aside from this local meaning, an *owl:imports* annotation also imports the contents of another OWL ontology into the current ontology. The imported ontology is the one, if any, that has as **name** the argument of the imports construct. (This treatment of imports is divorced from Web issues. The intended use of names for OWL ontologies is to make the name be the location of the ontology on the Web, but this is outside of this formal treatment.)

MANCHESTER 1824

The University of Manchester

Imports

(OWL Reference)

7.3 Importing an ontology

An `owl:imports` statement references another OWL ontology containing definitions, whose meaning is considered to be part of the meaning of the importing ontology. **Each reference consists of a URI specifying from where the ontology is to be imported. Syntactically, `owl:imports` is a property with the class `owl:Ontology` as its domain and range.**

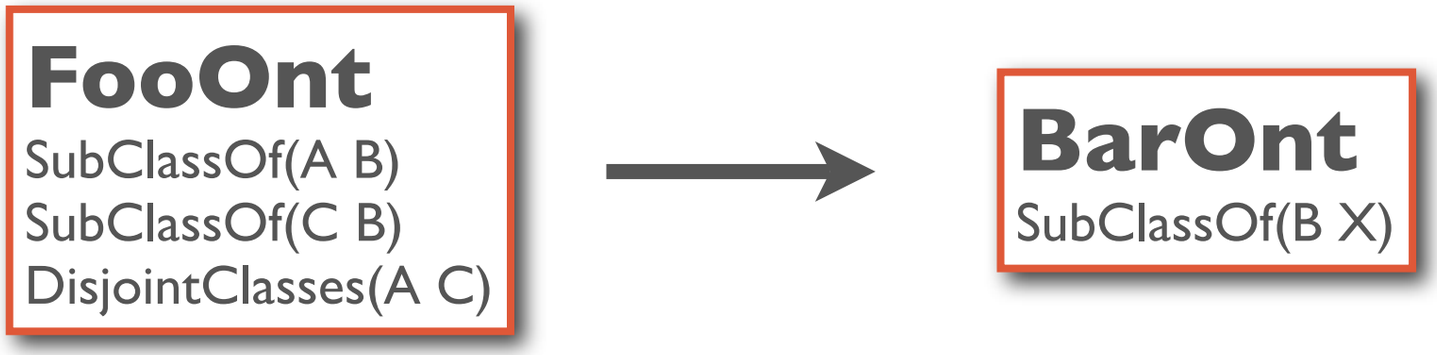
Imports

(OWL Guide)

`owl:imports` provides an include-style mechanism. `owl:imports` takes a single argument, identified by the `rdf:resource` attribute.

Importing another ontology brings the entire set of assertions provided by that ontology into the current ontology. **In order to make best use of this imported ontology it would normally be coordinated with a namespace declaration**. Notice the distinction between these two mechanisms. The namespace declarations provide a convenient means to *reference* names defined in other OWL ontologies. Conceptually, `owl:imports` is provided to indicate your intention to *include* the assertions of the target ontology. Importing another ontology, *O2*, will also import all of the ontologies that *O2* imports.

Imports



How do we determine the names of ontologies?

MANCHESTER 1824

The University of Manchester

Ontology Header

2.2. Ontology Headers

Once namespaces are established we normally include a collection of assertions about the ontology grouped under an `owl:Ontology` tag. These tags support such critical housekeeping tasks as comments, version control and inclusion of other ontologies.

```
<owl:Ontology rdf:about="">
  <rdfs:comment>An example OWL ontology</rdfs:comment>
  <owl:priorVersion rdf:resource="http://www.w3.org/TR/2003/PR-owl-guide-20031215/wine"/
  <owl:imports rdf:resource="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food"/>
  <rdfs:label>Wine Ontology</rdfs:label>
  ...
```

Note that we use '...' to indicate that there is additional text that has been elided for purposes of the example.

.....

The `rdf:about` attribute provides a name or reference for the ontology. Where the value of the attribute is "", the standard case, the name of the ontology is the base URI of the `owl:Ontology` element. **Typically, this is the URI of the document containing the ontology.** An exception to this is a context that makes use of `xml:base` which may set the base URI for an element to something other than the URI of the current document.

Conclusions

- Fix RDF mapping for Nary constructs - e.g. DisjointClasses.
- Stronger typing for RDF would be nicer
- Fix imports specification mess
- Test cases for OWL 1.0 were very useful - please can we have some for OWL 1.1?

class, ***objectExistsSelf*** denotes the set of objects that are connected to themselves via the given object property, and ***objectHasValue*** denotes the set of objects that are connected via the given object property to the object denoted by the given individual. The grammar for the restrictions is as follows:

objectAllValuesFrom := 'ObjectAllValuesFrom' '(' ***objectPropertyExpression description*** ')'
objectSomeValuesFrom := 'ObjectSomeValuesFrom' '(' ***objectPropertyExpression description*** ')'
objectExistsSelf := 'ObjectExistsSelf' '(' ***objectPropertyExpression*** ')'
objectHasValue := 'ObjectHasValue' '(' ***objectPropertyExpression individualURI*** ')'

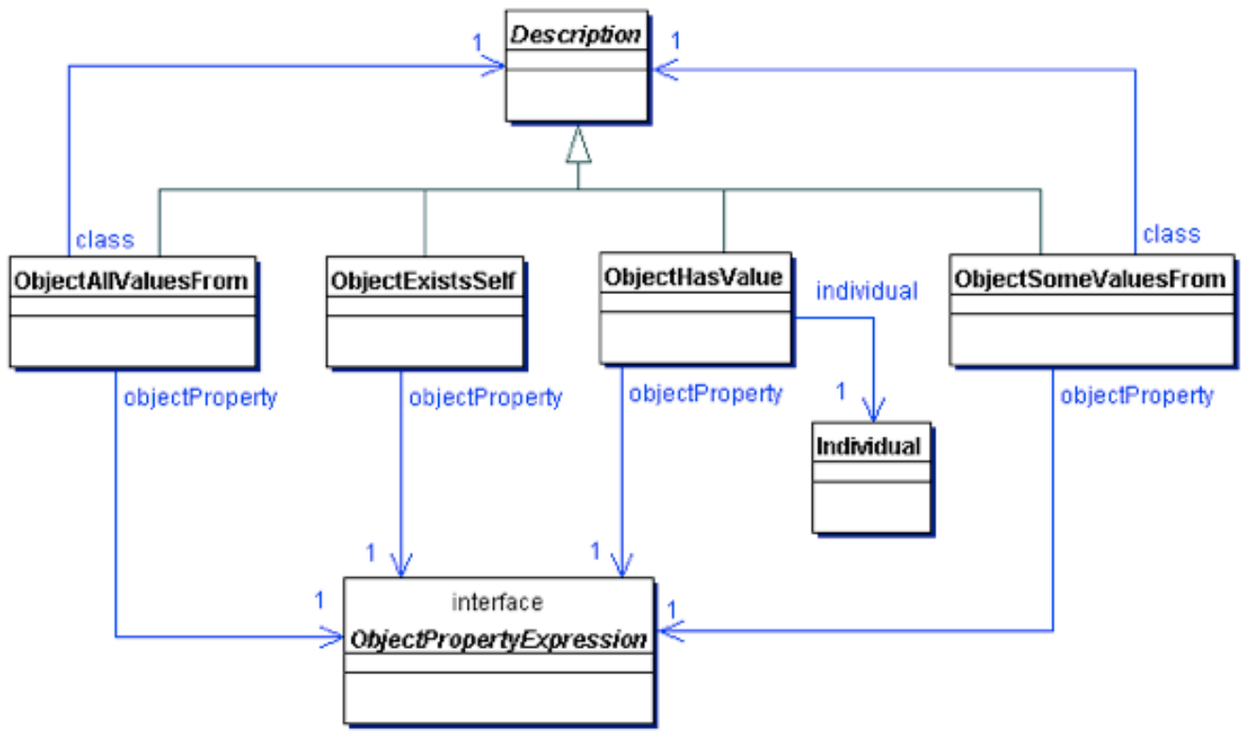


Figure 8. OWL 1.1 Classes Defined by Restriction on Object Properties

Finally, OWL 1.1 classes can be defined by restricting the cardinality of associations between objects, as shown in Figure 9. Cardinality restrictions can be qualified or unqualified, depending on whether there is a restriction on the class of the connected individual; an unqualified cardinality restriction is equivalent to a qualified one where the restricting class is `owl:Thing`. The class ***objectMinCardinality*** denotes the set of objects that are connected via the given object property to at least the given number of instances of the given class, the class ***objectMaxCardinality*** denotes the set of objects that are connected via the given object property to at most the given number of