
MAPPING OBO TO OWL

Table of Contents

History	1
About	1
This Document	2
Downloading ontologies in OWL	2
GO in OWL	2
Other OBO ontologies in OWL	2
Inferred DL definitions	3
Status of mapping	3
Background Material	3
Identifiers	3
OBO Terms	4
OBO Typedefs	4
OBO Names	4
OBO Natural Language Definitions	5
OBO Metadata	5
OBO Instances	5
Subclassing	5
Relationships	5
Class definitions	6
Other OWL constructs	6
Tools	6
See Also	7
FAQ	7
ACKNOWLEDGEMENTS	8

Chris Mungall, BDGP

History

- 2004-02-02 :
- 2004-02-16 :
- 2005-03-15 : finalised intersection_of mapping
- 2005-12-01 : updated to reflect latest obo 1.2 spec

About

This document describes the mapping between the OBO 1.2 file format and the OWL W3C standard ontology language. The mapping is available as an XSLT file [here](#):

- XSLT [http://www.godatabase.org/dev/xml/xsl/oboxml_to_owl.xsl]

Documentation on the OBO 1.2 file format can be found here:

- OBO-1_2 [http://www.godatabase.org/dev/doc/obo_format_spec.html]

(this spec is not yet finalised)

This Document

This document can be viewed online:

- Text [<http://www.godatabase.org/dev/doc/mapping-obo-to-owl.txt>]
- HTML [<http://www.godatabase.org/dev/doc/mapping-obo-to-owl.html>]
- PDF [<http://www.godatabase.org/dev/doc/mapping-obo-to-owl.pdf>]

It also available in the go-dev cvs repository; see

- GO-Dev [<http://www.godatabase.org/dev>]

Downloading ontologies in OWL

GO in OWL

OWL downloads of the daily and monthly builds of the GO Database are available from the go-dev website. These are purely syntactic conversions. Currently all OBO ontologies use only a small subset of the OBO formalism, and hence only corresponds to a small subset of OWL.

- GO-Database [<http://www.godatabase.org/dev/database>]

(look for the file marked go_YYYYMM-termdb.owl.gz)

Other OBO ontologies in OWL

Daily exports of most OBO ontologies into OWL can be found here:

- OBO-Download [<http://www.fruitfly.org/~cjm/obo-download>]

Individual OBO ontologies can be downloaded, or the entire set can be downloaded in OWL from the ftp

site above.

This is not a stable URL - this service will soon be subsumed into the services provided by the new National Center for Biomedical Ontologies (NCBiO [<http://www.ncbo.us>]).

Inferred DL definitions

Semantic conversions, including automated inference of class definitions are available from the Obol page. Note that these are automated, highly experimental and subject to change.

- Obol [<http://www.fruitfly.org/~cjm/obol>]

Status of mapping

Not complete. Reasonably stable - any new changes introduced from here on should be backwards compatible

need to add more AnnotationProperties to capture missing aspects of the OBO 1.2 model, eg synonyms.

Background Material

- W3C OWL Recommendations [<http://www.w3.org/2004/01/sws-pressrelease>]
- W3C OWL Guide [<http://www.w3.org/TR/2003/PR-owl-guide-20031215/>]
- GO Formats [<http://www.geneontology.org/GO.format.shtml>]
- OBO 1.2 Format [http://www.godatabase.org/dev/doc/obo_format_spec.html]
- OBO 1.0 DTD [<http://www.godatabase.org/dev/xml/dtd/obo-xml.dtd>]

Identifiers

OWL is layered over RDF/RDFS. The OBO identifier model is consistent with the RDF/XML identifier model.

An OBO identifier consists of an idspace and a local ID (eg GO and 0008045). These are normally flattened using a colon separator, e.g. GO:0008045. An idspace can have both short and long forms. The short form would be unique within OBO, the long form would be a guaranteed globally unique URI prefix.

The current OWL mapping simply substitutes the : for a _ in the ID, and prepends a generic URI prefix.

In future this will be changed. The RDF/XML ID will be composed of the idspace URI (corresponding to XML namespace) and the local ID (e.g. <http://www.geneontology.org/GO#0008045>, or whatever URI scheme we choose to use). The idspace short form (eg GO) will be used as the XML namespace qname.

OBO Terms

An OBO term is signified with the stanza [Term] in the OBO flatfile. In Obo-XML the corresponding element is `<term>`

All OBO terms correspond to `<owl:Class>`

OBO Typedefs

In an OBO text file these have the [Typedef] stanza. In OBO-XML these have a `<typedef>` element.

Mapping Typedefs is much the same as mapping Terms

Typedefs generally correspond to `<owl:ObjectProperty>` or `<owl:TransitiveProperty>` depending on whether the obo file has the tag *is_transitive* set

Typedef *is_a*: tags are mapped to `<rdfs:subPropertyOf>`

OBO typedefs in which the range is an xsd datatype should be mapped to a `owl:DatatypeProperty`. This is not currently done in the mapping (this extension to the obo model was only introduced in 1.2 and is not yet in widespread use)

The OBO 1.2 format has introduced tags corresponding to the following owl or rdfs properties:

- `rdfs:domain` (domain tag)
- `rdfs:range` (range tag)
- `owl:inverseOf` (inverse_of tag)

These are currently ignored in the existing mapping implementation - this will be fixed in a future version.

The OBO text and xml formats are extensible, and we may decide to add extra tags to allow us to represent these things.

The following OBO typedef stanza tags have no OWL equivalent:

- *is_cyclic*
- *is_symmetric*
- *is_antisymmetric*

OBO Names

An OBO term must have a *name:* tag. This is the `<name>` element in OBO-XML

OBO names are mapped to `rdfs:label`, which is a sub-element of `<owl:Class>`

OBO Natural Language Definitions

OBO terms have optional human-readable text definitions; this is the *def:* tag and is represented as `<definition>` in OBO-XML

definitions are currently mapped to `<rdfs:comment>`

Note that this is not ideal - a comment is more general than a definition. This will be fixed in future to use an `owl:AnnotationProperty`

OBO Metadata

OBO ontologies typically have a lot of metadata associated with each term. This includes comments, the text definition, as well references for the definition. Terms can also have typed synonyms, and terms can have analogs (not necessarily equivalent) in other ontologies. Terms can be marked obsolete yet remain "in the ontology".

Currently the mapping excludes much of the obo-format metadata.

In future, we will introduce `AnnotationProperties` for these

OBO Instances

Obo-format allows instance data to be specified. This is not yet in use. The mapping to OWL has still to be specified here and in the XSLT.

Subclassing

The tag *is_a:* is part of the OBO specification; it appears within the [Term] stanza, and corresponds to the element `<is_a>` in OBO-XML. The value of this tag is the Id for the parent term

is_a is mapped to `<rdfs:subClassOf>`, with the Id for the parent term going in the attribute `rdf:resource` (with `:` mapped to `_`)

If an *is_a:* tag or element appears in an OBO typedef, this is mapped to `<rdfs:subPropertyOf>`

Relationships

Relationships other than the axiomatic *is_a* in OBO are represented with the *relationship:* tag in OBO, and corresponds to `<relationship>` in OBO-XML

Relationships have two sub-fields: the relationship type, and the object of the relationship (the subject of the relationship is the enclosing term)

In OBO-XML, these fields are mapped to the elements `<type>` and `<to>` respectively (type is the name of the relationship type, to is the parent ID)

In OWL, OBO relationships other than *is_a* are mapped to subclasses of restrictions that are existentially qualified (`someValuesFrom`).

Currently the OBO format does not make a distinction between existentially or universally qualified relationships. We assume they are all existentially qualified. This is in line with the definition provided in:

- OBO-Relations [<http://obo.sourceforge.net/relationship>]

As an example, the OWL representation of the relationship *part_of* *GO:0005933* looks like this:

```
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource='part_of'></owl:onProperty>
    <owl:someValuesFrom rdf:resource='GO_0005933'></owl:someValuesFrom>
  </owl:Restriction>
</rdfs:subClassOf>
```

These relations must be defined - see documentation on mapping [Typedefs]

We reserve a 3rd subfield of the relationship: tag for these kind of extensions; this will probably be a { } or [] enclosed list of axiomatic qualifiers and tag-values (i.e. they will be part of the OBO spec). These qualifiers and tag-values will most likely correspond to the OWL equivalents (eg *minCardinality*., *cardinality*., etc). This has not yet been finalised.

Class definitions

Most OBO ontologies only provide necessary conditions for class membership (i.e. the relationship: tag). The OBO 1.2 format also allows for necessary and sufficient conditions using the *intersection_of*: tag, which maps to *owl:intersectionOf*. This is commonly referred to in the OBO community as *cross-products*

As part of the Obol project, we are curating DL style definitions for GO biological process classes that involve types of cells. Information will be posted here:

- Obol [<http://www.fruitfly.org/~cjm/obol>]

You can also find uncurated automatically generated DL definitions for GO and the plant ontology at the above URL (in both OBO and OWL formats)

In practice we will usually have *aristotelian definitions* in OBO: these are terms defined by a *core* or *principal* term and one or more *differentiating characteristics* (also known as defining *species* from *genus* and *differentiae*, although these terms are overloaded in biology!)

Other OWL constructs

We will add mappings for unions, disjointness axioms as required. Currently no OBO ontologies have these.

Tools

Currently the go-perl library contains scripts and modules for converting between OBO,GO-legacy,

OBO-XML and OWL.

- GO-Perl [<http://search.cpan.org/~cmungall/go-perl>]

go-perl uses the oboxml-to-owl XSL file.

OBO-Edit also has an OWL writer, although its mapping may differ from the one provided here

See Also

The details of the mapping implemented here are heavily indebted to the follow people and groups, (although mistakes and shortcomings are our own):

Stuart Aitken of the Edinburgh AIAI group has mapped GO to OWL

- AIAI [<http://www.aiai.ed.ac.uk/resources/go>]

Claude Pasquier has independently mapped GO to OWL

- ClaudePasquier [http://bioinfo.unice.fr/equipe/Claude.Pasquier/current_work.htm]

Mikel Egana Aranguren's thesis:

- PDF [<http://www.sindominio.net/~pik/thesis.pdf>]

The Manchester GONG project:

- GONG [<http://gong.man.ac.uk/>]

See this thread on obo-discuss:

- Obo-discuss [http://sourceforge.net/mailarchive/forum.php?thread_id=6443655&forum_id=33155]

FAQ

Why use OBO (text or XML) rather than OWL as the primary format?

- not (so) human readable
- not (so) human editable

- bulkier
- not well suited for CVS management
- too complex for typical bioinformatics applications to handle; most bio apps just want to get at basic DAG structure
- issues with metaclasses, OWL Full
- RDF mapping ugly (mapping IDs)
- RDF not easily amenable to XML tools (eg XSLT)
- we can convert back and forth to OWL anyway, using the XSLT, so we have interoperability
- Using OWL commits us to using DLs. We think it is better to explore DLs such as OWL using this transformation spec/tools, and retain the option of not using them (instead using simple DAGs, or a prototype-based formalism, or any other options)

ACKNOWLEDGEMENTS

Thanks to John Day-Richter, Olivier Dugus, Phillip Lord, Stuart Aitken, Claude Pasquier, Chris Wroe and Mikel Egana Aranguren for comments