Applied Ontology 0 (2009) 1–0
IOS Press

MIREOT: the Minimum Information to Reference an External Ontology Term

Mélanie Courtot [a,*], Frank Gibson [b], Allyson L. Lister [c], James Malone [d], Daniel Schober [e], Ryan R. Brinkman [a] and Alan Ruttenberg [f]

[a] *BC Cancer Agency, Vancouver, BC, Canada*
E-mail: mcourtot@gmail.com, rbrinkman@bccrc.ca
[b] *Abcam plc, 332 Cambridge Science Park, Cambridge, CB4 OWN, UK*
E-mail: fgibson@gmail.com
[c] *CISBAN and School of Computing Science, Newcastle University, Newcastle upon Tyne, UK*
E-mail: a.l.lister@newcastle.ac.uk
[d] *The European Bioinformatics Institute, Cambridge, CB10 1SD, UK*
E-mail: malone@ebi.ac.uk
[e] *Institute of Medical Biometry and Medical Informatics (IMBI), University Medical Center, 70104 Freiburg, Germany*
E-mail: schober@imbi.uni-freiburg.de
[f] *Science Commons, Cambridge, MA, USA*
E-mail: alanruttenberg@gmail.com

Abstract. While the Web Ontology Language (OWL) provides a mechanism to import ontologies, this mechanism is not always suitable. First, as current editing tools have issues 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 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 and an implementation, present some examples of application, and outline future work and extensions.

Keywords: ontology import, data integration, development tool

1. Introduction

Ability to share and reuse existing ontological resources is a key factor when developing a new ontology. For example when developing an ontology related to the biomedical domain, it may be useful to include terms from the Gene Ontology (GO)(13) to represent biological processes or from Phenotypic Quality Ontology (PATO)(27) to represent properties of entities. Ontologies such as GO and PATO are built collaboratively by communities of experts and are the products of substantial effort. Redoing this work instead of reusing it would be a duplication of the development effort but also of the resulting ontologies. It would also result in different projects having different identifiers to denote the same entity, which would require post-hoc, error prone, identifier mapping systems to enable data integration. While it seems that building upon existing vocabularies is the best way to proceed, ontology developers are faced with some difficulties when actually trying to do so. The easiest way to integrate an existing body of work is to rely on the Web Ontology Language (OWL) (6) mechanism *owl:imports*, which imports the external resource as a whole. However current limitations in tools and reasoners can sometimes make such a solution impractical on a day-to-day basis. First, popular OWL tools (*e.g.*, Protégé, Pellet) can neither load nor reason over very large ontologies such as the NCBI Taxonomy (7) or the Foundational Model

[*] Corresponding author: Mélanie Courtot, BC Cancer Agency, Vancouver, BC, Canada.

of Anatomy (8), making direct OWL imports of such ontologies impractical. Second, target ontologies may have been constructed using design principles which may not align with the practice of the importing ontology. Importing such ontologies as a whole could lead to inconsistencies or unintended inferences. Other import options are possible, for instance using software that extracts a *module* (1) of the external ontology. A module can be seen as a fragment of an ontology that, when imported by an other ontology, allows the same inferences to be drawn with respect to the classes of interest as if the whole ontology had been imported. This solution allows developers to pick only pieces of the source ontology (and thus overcome size issues) without losing any reasoning power. However, if a so extracted module is to be useful the external ontology needs to be structured in a way that is compatible with the importing ontology (for example, using the same upper ontology), and the logical axioms need to be accurate. This is not always the case at the current stage of development of some ontologies. For example, during the development of the Ontology of Biomedical Investigations (OBI)(9), importing the root class of the Common Anatomy Reference Ontology (CARO)(12) was not desired as its definition intersected multiple classes in OBI that was not considered useful.

In addition, although software that extracts *modules* are available, most are only in early stages of development.

We tried several modularization tools (2) (3) (4) (5). All of them discarded annotations, resulting in modules containing only the class declarations, and no annotation properties, such as labels or definitions. We also experienced crashes on large ontologies (with varying sizes depending on the tool considered: for example we were able to load ChEBI (28) with SWOOP but not with Protégé 3.4). One tool had undocumented assumptions about the form of URIs used as class names and therefore extracted empty modules. Finally, tools that were able to extract modules either extracted a single term or a large number of them (depending on the arguments passed), as they try and approximate a module without discarding potentially useful information. These large modules undermine the goal having imports of a manageable size. Our conclusion is that the current ontology tool set is in early stages of development and, though promising, cannot be used as is.

To address these issues we 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 Minimum Information to Reference an External Ontology Term (MIREOT) guidelines were created to aid the development of the OBI. OBI uses the Basic Formal Ontology (BFO) (10) as upper-level ontology and has been submitted for inclusion in the Open Biomedical Ontologies (OBO) Foundry (11). 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 selectively import external terms from ontologies even if they do not use our chosen upper ontology or even OWL DL.

2. Policy

In deciding upon a minimum unit of import, our first step was to consider the practice of other ontologies. For example, in the GO, the intended denotation of classes remain stable such that 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 (14). 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 ontologies that aspire to the Foundry, and so adhere to a similar deprecation policy.

The minimum amount of information needed to reference an external class is the source ontology URI (*i.e.*, where the term comes from) and the external term's URI (*i.e.*, the identifier for this term). 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 desired position in the hierarchy, specifically the URI of its direct superclass in the target ontology (*i.e.*, under which class the term is asserted)

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 importing ontology.

To ease development of the importing ontology we also recommend, although do not require, that additional information about the external class be added, such as its label and textual definition, or any other kind of information that may be deemed useful by the ontology developers. This additional information, when appropriate, is mapped into the importing ontology's annotation properties.

To keep this information up-to-date, we decided to store it in a separate file that can be removed and rebuilt on a regular basis.

3. 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*. (All our scripts and files are available under the OBI Subversion Repository (15)). In the current implementation, a Perl script, *add-to-external.pl*, can be used to append the minimum information set to the *external.owl* file, or the information can be entered in an ontology editing tool. In the future we anticipate this operation will have specific support within editing tools.

The script takes as arguments the identifier of the external class to be imported and its parent class in the target hierarchy. In addition, a mapping 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.

In the current implementation, additional elements can be obtained programmatically via SPARQL(16) CONSTRUCT queries, as described in Figure 1. These queries(31) specify, for each source ontology, 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. However as module extraction technology matures, and where appropriate, additional elements could be retrieved using such technology.

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, the value of the rdfs:label of the oboInOwl:Definition will be set to the value of iao:definition. Only annotation properties which map directly to the target ontology's own metadata are copied; new properties, if not specified in the source ontology, are not created. The external term's property values are copied *"as-is"* from 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 OBO SPARQL endpoint(18; 19).

```
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}
}
```

Fig. 1. Template SPARQL query. For convenience, we use alias:preferredTerm and alias:definition to reference our annotations properties IAO_0000111 and IAO_0000115 (17) respectively. The _ID_GOES_HERE_ pattern will be replaced by the script when building the CONSTRUCT query.

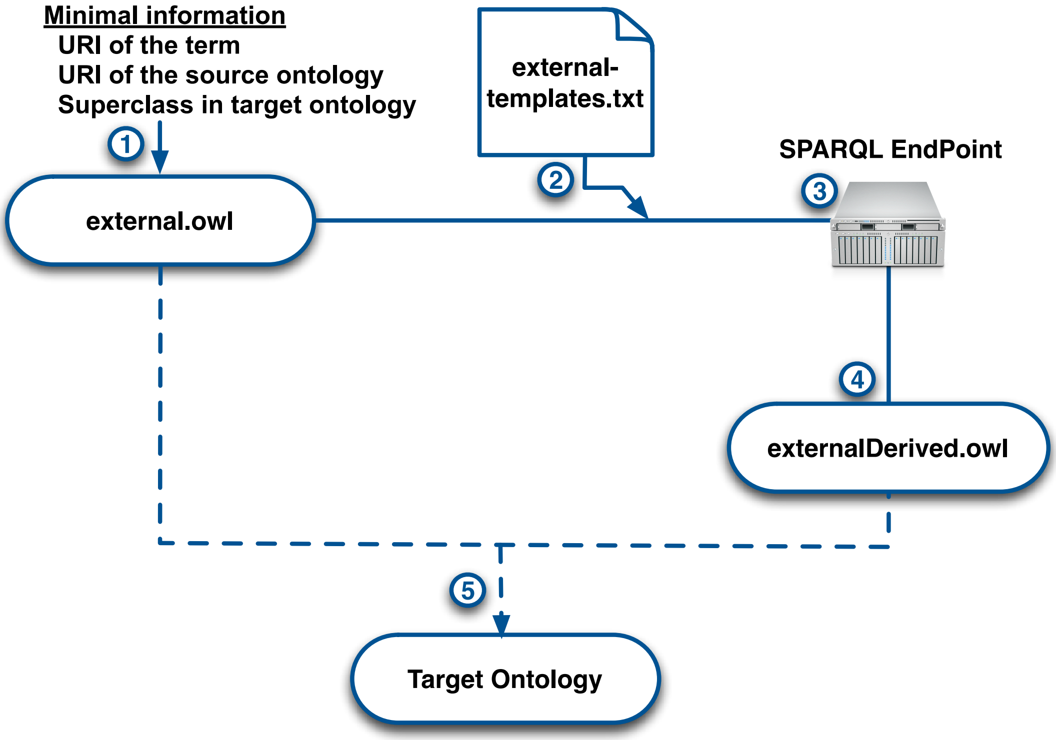

Fig. 2. Diagram of the MIREOT mechanism. 1. the minimal information is added into the external.owl file 2. a script parses the external.owl file, and for each class a sparql template is selected from external-templates.txt by matching the URI to patterns specified for each template. It uses the URI and the appropriate SPARQL template to generate a CONSTRUCT query 3. the SPARQL query is executed against a SPARQL endpoint (e.g. Neurocommons) 4. the results of the SPARQL queries are combined into the externalDerived.owl file 5. the target ontology imports the external.owl and externalDerived.owl files.

This supplementary information, which is prone to change as the source ontologies evolve, is stored in a second file, *externalDerived.owl*. This file can be removed on a regular basis, *e.g.*, before releasing new versions of the target ontology. It can then be rebuilt via script based on *external.owl* in order to refresh the additional information (*e.g.*, label). The two files, *external.owl* and *externalDerived.owl*, are then imported by the target ontology, providing the necessary information to the editors while at the same time keeping it independent from the target ontology's proper classes (see Figure 2).

In the following sections we present two different cases of application of the MIREOT guidelines, implemented during the OBI development.

Use Case One - Basophil and Cell classes

We replaced the OBI class $Cell$ with that from the Cell Type (CL) ontology (20). 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. This class can then be used in turn to import other classes as needed. For example the following invocation of the *add-to-external.pl* script:

```
perl add-to-external.pl CL:0000767 CL:0000000
```

will add the class $basophil$ (CL:0000767) as subclass of the class $cell$ (CL:0000000), and set the source ontology URI as `http://purl.org/obo/owl/CL`. Once imported, the $basophil$ and $cell$ classes 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 than that single external term, and to account for this MIREOT has been devised to be flexible.

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 the ontology, which are in turn mapped from the NCBI taxonomy. We can easily imagine that somebody would want to have a query of the form “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, we decided to retrieve all its superclasses as well up to the root of the NCBI taxonomy. As per the mechanism described above, the mapped class (*e.g.*, human) is defined in external.owl, whereas this additional information to the human class (*i.e.*, its superclasses) are stored in externalDerived.owl.

When the *create-external-derived.lisp* script parses the *external.owl* file and encounters an NCBI taxonomy ID, it will therefore invoke a specific SPARQL query (cf figure 3).

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 mechanism is currently implemented and used by several ontologies, including OBI, the Information Artifact Ontology (IAO)(17), the Vaccine Ontology (VO)(21), the Infectious Disease Ontology (IDO)(23) and the Influenza Ontology (InfluenzO)(24). In the context of OBI, we currently explicitly import 472 terms, which in turn led to actual integration of 1447 classes (due to the automatic retrieval of parents when using the NCBI taxonomy).

```
# give names to the top taxa
alias:bacteria=tax:_2
alias:eukaryota=tax:_2759
alias:archaea=tax:_2157
alias:viruses=tax:_10239
alias:cellularOrganism=tax:_131567


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.
   ?super alias:importedFrom <http://purl.org/obo/owl/NCBITaxon>
 }
where
{
 { # We harvest the transitive superclass annotations
    _ID_GOES_HERE_ rdfs:subClassOf ?super.
    graph <http://purl.org/science/graph/obo/NCBITaxon>
      { ?super rdfs:subClassOf ?parent.
        ?super rdfs:label ?label.
      }
  }
UNION

 { graph <http://purl.org/science/graph/obo/NCBITaxon>
   { ?super rdfs:subClassOf ?parent.
     ?super rdfs:label ?label.
     FILTER (?super=_ID_GOES_HERE_)
   }
  }

FILTER (!((?super=alias:bacteria) || (?super=alias:eukaryota) || (?super=alias:viruses) || (?super=alias:archaea)
 || (?super = alias:cellularOrganism) || (?parent = alias:cellularOrganism)))
}
```

Fig. 3. Template SPARQL query for import from the NCBI taxonomy. The _ID_GOES_HERE_ pattern will be replaced by the relevant NCBITax ID dynamically via script when building the CONSTRUCT query. This query allows retrieval of the class of interest and its parents up to a set of defined root classes. Note that the graph <http://purl.org/science/graph/obo/NCBITaxon> contains the source ontology, but the full store includes inferred subClassOf triples.

A consideration for using this approach is the status of assertions made on imported terms. In adding axioms such as the subclass axiom when importing the external term, the aim is to only assert true statements, for instance those that the developers of the source ontology would agree with. If such additional restrictions are required those should be stored in the importing 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 importing ontology may migrate to the source ontologies at some point in the future; a fruit of the collaborative nature of OBO Foundry ontology development.

If additional annotations are added to the imported terms (for example, we want to add an example of usage for the GO imported term *protein complex*), we also need to ensure that if the imported term is deprecated or replaced by an other (*e.g.*, replacing the GO *protein complex* term with the Protein Ontology (PRO) one), the annotation is similarly removed. With broader use of the MIREOT mechanism by OBI and other resources, several minor issues arose.

For example, consider the case of IAO devlopers needing the term $investigation$. This class already exists in OBI, and IAO developers therefore chose to *MIREOT* it, effectively integrating the class `http://purl.obolibrary.org/obo/OBI_0000066` and distributing it as part of the IAO releases. However, OBI imports IAO, and therefore reimports, *via IAO*, its own $investigation$ class. This is not problematic in general, redundancy of information in OWL files being of no consequence. However when OBI curators decided to update the definition of the $investigation$ class, the information natively in OBI and that imported from IAO became out-of-sync: two different definitions were displayed to the curators. Moreover one of them couldn't be edited as it is not in the active ontology file.

One solution would be to update the IAO import - but this requires a release of OBI with the updated $investigation$ definition, its upload on Neurocommons, and for the IAO developers to update their information and produce a new release of IAO. At best, this implies a delay of a few days, more realistically of a few weeks until the information in both files is again synchronized. Another solution that we think is more sensible would be for tools to recognize and prioritize the origin of a class based on its URI. Ontology editing tools would display only the information originating from the target ontology when editing the target ontology file.

Interestingly, this had an other consequence: when updating the information from Neurocommons, we needed to specify which Resource Description Framework (RDF) graph (26) the term *originally* belongs to. Taking again our example of the $investigation$ class, when querying based on its URI without specifying the RDF graph, the SPARQL endpoint would return the OBI class, but also the one distributed by IAO, which is not the desired behavior: remember that in our example the IAO annotation property values are now out of date compared to the original, authoritative OBI file. As it turns out such examples of copying properties also occur where terms have been reused by different OBO ontologies.

This leads us to our last potential issue: when updating MIREOTed information (*e.g.*, IAO updates its *externalDerived.owl* file), we need to ensure that the SPARQL endpoint where the information resides is up-to-date. As we currently rely on the OBO Foundry resources, we know that the Neurocommons OBO distribution is updated nightly with the latest information from the OBO server, and we are therefore reasonably certain that we are working with current resources. This may not always be easy to know if extending the mechanism to an other SPARQL endpoint, or other sets of ontological resources.

When using the MIREOT standard one must be cognizant of the trade-off between complete consistency checking and heavyweight importing versus lightweight importing but only partial 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, or a module, are imported.

Since we are doing only partial, reasoner supported, consistency checking, we take extra care when we need to make assertions about an imported term. We review the textual definition and, if needed, talk with the original term editor to ensure we understand the denotation. As we are importing from OBO Foundry candidate 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.

Future work

The current implementation of the MIREOT guidelines relies on command-line scripts, making it difficult for some curators to use. A webservice, OntoFox(30), has been developed by the He group at the University of Michigan to facilitate the process: ontology editors can use web forms to input their requirements, or submit specific OntoFox-formatted files for batch creation or update. Ideally, a Protégé (22) plugin could be developed to improve the interaction between the curators and the tool and the implementation of the MIREOT guidelines. NCBO developers have created a widget allowing insertion of external references in an ontology(29), and we hope it will be updated to fully support 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*.As module extraction technology matures, we intend to include the ability to use such mechanisms for doing targeted imports, on a source by source basis.

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 author's work is partially supported by funding from the NIH(R01EB005034), the Public Health Agency of Canada / Canadian Institutes of Health Research Influenza Research Network (PCIRN), 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), and the Michael Smith Foundation for Health Research.

References

[1] Grau BC, Horrocks I, Kazakov Y, and Sattler U. (2007) . Extracting Modules from Ontologies: A Logic-based Approach. Proc. of the Third OWL Experiences and Directions Workshop, number 258 in CEUR

[2] B. Cuenca Grau, I. Horrocks, Y. Kazakov and U. Sattler (2007) Just the right amount: Extracting modules from ontologies. In proc. of the 16th International World Wide Web Conference (WWW 2007)

[3] E. Jimenez-Ruiz, B.Cuenca-Grau, U. Sattler, T. Schneider and R. Berlanga (2008) Safe and Economic Re-Use of Ontologies: A Logic-Based Methodology and Tool Support. 5th European Semantic Web Conference (ESWC 2008)

[4] J. Seidenberg, A. Rector (2006) Web ontology segmentation: analysis, classification and use. In proc. of the 15th International World Wide Web Conference (WWW 2006)

[5] Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., and Katz, Y. (2007). Pellet: A practical OWL-DL reasoner. Web Semant. 5, 2 (Jun. 2007), 51-53.

[6] Web Ontology Language (OWL), `http://www.w3.org/2007/OWL/`.

[7] 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.

[8] 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.

[9] OBI Ontology, `http://purl.obolibrary.org/obo/obi`.

[10] 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.

[11] 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. Whet- zel, and S. Lewis. The obo foundry: coordinated evolution of ontologies to support biomedical data integration. Nature biotechnology, 25(11):1251-1255, Nov 2007.

[12] Melissa A. Haendel, Fabian Neuhaus, David Osumi-Sutherland, Paula M. Mabee, Jose L.V. Mejino Jr., Chris J. Mungall, Barry Smith. CARO âĂŤ The Common Anatomy Reference Ontology, in Anatomy Ontologies for Bioinformatics Principles and Practice, Series: Computational Biology , Vol. 6, ed. Burger, Albert; Davidson, Duncan; Baldock, Richard, 2008

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

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

[15] OBI scripts - `http://purl.obolibrary.org/obo/obi/repository/`.

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

[17] The Information Artifact Ontology (IAO), `http://code.google.com/p/information-artifact-ontology/`.

[18] Neurocommons OBO SPARQL endpoint - http://sparql.obo.neurocommons.org/.

[19] Ruttenberg A, Rees J, Samwald M, Marshall M. Life sciences on the Semantic Web: the Neurocommons and beyond Briefings in Bioinformatics 10 , 193-204 2009 available at `http://purl.org/science/publication/neurocommons2009.pdf`

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

[21] The Vaccine Ontology - `http://www.violinet.org/vaccineontology/`

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

[23] The Infectious Disease Ontology - `http://www.infectiousdiseaseontology.org/`

[24] The Influenza Ontology - https://sourceforge.net/projects/influenzo/.
[25] Darren A. Natale, Cecilia N. Arighi, Winona Barker, Judith Blake, Ti-Cheng Chang, Zhangzhi Hu, Hongfang Liu, Barry Smith, and Cathy H. Wu, "Framework for a Protein Ontology", Proceedings of the First International Workshop on Text Mining in Bioinformatics, 2006, pp. 29-36.
[26] RDF/XML Syntax Specification - http://www.w3.org/TR/rdf-syntax-grammar/
[27] Phenotypic Quality Ontology - http://obofoundry.org/wiki/index.php/PATO:Main_Page
[28] Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., AlcÃČÂąntara, R., Darsow, M., Guedj, M. and Ashburner, M. (2008) ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Res. 36, D344-D350.
[29] BioPortal Reference Plugin - http://protegewiki.stanford.edu/index.php/BioPortal_Reference_Plugin.
[30] OntoFox - http://ontofox.hegroup.org/.
[31] - SPARQL queries template file - http://obi.svn.sourceforge.net/svnroot/obi/trunk/src/tools/build/external-templates.txt

BFO Basic Formal Ontology
CL Cell Type
GO Gene Ontology
MIREOT Minimum Information to Reference an External Ontology Term
OBI Ontology of Biomedical Investigations
OBO Open Biomedical Ontologies
OWL Web Ontology Language
IAO Information Artifact Ontology
VO Vaccine Ontology
IDO Infectious Disease Ontology
InfluenzO Influenza Ontology
CARO Common Anatomy Reference Ontology
PRO Protein Ontology
RDF Resource Description Framework
FMA Foundational Model of Anatomy
PATO Phenotypic Quality Ontology
ChEBI Chemical Entities of Biological Interest