MIREOT: the Minimum Information to Reference an External Ontology Term

Mélanie Courtot[1], **Frank Gibson**[2], **Allyson L. Lister**[3], **James Malone**[4], **Daniel Schober**[4,5], **Ryan R. Brinkman**[1], **Alan Ruttenberg**[6]

[1]BC Cancer Agency, Vancouver, BC, Canada, [2]Abcam plc, 332 Cambridge Science Park, Cambridge, CB4 OWN, UK, [3]CISBAN and School of Computing Science, Newcastle University, Newcastle upon Tyne, UK, [4]The European Bioinformatics Institute, Cambridge, CB101SD, UK, [5]Institute of Medical Biometry and Medical Informatics (IMBI), University Medical Center, 70104 Freiburg, Germany, [6]Science Commons, Cambridge, MA, USA

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 sometimes proven impractical for day-to-day development. Second, ontologies chosen for integration may be under active development and not aligned with the chosen design principles. Importing heterogeneous 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 resource into a target ontology. We describe the guidelines, their implementation, present some examples of application, and outline future work and extensions.

Introduction

While the **OWL!** (**OWL!**)[?] provides a mechanism to import ontologies (*owl:imports*), current limitations in tools and reasoners can sometimes make such a solution impractical on a day-to-day basis. First, some OWL tools (*e.g.*, Protégé, SWOOP) can neither load nor reason over very large ontologies, such as the NCBI Taxonomy[?] or the Foundational Model of Anatomy[?], making direct **OWL!** imports of such ontologies impractical. Second, different resources may have been constructed using different design principles, which may not align. Importing such ontologies as a whole could lead to inconsistencies or unintended inferences.

To address these issues, we have developed a set of guidelines for importing terms from multiple ontology resources, avoiding the overhead of importing the complete ontology from which the terms derive.

The **MIREOT!** (**MIREOT!**) guidelines were created to aid the development of the **OBI!** (**OBI!**)[?]. **OBI!** uses the **BFO!** (**BFO!**)[?] as upper-level ontology and is part of the **OBO!** (**OBO!**) Foundry[?]. 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. **MIREOT!** allows us to do so by providing a way to import external terms from ontologies not yet using BFO as an upper ontology, or not yet using OWL DL.

Policy

In deciding upon a minimum unit of import, our first step was to consider the practice of other ontologies. For example, in the Gene Ontology (GO[?]), the intended denotation of classes remain stable. Even when the ontology is repaired or reorganized, the effects of such changes do not change the intended meaning of individual terms. Rather the changes are towards more carefully expressing the logical relations between them. When a term's definition changes meaning, the term is deprecated[?]. We can therefore consider a term as stable, in isolation from the rest of the ontology, and use terms (*i.e.*, individual classes in isolation from the ontology) as basic unit of import. The current implementation of **MIREOT!** has been limited to import of terms from other Foundry ontologies, which adhere to a similar deprecation policy.

The minimum amount of information needed to *reference* an external class is the source ontology URI and the external term'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 its direct superclass in the target ontology.

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

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

source term URI The logical URI of the specific term to import.

target direct superclass URI The logical URI of

the direct asserted superclass in the target ontology.

To ease development of the target ontology we also recommend, although do not require, that additional information about the external class be added such as its label and textual definition.

Implementation

An implementation of the **MIREOT!** guidelines was performed in the context of the **OBI!** project, and can be decomposed into a two-step process:

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

Once the external term is identified for import, the first step is to gather the corresponding minimum information set.

This set is stored in a file that we call *external.owl*[1]. A Perl script, *add-to-external.pl*[?] is used to automatically append the minimum information set to the *external.owl* file. This 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.

Additional elements can be obtained programmatically via SPARQL[?] CONSTRUCT queries (Figure **??**). These queries[2] specify which extra information about the class to gather, such as the definition and preferred label, and how to map these into the corresponding OBI annotation properties.

For example, in the current **OWL!** rendering of **OBO!** files, definitions are individuals and the rdfs:label of those individuals record the text of the definitions. Within the **OBI!** implementation of the **MIREOT!** guidelines, only annotation properties which map directly to our own metadata are mapped: new properties (*e.g.*, curation status annotation property, definition editor or definition source), if not specified in the source ontology, are not created.

The external term is directly imported from the external resource, with the status and definition as defined by the external resource. Finally, a script, *create-external-derived.lisp*[?], iterates through the minimum information stored in *external.owl*. Depending on the source ontology URI of each of our imported terms, it then selects the correct SPARQL template and substitutes the relevant ID. The queries are then executed against the Neurocommons SPARQL endpoint[?].

```
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 obo: <http://www.geneontology.org/formats/oboInOwl#>

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

This supplementary information, which is prone to change as the source ontologies evolve, is stored in a second file, *externalDerived.owl*[3]. This file can be removed on a regular basis, *e.g.*, before release of OBI. It is then rebuilt via script based on *external.owl*, allowing us to keep imported information up-to-date. The two files, *external.owl* and *externalDerived.owl*, are then imported by *obi.owl*, providing the necessary information to OBI editors while at the same time keeping it independent from OBI proper classes.

In the following sections we present two different cases of application of the **MIREOT!** guidelines.

Use Case One - Cell class

We replaced the **OBI!** class *Cell* with that from the **CL!** (**CL!**) ontology[?]. **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. The following invocation of the *add-to-external.pl* script:

```
perl add-to-external.pl CL:0000000 IAO:0000018
```

[1] http://tinyurl.com/b7vvlt
[2] http://tinyurl.com/bss9mw
[3] http://tinyurl.com/bmb3f4

will add the class *cell* (CL:0000000) as subclass of the class material entity (IAO:0000018), and set the source ontology URI as `http://purl.org/obo/owl/CL`. Once imported, the *cell* class can be used as would be any other OBI class. For example, the process "electroporation" is defined as:

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

More generally, additional axioms may be used to relate members of the class to other entities in the ontology.

Use Case Two - taxonomic information

The `cell` use-case highlights what is likely to be the most common import scenario (*i.e.*, a simple import of one external term, making it available for direct use in the target ontology. However, in some cases, we may require more, and to account for this **MIREOT!** has been devised to be flexible.

OBI currently uses the NCBI taxonomy for its species terms. 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 all its superclasses up to one of a set of top-level classes in the taxonomy.

When the *create-external-derived.lisp* script parses the *external.owl* file and encounters an NCBI taxonomy ID, it will invoke a specific SPARQL query (cf figure **??**). 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 (*e.g.*, rank information - genus, kingdom, phylum) is stored in *externalDerived.owl*.

```
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>

construct
 { ?super rdf:type owl:Class.
   ?super rdfs:subClassOf ?parent.
   ?super alias:preferredTerm ?label.
   ?super rdfs:label ?label.
 }
 where
  {
     {  #direct superclass.
        { _ID_GOES_HERE_ rdfs:subClassOf ?super. }
        { ?super rdfs:subClassOf ?parent.
          ?super rdfs:label ?label.
        }
      }
   UNION
        # now harvest the class annotations
      { ?super rdfs:subClassOf ?parent.
        ?super rdfs:label ?label.
FILTER (?super=_ID_GOES_HERE_)
      }
   FILTER (!((?parent=alias:bacteria) || (?parent=alias:eukaryota) ||
     (?parent=alias:viruses) || (?parent=alias:archaea) ||
     (?parent = alias:cellularOrganism) || (?super=alias:bacteria) ||
     (?super=alias:eukaryota) || (?super=alias:viruses) ||
     (?super=alias:archaea) || (?super = alias:cellularOrganism)))
}
```

Figure 2: Template SPARQL query for import from the NCBI taxonomy.

Discussion

The MIREOT standard presents an approach to importing classes from external ontologies that removes the overhead of full ontology imports whilst maintaining a decoupled but usable reference to the external classes. There is a clear tradeoff that MIREOT offers between praticality and full, axiomatic completeness. By importing only the desired parts of an external ontology and 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 and aximoatization is imported. This does present one important advantage which is that it overcomes the obstacle presented by ontologies which are not fully interoperability at present. In the future, efforts such as those endorsed by the OBO Foundry, may offer this complete level of integration by the use of a uniform framework, e.g. the same set of relations and a single upper ontology.

When deciding to import an external term the textual definition is reviewed and, if required, discussion with the original editor is undertaken. For our use case, we are imported from OBO Foundry ontologies where a community process for monitoring change and a shared understanding of the scope of each ontolgoy is in place. Therefore, it is expected 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 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. If additional restrictions are required (for example in OBI, *cell* is the bearer of the role reagent role or

specimen role), those 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 the statements added by the target ontology 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 relies on command-line scripts, making it sometimes uncomfortable to use for our curators. Ideally, a Protégé? 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, like the Vaccine Ontology (VO)?, and we ultimately hope that combined feedback will allow us to perfect the mechanism.

Acknowledgments

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. The authors' work was partially supported by funding from the NIH(R01EB005034), the EC EMERALD project (LSHG-CT-2006-037686), the BBSRC(BB/C008200/1, BB/D524283/1, BB/E025080/1), the EU FP7 DebugIT project (ICT-2007.5.2-217139), the Public Health Agency of Canada / Canadian Institutes of Health Research Inuenza Research Network (PCIRN), and the Michael Smith Foundation for Health Research.

References