MIREOT: the Minimum Information to Reference an External Ontology Term

author1 [a,*], author2 [b], author3 [c], author4 [d], author5 [e], author6 [f,*]

[a] *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

[b] *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

[c] *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

[d] *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

[e] *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

[f] *xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx*

Abstract

While the Web Ontology Language (OWL [1]) 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 are not practical for day-to-day development. Second, the other ontologies used may be under active development and not aligned with the chosen development approach, and importing such ontologies as a whole could lead to inconsistencies or unintended inferences. In this paper we propose a new minimum information standard for importing required terms into an ontology. We suggest an implementation, present some example of application, scripts for automation of the import process and finally some ideas for future work and extensions.

Introduction

While 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, most current OWL tools can neither load nor reason over very large ontologies such as the NCBI Taxonomy [2] or the Foundational Model of Anatomy [3], making direct OWL imports of such ontologies, as a whole, impractical for day-to-day development. Second, other ontologies may be under active development and not aligned with the design of the resource that needs to reference their terms. Importing such ontologies as a whole could lead to inconsistencies or unintended inferences.

To address these issues, we devised a more involved, but more flexible, mechanism.

Our solution is a new standard we call the Minimum Information to Represent an External Ontology Term (MIREOT). MIREOT provides guidelines on importing selected terms without the overhead of importing the complete ontology from which the terms derive.

MIREOT has been implemented during the development of the Ontology for Biomedical Investigations (OBI [4]). OBI uses the Basic Formal Ontology (BFO [5]) as upper-level ontology and is part of the OBO Foundry [6]. One of the fundamental principles of the OBO Foundry is to reuse, where sensible, existing ontology resources, therefore avoiding duplication of efforts and ensuring orthogonality. MIREOT allows the import of external terms from ontologies not 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 practices of other ontologies. The practice of the Gene Ontology (GO [7]) is that the intended meanings 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 [8]. We can therefore consider a term as stable, in isolation from the rest of the ontology, and use terms (i.e. classes) a basic unit of import.

[*] Corresponding authors.
Email addresses: `email1@mail.com` (author1), `email2@mail.com` (author6).

```
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}
}
```

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

The minimum amount of information needed to reference an external class is the ontology URI and the term's URI. Generally, these items remain stable and can be used to unambiguously reference the external class from within the importing ontology. The minimum amount of information to integrate this class is its position in the hierarchy, i.e., what class the imported class is a subclass of.

Taken together, this set (ontology URI, term URI, OBI superclass) are enough to consistently reference an external term. In addition to this minimum information, and in order to ease editing of the ontology, we also want to provide extra information about our imported classes such as their label and definition.

Implementation

The practical implementation of the MIREOT standard has been performed in the context of the OBI project. It involves a two-steps process:

(i) gather the minimum information for external terms
(ii) use this minimum information to fetch additional elements, like labels and definitions.

Once the classes to import have been identified by the curators, the first step is to gather their corresponding minimum information set. This set is stored in a file that we call *external.owl*, either by creating it, or by appending the new information to the pre-exisitng one if some terms were already imported. This file can be edited by hand, but to ease the process of importing external terms, we provide a perl script that can be used in command line to append to the file.

This perl script, add-to-external.pl, takes as arguments the identifiers of the term to be imported and its parent in the OBI hierarchy. A mapping mechanism between the prefix used in the identifier and the source ontology URI is built in the script. Curators therefore need only specify the ID of the term to import and its OBI superclass' ID. 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`.

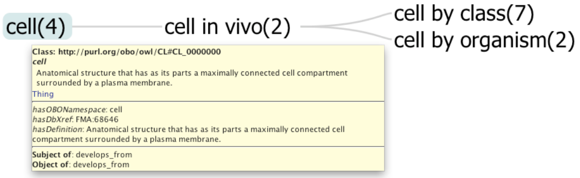
Once the minimum information has been collected and stored in the *external.owl* file, we can use it in order to get additional elements programatically.

In order to do so, we created a set of SPARQL [9] CONSTRUCT queries that we use as template. These queries 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 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. It is worth mentioning that we only map existing annotation properties into our own metadata: we don't create new ones (*e.g.*, curation status annotation property, definition editor or definition source). The term is directly imported from the external resource, with the status and definition as defined by the outside resource.

A script iterates through the minimum information stored in *external.owl*. Depending on the source 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, that we called *externalDerived.owl*.

Finally, the script gathers the results of the queries and creates the *externalDerived.owl* file. This file can be removed on a regular basis, and rebuild via script based on *external.owl*, allowing us to keep imported information up-to-date.



Class: http://purl.org/obo/owl/CL#CL_0000000
cell
Anatomical structure that has as its parts a maximally connected cell compartment surrounded by a plasma membrane.

hasOBONamespace: cell
hasDbXref: FMA:68646
hasDefinition: Anatomical structure that has as its parts a maximally connected cell compartment surrounded by a plasma membrane.

Subject of: develops_from
Object of: develops_from

Fig. 2. The "cell" class in the Cell Type Ontology. "cell by class" and "cell by organism" are examples of classes we would rather not import.

The two files,*external.owl* and *externalDerived.owl*, are then imported by *obi.owl*, providing the combined necessary information to our editors while keeping it isolated from the core OBI ontology.

Use case one - Cell class

As an example, we recently replaced the OBI class cell with that from the OBO Foundry Cell Type (CL) ontology (cf figure 2). The Cell Type Ontology [11] 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 redundant class.

Following the MIREOT guidelines, we identify the minimum information required in this case:

(i) the URI of the term cell: `http://purl.org/obo/owl/CL#CL_0000000`
(ii) the ontology from which the term is imported: `http://purl.org/obo/owl/CL`
(iii) the position of "cell" in the OBI hierarchy: as a subclass of material entity

The imported "cell" class can be used as would be any other OBI class: for example, 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

It is expected that in most cases we will want to import information as described in the cell use case above, i.e. a simple one-to-one mapping towards an external class. However in some cases we might want or need more than that, and the MIREOT mechanism has been devised to be flexible and allow to store other types of information as deemed necessary.

OBI currently uses the NCBI taxonomy for its species terms. Consider the scenario in which we have two experiments, one in human and one in mouse. The files are annotated with the classes human and mouse from OBI, which are in turn individually mapped from the NCBI taxonomy.

We can easily imagine that somebody would want to have a query like "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 the root of the NCBI taxonomy. In this case, we use a specific SPARQL query, which retrieves all direct superclasses up to one of a set of top-level classes in the taxonomy.As per the mechanism described above, the minimum information about the mapped 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 standard is a trade-off between complete consistency checking and heavyweight importing versus lightweight importing but partial consistency checking.

We are aware of and accept that 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 expect to adjust our import of terms as the other ontologies start enhancing their logical definitions.

An other 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 restriction should be stored in the importing ontology file: 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

Even though we aimed at providing an easy mechanism for our editors to use the MIREOT standard, we still rely on command-line scripts. Ideally, a Protégé[12] plugin could be developed to improve the interaction between the curators and the tool. 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 standard is 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, 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.

References

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

[2] 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.

[3] 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.

[4] OBI Ontology, `http://purl.obfoundry.org/obo/obi`.

[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/`.