

Coding an OWL ontology in HTML5 and RDFa

Egon Willighagen 

Published September 27, 2015

Citation

Willighagen, E. (2015). Coding an OWL ontology in HTML5 and RDFa. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/4k55a-8c261>

Keywords

Ontology, Bridgedb, Web, Rdf, Owl

Copyright

Copyright © Egon Willighagen 2015. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

There are many fancy tools to edit ontologies. I like simple editors, like [nano](#). And like any hacker, I can hack [OWL](#) ontologies in nano. The hacking implies OWL was never meant to be hacked on a simple text editor; I am not sure that is really true. Anyways, [HTML5](#) and [RDFa](#) will do fine, and here is a brief write up. This post will not cover the basics of RDFa and does assume you already know how triples work. If not, read this [RDFa primer](#) first.



The BridgeDb DataSource Ontology

This page describes the BridgeDb ontology. Make sure to visit our [homepage](#) to

Classes

Data Source

A resource that defines identifiers for some biological entity, like a gene, prote

Organism

An organism

The BridgeDb DataSource Ontology

This example uses the [BridgeDb DataSource Ontology](#), created by BridgeDb developers from Manchester University (Christian, Stian, and Alasdair). The ontology covers describing data sources of identifiers, a technology outlined in the BridgeDb paper by Martijn (see below) as well as terms from the Open PHACTS [Dataset Descriptions for the Open Pharmacological Space](#) by Alasdair et al.

Because I needed to put this online for [Open PHACTS](#) (BTW, [the project won a big award!](#)) and our previous solution did not work well enough anymore. You may also see the HTML of the result first. You may also want to verify it really is [HTML](#): here is the [HTML5 validation report](#). Also, you may be interested in what the ontology in RDF looks like: here is [the extracted RDF for the ontology](#). Now follow the HTML+RDFa snippets. First, the ontology details (actually, I have it split up):

```
<div about="http://vocabularies.bridgedb.org/ops#"
  typeof="owl:Ontology">
  <h1>The <span property="rdfs:label">BridgeDb DataSource Ontology</span>
    (version <span property="owl:versionInfo">2.1.0</span></h1>
  <p>
    This page describes the BridgeDb ontology. Make sure to visit our
    <a property="rdfs:seeAlso" href="http://www.bridgedb.org/">homepage</a> too!
  </p>
</div>
<p about="http://vocabularies.bridgedb.org/ops#">
  The OWL ontology can be extracted
  <a property="owl:versionIRI"
    href="http://www.w3.org/2012/pyRdfa/extract?uri=http://vocabularies.bridgedb.org/ops#"
  >The Open PHACTS specification on
  <a property="rdf:seeAlso"
    href="http://www.openphacts.org/specs/2013/WD-datadesc-20130912/#bridgedb"
  >Dataset Descriptions</a> is also useful.
</p>
```

This is the last time I show the custom color coding, but for a first time it is useful. In red are basically the predicates, where `@about` indicates a new resource is started, `@typeof` defines the

chem-bla-ics

`rdf:type`, and `@property` indicates all other predicates. The blue and green blobs are literals and object resources, respectively. If you work this out, you get this OWL code (more or less):

```
bridgedb: a owl:Ontology;
  rdfs:label "BridgeDb DataSource Ontology"@en;
  rdf:seeAlso
    <http://www.openphacts.org/specs/2013/WD-datadesc-20130912/#bridgedb>;
  rdfs:seeAlso <http://www.bridgedb.org/>;
  owl:versionIRI
    <http://www.w3.org/2012/pyRdfa/extract?uri=http://vocabularies.bridgedb.org/ops#>;
  owl:versionInfo "2.1.0"@en .
```

An OWL class

Defining OWL classes are using the same approach: define the resource it is `@about`, define the `@typeOf` and giving is properties. BTW, note that I added a `@id` so that ontology terms can be looked up using the HTML # functionality. For example:

```
<div id="DataSource"
  about="http://vocabularies.bridgedb.org/ops#DataSource"
  typeof="owl:Class">
  <h3 property="rdfs:label">Data Source</h3>
  <p property="dc:description">A resource that defines
    identifiers for some biological entity, like a gene,
    protein, or metabolite.</p>
</div>
```

An OWL object property

Defining an OWL data property is pretty much the same, but note that we can arbitrary add additional things, making use of ``, `<div>`, and `<p>` elements. The following example also defines the `rdfs:domain` and `rdfs:range`:

```
<div id="aboutOrganism"
  about="http://vocabularies.bridgedb.org/ops#aboutOrganism"
  typeof="owl:ObjectProperty">
  <h3 property="rdfs:label">About Organism</h3>
  <p><span property="dc:description">Organism for all entities
    with identifiers from this datasource.</span>
    This property has
    <a property="rdfs:domain"
      href="http://vocabularies.bridgedb.org/ops#DataSource">DataSource</a>
    as domain and
    <a property="rdfs:range"
      href="http://vocabularies.bridgedb.org/ops#Organism">Organism</a>
```

chem-bla-ics

```
    as range.</p>  
</div>
```

So, now anyone can host an OWL ontology with dereferencable terms: to remove confusion, I have used the full URLs of the terms in **@about** attributes.