

Comparison of Semantic Web serialization syntaxes

Tony Mallia Edmond Scientific 7 March 2015

Introduction

This is the comparison of serialization syntaxes supported by Protégé.

The sample contains two files –

- inferenceTest1 contains the model of two classes – foo, bar
- inferenceTest2 contains an instance individualbar and imports inferenceTest1

The model is testing the pattern by Eric Prud'hommeaux for inferring type from a Dataproperty and is described at the end of this paper.

The following syntaxes are compared:

- RDF/XML
- OWL/XML
- Turtle
- OWL Functional Syntax Rendering

Other syntaxes which could be used but are not included here are:

- Manchester OWL Syntax
- OBO
- KRSS2
- Latex

If there is specific interest these examples can be generated.

JSON-LD is not available with Protégé.

RDF/XML

testInference1

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY testInference1 "http://testInference1#" >
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://testInference1#"
  xml:base="http://testInference1"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```

xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:testInference1="http://testInference1#">
<owl:Ontology rdf:about="http://testInference1"/>

<!--
////////////////////////////////////
// Data properties
////////////////////////////////////
-->

<!-- http://testInference1#isType -->

<owl:DatatypeProperty rdf:about="&testInference1;isType"/>

<!--
////////////////////////////////////
// Classes
////////////////////////////////////
-->

<!-- http://testInference1#bar -->

<owl:Class rdf:about="&testInference1;bar">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <rdf:Description rdf:about="&testInference1;foo"/>
        <owl:Restriction>
          <owl:onProperty rdf:resource="&testInference1;isType"/>
          <owl:hasValue>fum</owl:hasValue>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
</owl:Class>

<!-- http://testInference1#foo -->

<owl:Class rdf:about="&testInference1;foo">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</owl:Class>

<!-- http://www.w3.org/2002/07/owl#Thing -->

<rdf:Description rdf:about="&owl;Thing">
  <rdfs:subClassOf rdf:resource="&owl;Thing"/>
</rdf:Description>
</rdf:RDF>

```

testInference2

```

<?xml version="1.0"?>

<!DOCTYPE rdf:RDF [
  <!ENTITY testInference1 "http://testInference1#" >
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<rdf:RDF xmlns="http://testInference2#"
  xml:base="http://testInference2#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

```

```

    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:testInference1="http://testInference1#">
<owl:Ontology rdf:about="http://testInference2">
  <owl:imports rdf:resource="http://testInference1"/>
</owl:Ontology>

<!--
////////////////////////////////////
// Individuals
////////////////////////////////////
-->

<!-- http://testInference2#individualBar -->

<owl:NamedIndividual rdf:about="http://testInference2#individualBar">
  <rdf:type rdf:resource="&testInference1;foo"/>
  <testInference1:isType>fum</testInference1:isType>
</owl:NamedIndividual>
</rdf:RDF>

```

OWL/XML

testInference1

```

<?xml version="1.0"?>

<!DOCTYPE Ontology [
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<Ontology xmlns="http://www.w3.org/2002/07/owl#"
  xml:base="http://testInference1"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xml="http://www.w3.org/XML/1998/namespace"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  ontologyIRI="http://testInference1">
  <Prefix name="" IRI="http://testInference1#"/>
  <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#"/>
  <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#"/>
  <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#"/>
  <Prefix name="testInference1" IRI="http://testInference1#"/>
  <Declaration>
    <Class IRI="#bar"/>
  </Declaration>
  <Declaration>
    <Class IRI="#foo"/>
  </Declaration>
  <Declaration>
    <DataProperty IRI="#isType"/>
  </Declaration>
  <EquivalentClasses>
    <Class IRI="#bar"/>
    <ObjectIntersectionOf>
      <Class IRI="#foo"/>
      <DataHasValue>
        <DataProperty IRI="#isType"/>
        <Literal datatypeIRI="&rdf;PlainLiteral">fum</Literal>
      </DataHasValue>
    </ObjectIntersectionOf>
  </EquivalentClasses>
  <SubClassOf>
    <Class IRI="#foo"/>

```

```

        <Class abbreviatedIRI="owl:Thing"/>
    </SubClassOf>
    <SubClassOf>
        <Class abbreviatedIRI="owl:Thing"/>
        <Class abbreviatedIRI="owl:Thing"/>
    </SubClassOf>
</Ontology>

<!-- Generated by the OWL API (version 3.5.0) http://owlapi.sourceforge.net -->

```

testInference2

```

<?xml version="1.0"?>

<!DOCTYPE Ontology [
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY xml "http://www.w3.org/XML/1998/namespace" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
]>

<Ontology xmlns="http://www.w3.org/2002/07/owl#"
    xml:base="http://testInference2"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xml="http://www.w3.org/XML/1998/namespace"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    ontologyIRI="http://testInference2">
    <Prefix name="rdf" IRI="http://www.w3.org/1999/02/22-rdf-syntax-ns#" />
    <Prefix name="rdfs" IRI="http://www.w3.org/2000/01/rdf-schema#" />
    <Prefix name="xsd" IRI="http://www.w3.org/2001/XMLSchema#" />
    <Prefix name="owl" IRI="http://www.w3.org/2002/07/owl#" />
    <Import>http://testInference1</Import>
    <Declaration>
        <NamedIndividual IRI="#individualBar"/>
    </Declaration>
    <ClassAssertion>
        <Class IRI="http://testInference1#foo"/>
        <NamedIndividual IRI="#individualBar"/>
    </ClassAssertion>
    <DataPropertyAssertion>
        <DataProperty IRI="http://testInference1#isType"/>
        <NamedIndividual IRI="#individualBar"/>
        <Literal datatypeIRI="&rdf;PlainLiteral">fum</Literal>
    </DataPropertyAssertion>
</Ontology>

<!-- Generated by the OWL API (version 3.5.0) http://owlapi.sourceforge.net -->

```

Turtle

testInference1

```

@prefix : <http://testInference1#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix testInference1: <http://testInference1#> .
@base <http://testInference1> .

<http://testInference1> rdf:type owl:Ontology .

```

```
#####
#   Data properties
#####

### http://testInference1#isType

:isType rdf:type owl:DatatypeProperty .

#####
#   Classes
#####

### http://testInference1#bar

:bar rdf:type owl:Class ;

    owl:equivalentClass [ rdf:type owl:Class ;
                          owl:intersectionOf ( :foo
                                                  [ rdf:type owl:Restriction ;
                                                    owl:onProperty :isType ;
                                                    owl:hasValue "fum"
                                                  ]
                                                )
                          ] .

### http://testInference1#foo

:foo rdf:type owl:Class ;

    rdfs:subClassOf owl:Thing .

### http://www.w3.org/2002/07/owl#Thing

owl:Thing rdfs:subClassOf owl:Thing .

### Generated by the OWL API (version 3.5.0) http://owlapi.sourceforge.net
```

testInference2

```
@prefix : <http://testInference2#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@base <http://testInference2> .

<http://testInference2> rdf:type owl:Ontology ;

    owl:imports <http://testInference1> .

#####
#   Individuals
#####

### http://testInference2#individualBar

:individualBar rdf:type <http://testInference1#foo> ,
              owl:NamedIndividual ;

    <http://testInference1#isType> "fum" .

### Generated by the OWL API (version 3.5.0) http://owlapi.sourceforge.net
```

OWL Functional Syntax Rendering

testInference1

```
Prefix(:=<http://testInference1#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)
Prefix(testInference1:=<http://testInference1#>)
```

```
Ontology(<http://testInference1>
```

```
Declaration(Class(:bar))
Declaration(Class(:foo))
Declaration(DataProperty(:isType))
EquivalentClasses(:bar ObjectIntersectionOf(DataHasValue(:isType "fum") :foo))
SubClassOf(:foo owl:Thing)
SubClassOf(owl:Thing owl:Thing)
)
```

testInference2

```
Prefix(:=<http://testInference2#>)
Prefix(owl:=<http://www.w3.org/2002/07/owl#>)
Prefix(rdf:=<http://www.w3.org/1999/02/22-rdf-syntax-ns#>)
Prefix(xml:=<http://www.w3.org/XML/1998/namespace>)
Prefix(xsd:=<http://www.w3.org/2001/XMLSchema#>)
Prefix(rdfs:=<http://www.w3.org/2000/01/rdf-schema#>)
```

```
Ontology(<http://testInference2>
```

```
Import(<http://testInference1>)

Declaration(NamedIndividual(:individualbar))
ClassAssertion(<http://testInference1#foo> :individualbar)
DataPropertyAssertion(<http://testInference1#isType> :individualbar "fum")
)
```

Example Description

The example tests a reasoner (Hermit) to infer the subclass for an individual “individualbar” which is declared to be type “foo”. Type “bar” has an equivalent class expression which is that it is a subclass of “foo” and has a dataproperty “isType” with a value “fum”.

Thus after running the reasoner, “individualbar” is inferred to have a type “bar”.

This is the mechanism for assigning a terminology type to an individual fhir:code or fhir:coding based on the value of the dataproperties (value) of system and code objects. The example just shows the equivalent of code – “bar”. “foo” is the equivalent of a simple valueset.