MIREOT: the Minimum Information to Reference an External Ontology Term

Author One[1,2], Author Two[2], Author Three[1,2]
[1]One Location, USA, [2]Another Location, UK

Abstract

While the Web Ontology Language (OWL) provides a mechanism to import ontologies, this mechanism is not always suitable. First, given the current state of editing tools and the issues they have working with large ontologies, direct OWL imports have proven impractical for day-to-day development. Second, ontologies chosen for integration may be under active development and not aligned with the chosen development methodology. Importing heterogenous ontologies in their entirety may lead to inconsistencies or unintended inferences. In this paper we propose a set of guidelines for importing required terms from an external artefact into a target ontology. We describe the guidelines, their implementation, present some examples of application, and outline future work and extensions.

Introduction

The Web Ontology Language (OWL)[1] provides a built-in mechanism to import ontologies (*owl:imports*). The *owl:imports* mechanism has proved insufficient for the needs of the authors during development of Ontology of Biomedical Investigations (OBI)[2], a large integrative ontology for the description of life-science and clinical investigations. Limitations in the currently available tools and reasoners can make the implementation of the *owl:imports* mechanism problematic. For example, most current OWL tools can neither load nor reason over very large ontologies, such as the NCBI Taxonomy[3] or the Foundational Model of Anatomy[4], making direct OWL imports of such ontologies impractical.

Ontologies are typically a knowledge resource that must be not only maintained but constantly updated. This is evident within the bio-ontology domain, where the codification of knowledge must keep pace with scientific knowledge discovery. The flux of knowledge within an ontology can also present issues when trying to import or integrate different resources. The individual resources may have been constructed using different desing principles, which may not align. These issues of dynamic development, and different design principles could lead to inconsistencies or unintended inferences, when integrated.

To address these issues, we have developed a set of guidelines for importing classes from multiple ontology resources, avoiding the overhead of importing the complete ontology from which the terms derive. These guidelines are called the Minimum Information to Reference an External Ontology Term (MIREOT).

The MIREOT guidelines were created to aid the development of OBI[2]. OBI uses the Basic Formal Ontology (BFO)[5] as upper-level ontology and is part of the Open Biomedical Ontologies (OBO) Foundry[6]. One of the fundamental principles of the OBO Foundry is to reuse, where sensible, existing ontology resources, therefore avoiding duplication of effort and ensuring orthogonality.

The MIREOT guidelines describes a methodology for the import of external terms from ontologies irrespective of the chosen upper ontology, using other OWL flavours or even differrent syntax formats such as the OBO-Format.

Policy

In deciding upon a minimum unit of import, our first step was to consider the practices of other ontologies. The practice of the Gene Ontology (GO)[7] is that the intended meaning of classes remains stable. The repair or reorganization of an ontology should not change the intended meaning of individual classes, but rather provide a better expression of the logical relations between them. When the class definition changes meaning, the class is deprecated rather than creating a situation where a class's semantics changes[8]. We can therefore consider a class as stable, in isolation from the rest of the ontology.

The minimum amount of information needed to *reference* an external class is the ontology URI and the external class's URI. Generally, these items remain stable and can be used to unambiguously reference the external class from within the importing target ontology. The minimum amount of information to *integrate* this class is its position in the hierarchy, specifically the URI of the parent class from the target ontology.

Taken together, the following minimal set is enough to consistently reference an external class:

external source ontology URI The logical URI for the ontology containing the external to be imported.

external source term URI The logical URI for the specific term to import.

target direct superclass URI The logical URI for the class in the target ontology that the external class should be imported under.

To ease development of the target ontology we also recommend, although do not require, additional metadata about the external class we wish to import, such as their label and definition.

Implementation

An implementation of the MIREOT guidelines was performed in the context of the OBI project, and involved a two-step process:

1. gather the minimum information for the external class, and
2. use this minimum information to fetch additional elements, like labels and definitions.

Once the external class is identified for import, the first step is to gather the corresponding minimum information set. This set is stored in a file that we arbitrarily call *external.owl*. A perl script `add-to-external.pl` **??**was used to automatically append the minimum information set, for multiple external classes to the *external.owl* file. The perl script, takes as arguments the identifiers of the external class to be imported and its parent class in the target hierarchy, in this case in the OBI hierarchy. In addition, a mapping mechanism between the prefix used in the identifier and the external source ontology URI is built into the script.

Curators therefore need only specify the ID of the external class to import and the ID of the class it should be imported under, within the target ontology. For example:

```
perl add-to-external.pl CHEBI:23367 IAO:0000018
```

will add the class molecular entities (CHEBI:23367) as subclass of the class material entity (IAO:0000018), and set the source ontology URI as `http://purl.org/obo/owl/CHEBI`.

Once the minimum information has been collected and stored in the *external.owl* file, additional elements are obtained programatically via SPARQL[9] CONSTRUCT queries, as described in (Figure 1).

These queries specify which extra information about the class to gather, such as the definition and

```
prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefix owl: <http://www.w3.org/2002/07/owl#>
prefix obi: <http://purl.obofoundry.org/obo/>
prefix tax: <http://purl.org/obo/owl/NCBITaxon#NCBITaxon>
prefix obo: <http://www.geneontology.org/formats/oboInOwl#>
prefix iao: <http://purl.obofoundry.org/obo/>

construct
{
  _ID_GOES_HERE_ rdf:type owl:Class.
  _ID_GOES_HERE_ alias:preferredTerm ?label.
  _ID_GOES_HERE_ rdfs:label ?label.
  _ID_GOES_HERE_ alias:definition ?definition.
}
where
{
  {   _ID_GOES_HERE_ rdfs:label ?label. }
  UNION
  {   _ID_GOES_HERE_ obo:hasDefinition ?blank.
      ?blank rdfs:label ?definition}
}
```

Figure 1: Template SPARQL query. For convenience, we use alias:preferredTerm and alias:definition to reference our annotations properties IAO_0000111 and IAO_0000115 respectively

preferred label, and how to map these into the corresponding OBI annotation properties.

For example, in the current OWL rendering of OBO-Format, definitions are individuals and the rdfs:label of that individual records the text of the definition. The query (Figure 1) will map the `rdfs:label` of the `oboInOwl:Definition` instance into the value of the `obi:definition` property. Within the OBI implementation of MIREOT guidelines, only annotation properties which map directly to our own metadata **??** are mapped: new properties are not created. Additional metadata is integrated in accordance with the agreed minimal metadata policy (***Add URL as long minimal metadata is not out), *e.g.*, curation status annotation property, definition editor or definition source. The external term is directly imported from the external resource, with the status and definition as defined by the external resource.

A script iterates through the minimum information stored in *external.owl*. Depending on the external ontology, it then selects the correct SPARQL template and substitutes the relevant ID. The queries are executed against the Neurocommons SPARQL endpoint[10].

This supplementary information, which is prone to change as the source ontologies evolve, is stored in a second file, `externalDerived.owl` and generated via a script. This file can be removed on a regular basis (e.g. for a slim release), and rebuilt via script based on external.owl, allowing us to keep imported information up-to-date. The two files, `external.owl`

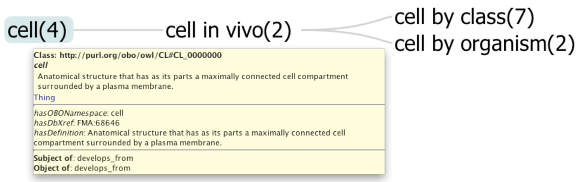

Figure 2: The `cell` class in the Cell Type Ontology. The classes `cell by class` and `cell by organism` are examples of classes that did not conform to the OBI development principles and therefore were not suitable for integration within OBI.

and `externalDerived.owl`, are then imported by `obi.owl`, providing the necessary information to OBI editors while at the same time keeping it isolated from the core OBI ontology.

The following sections present differing use-cases identified and implemented withing OBI describing the application of the MERIOT guidelines utilises classes from the OBO-Format version of theCell Type (CL) ontology and the ***format of the NCBI Taxonomy.

Use Case One - Cell class

As an example, we replaced the OBI class $Cell$ with that from the CL ontology[11] (Figure 2). The CL is part of the OBO Foundry effort, and we would like to use the *cell* class as defined by this resource, instead of creating our own duplicated class.

Following the MIREOT guidelines, the minimum information required to integrate the CL class `cell` within OBI is:

1. *external source ontology URI*: the ontology from which the term is imported: `http://purl.org/obo/owl/CL`
2. *source exernal class URI*: the URI of *cell*: `http://purl.org/obo/owl/CL#CL_0000000`
3. *target parent URI*: the position of *cell* in the OBI hierarchy: as a subclass of material entity

The imported *cell* class can be used within axiomatisations as with any other OBI class: for example, using Manchester OWL syntax, the process $electroporation$ is defined as:

```
is_a cell permeabilzation
has_specified_input some cell
has_specified_output some
  (cell and has_quality some electroporated))
utilizes_device some power supply
```

It can also be subclassed, either by other imported classes or by OBI classes.

Use Case Two - taxonomic information

The `cell` use-case highlights what is likey to be the most common import scenario, i.e. a simple one-to-one mapping towards an external class. However, in some cases the class import may be more complex. To account for this the MIREOT mechanism has been devised to be extensible, and account for additional types of information, as required by a particular use-case.

The following example describes a more complex query that can be perfomed on classes that have been imported from an external resource. A common use of OBI is that it can be used to annotate data from biological experiments. In the principle of re-using pre-exsiting resources, OBI uses the required parts of the NCBI taxonomy (**REF?***) to describe species of organisms.

We can easily imagine that somebody would want to query a dataset asking the question "give me all experiments in mammals". In this case, we would need to know that human and mouse are subclasses (even indirect) of mammals in the NCBI taxonomy. Therefore, when mapping towards an NCBI term, it is needed to get the class itself and the parent classes up to the root class of the NCBI taxonomy. This is achieved by a specific SPARQL query, which retrieves all direct parent classes up to one of a set of upper-level classes in the taxonomy.As per the mechanism described above, the minimum information about the imported external class (*e.g.*, Mus musculus) is defined in *external.owl*, whereas the additional information (rank information - genus, kingdom, phylum, etc.) is stored in *externalDerived.owl*.

Discussion

The MIREOT guidelines define an import mechanism at the cost of partial consistency checks on the imported code, whereas complete importing via `owl:imports` provides a mechanism with complete consistency checking.

By copying only parts of an ontology there is the risk that inferences drawn may be incomplete or incorrect: correct inference using the external classes is only guaranteed if the full ontologies are imported. When deciding to import an external term we review the textual definition and, if needed, talk with the original editor. As we are importing from OBO Foundry ontologies we have a community process for monitoring change, a shared understanding of the basics of our domain, and the intention to eventually share the same upper-level ontology. Therefore, we expect that terms will be deprecated if there is a significant change in meaning, and are flexible enough to adjust and update our import of terms as the other ontologies start enhancing their logical definitions.

Another consideration using this approach is the status of assertions made on external terms. In adding axioms such as the subclass axiom when importing the external term, the aim is to only assert true statements about the terms. We can also add extra restrictions to these imported classes: for example in OBI, cell is the bearer of the role reagent role or specimen role. These additional restrictions should be stored in the target ontology: the `external.owl` and `externalDerived.owl` are meant to include only the imported information.

We anticipate that some of these statements may migrate to the source ontologies at some point in the future; a fruit of the collaborative nature of OBO Foundry ontology development.

Future work

The current implementation of the MIREOT guidelines is applied externally to an OWL IDE, such as Protégé via command-line scripts. Ideally, a Protégé[12] plugin could be developed to improve the interaction between the curators and the tool and the implementation of the MIREOT guidelines.

In the future, we also expect to provide an option in the OBI distribution that replaces external.owl with `imports.owl`, a file of imports statements generated by extracting the ontology URIs mentioned in external.owl.

The MIREOT guidelines are currently being implemented by other ontologies and we ultimately hope that combined feedback will allow us to perfect the mechanism.

Acknowledgements

In memory of our friend and colleague William Bug, Ontological Engineer.

The OBI consortium is (in alphabetical order): Ryan Brinkman, Bill Bug, Helen Causton, Kevin Clancy, Christian Cocos, Mélanie Courtot, Dirk Derom, Eric Deutsch, Liju Fan, Dawn Field, Jennifer Fostel, Gilberto Fragoso, Frank Gibson, Tanya Gray, Jason Greenbaum, Pierre Grenon, Jeff Grethe, Yongqun He, Mervi Heiskanen, Tina Hernandez-Boussard, Philip Lord, Allyson Lister, James Malone, Elisabetta Manduchi, Luisa Montecchi, Norman Morrison, Chris Mungall, Helen Parkinson, Bjoern Peters, Matthew Pocock, Philippe Rocca-Serra, Daniel Rubin, Alan Ruttenberg, Susanna-Assunta Sansone, Richard Scheuermann, Daniel Schober, Barry Smith, Larisa Soldatova, Holger Stenzhorn, Chris Stoeckert, Chris Taylor, John Westbrook, Joe White, Trish Whetzel, Stefan Wiemann, Jie Zheng.

References

[1] Web Ontology Language (OWL), `http://www.w3.org/2004/OWL/`.

[2] OBI Ontology, `http://purl.obofoundry.org/obo/obi`.

[3] D. L. Wheeler, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, W. Helmberg, D. L. Kenton, O. Khovayko, D. J. Lipman, T. L. Madden, D. R. Maglott, J. Ostell, J. U. Pontius, K. D. Pruitt, G. D. Schuler, L. M. Schriml, E. Sequeira, S. T. Sherry, K. Sirotkin, G. Starchenko, T. O. Suzek, R. Tatusov, T. A. Tatusova, L. Wagner, and E. Yaschenko. Database resources of the national center for biotechnology information. *Nucleic acids research*, 33(Database issue):D39–45, Jan 1 2005. LR: 20081120; JID: 0411011; OID: NLM: PMC540016; ppublish.

[4] C. Golbreich, S. Zhang, and O. Bodenreider. The foundational model of anatomy in owl: Experience and perspectives. *Web semantics (Online)*, 4(3):181–195, 2006.

[5] P. Grenon, B. Smith, and L. Goldberg. Biodynamic ontology: applying bfo in the biomedical domain. *Studies in health technology and informatics*, 102:20–38, 2004.

[6] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, OBI Consortium, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S. A. Sansone, R. H. Scheuermann, N. Shah, P. L. Whetzel, and S. Lewis. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255, Nov 2007. PUBM: Print; JID: 9604648; ppublish.

[7] Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(90001):D258–D261, 01/01/ 2004.

[8] Go editorial style guide - http://www.geneontology.org/GO.usage.shtml.

[9] SPARQL Query Language for RDF - http://www.w3.org/TR/rdf-sparql-query/.

[10] Neurocommons sparql endpoint - http://sparql.neurocommons.org/.

[11] J. Bard, S. Y. Rhee, and M. Ashburner. An ontology for cell types. *Genome biology*, 6(2):R21, 2005.

[12] The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/.