ICBO: International Conference on Biomedical Ontologies · Buffalo, NY, USA
Working with Multiple Biomedical Ontologies Workshop, July 26, 2011

Building the OBO Foundry – One Policy at a Time

Mélanie Courtot[1], Chris Mungall[2], Ryan R. Brinkman[1,3], Alan Ruttenberg[4]

[1]BC Cancer Agency, Vancouver, BC, Canada
[2]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[3]Department of Medical Genetics, University of British Columbia, Vancouver, BC, Canada
[4]University at Buffalo, NY, USA

Abstract. Policy drafting, discussion and implementation is not the most exciting or interesting thing to do when developing new resources. However, when trying to identify existing work that can be built upon in one's project, such policies are critical to allow interoperability and reliability. We describe some tools and guidelines developed under the OBO Foundry umbrella, and show how they help realize critical maintenance functions, increasing overall quality and sustainability of resources.

1 Introduction

With the increasing number of ontologies created in the biomedical domain, the ability to work with multiple resources is critical for developers. This allows developers to concentrate on new requirements rather than duplicate existing effort. However, in order to harmoniously build on several distinct bodies of work originating from different communities, guidelines should be established and followed. We present our experience taking part in the Open Biomedical Ontologies (OBO) Foundry [1] consortium. We briefly summarize some work that has been done under the OBO Foundry umbrella in defining a common ID policy[1], and a shared metadata set incorporated in the Information Artifact Ontology (IAO) [2]. Finally, we describe the issues related to a lack of a common deprecation policy, and propose a process for harmonizing expected behaviour across resources.

2 The OBO Foundry

The OBO Foundry is a set of ontologies which are intended to be interoperable, designed following a similar philosophy and implemented in accordance with a set of principles and guidelines. Authors of resources submitted to the OBO Foundry library[2] commit to working together to increase quality of resources. As a result of that collaborative work, resources part of the OBO Foundry are orthogonal in scope (i.e., each resource describes a specific, non-overlapping domain) – and common policies are devised and followed. To increase interoperability, ontologies use a common upper ontology (Basic Formal Ontology (BFO)) [3] and a common set of relations (Relations Ontology (RO)) [4]. Policy adoption at the level of the OBO Foundry is done by decision of the OBO coordinators, a set of individuals whose task is to help build a community adhering to the OBO principles, and facilitate collaborations and cooperation between groups and resources.[3]

Common Unique Identifier Policy

The OBO foundry currently hosts resources under the OBO format [5] and the Web Ontology Language (OWL) [6] format, and aims at providing tools such as the OWLAPI mapping for OBO format [4] to allow their interconversion. In order to do so, one key requirement is to rely on a common system to handle unique identifiers for entities. A policy, normative for Foundry resources, includes a Foundry-compliant Uniform Resource Identifier (URI) scheme, and rules to map from current OBO IDs and OBO legacy URIs towards them. Following a common ID policy allows URIs to be more reliable, and ensures they are unique within the Foundry consortium. It also helps

[1] The OBO Foundry policy is available at http://www.obofoundry.org/id-policy.shtml
[2] http://www.obofoundry.org/
[3] http://obofoundry.org/coordination.shtml
[4] http://code.google.com/p/oboformat/

building tools relying on this ID scheme. For example, the Ontology of Biomedical Investigations (OBI) [7] developers do not deal with ID management when creating entities; rather a script is run pre-release to check and homogenize URIs for format and stability (e.g., was any URI deleted since the last release?). Another feature is to allow dereferencing and provide useful information to a user trying to resolve terms' URIs. The OntoBee browser[5] displays a HTML page that provides human readable information on each term, such as label and textual definition, while the page source is RDF that can be machine-processed. Finally, the ID policy specifies versioning rules for ontology releases, effectively creating a version history for resources. By doing so, users are always free to access the latest published version and get the most up to date developments, or instead use a specifically dated release, and maintain stability of their own resource.

3 Improving Documentation by Sharing Metadata through the IAO

The IAO is an ontology of information entities, which aims at providing high-level blocks upon which specific resources can build upon. It describes classes such as *directive information entity*, which can for example be extended in a clinical-focused ontology by the *clinical guideline* subclass. As part of the IAO project, a distinct file defining common metadata properties[6] has been created. This file can be imported independently of the "core" IAO, and used by any developer. The IAO common metadata set contributes to the realization of the principle of documenting ontologies within the OBO Foundry.

Other efforts already exist to formalize metadata, such as the Simple Knowledge Organization System (SKOS) [8] and the Dublin Core (DC) Metadata element set [9]. However, we found them not adequate for our usage. If we consider the case of `dc:creator`, its definition reads "An entity primarily responsible for making the resource.", where the resource is the resource described by the class bearing this property. For example, if we describe a book, the `dc:creator` property value is set to the name of the author of the book, and does not capture the name of the author of the book description, which is what we would aim at capturing with `iao:definition_editor`[7]. Similarly, the definition of `skos:definition` defines concepts, which is not suitable in our case.

Common and expected annotation properties, such as *definition* and *editor preferred term* are documented, and allow tool developers to rely on them to build their user interface. Other properties such as *definition source* or *definition editor* were created to store any references used in developing the definition and who did created the term. This allows resource consumers to go back and check on the origin of the term and what its intended meaning is, and/or contact the relevant individual should they need more clarification about its usage. Similarly, curators of the ontology can add *example of usage* and *editor note* to further clarify what the term denotes and what its intended usage is. Other slightly more complex properties have been designed to enable quality assessment of the terms. Namely, the *curation status specification* class provides a list of predefined instances (*i.e.*, *'example to be eventually removed', 'metadata complete', 'organizational term', 'ready for release', 'metadata incomplete', uncurated, 'pending final vetting', 'to be replaced with external ontology term', 'requires discussion'*[8]) that can be used on each class to mark its degree of "readiness" and stability. Similarly, the class *obsolescence reason specification* offers a list of predefined values that can be used on obsoleted terms to give more information as to why that term was deprecated and indicate (in conjunction with for example an editor note) what the term replacement is. Finally, an *OBO Foundry unique label* annotation property (`http://purl.obolibrary.org/obo/IAO_0000589`), was recently added in the ontology-metadata file to allow disambiguation between

[5] http://www.ontobee.org/

[6] http://purl.obolibrary.org/obo/iao/ontology-metadata.owl

[7] http://dublincore.org/documents/dcmes-xml/, section 2.4

[8] http://code.google.com/p/information-artifact-ontology/wiki/OntologyMetadata

terms local to a resource when they are taken in the whole set of OBO Foundry ontologies. *OBO foundry unique labels* are automatically generated based on regular expressions provided by each ontology, when processed by the OBO package manager currently being written by the OBO Foundry custodians.

4 Maintaining Orthogonality through MIREOT

The OBO Foundry requires that newly created ontologies be orthogonal to resources already lodged within OBO. As a consequence, when in implementing a new resource, care should be taken to reuse work done in the context of other efforts where possible. Additionally, reusing terms from other resources allows developers to rely on the knowledge of domain experts who curated them and to dedicate more work time for novel terms *de novo*. Avoiding duplication of resources increases interoperability. A single URI is created per term, preventing the need for tedious mappings between terms with the same meaning in different resources.

When only few terms of interest are identified in external ontologies, those can be imported relying on the Minimum Information to Reference an External Ontology Term (MIREOT) guideline [10]. For example the Vaccine Ontology (VO) [11] defines the *vaccination* process as an "administering substance in vivo that involves in adding vaccine into a host (e.g., human, mouse) in vivo with the intend to invoke a protective immune response", and the Adverse Events Reporting Ontology (AERO) [12] uses it as a synonym of the "immunization process" needed to define vaccine adverse events. The MIREOT mechanism provides a way to selectively import a term from a source ontology into a target resource, without the overhead of importing the whole external file. A more complete discussion pertaining to the trade-off of using MIREOT vs. other options, such as *modules* [13], is available in the MIREOT manuscript [10].

5 Deprecation Policy, an Unmet Need

Sometimes terms need to be retired as ontologies evolve. The OBO Foundry doesn't currently formalize a standard deprecation policy, which leads to the problem of different policies within resources. As a general guideline, deprecated terms are not deleted from the ontology, as removing a term that has been used in the past can be confusing for users. Some discrepancies exist between the practice of the Gene Ontology (GO) [14] and OBI: in the GO [15], when terms are merged, one term effectively disappears from the ontology file and its identifier is maintained as an *alt_id* annotation property on the term it is merged with. By contrast in the OBI, one term is deprecated, and its *obsolescence reason specification* is set to "term merged", with the addition of an editor note indicating the replacement term. As a consequence, tools such as MIREOT, developed in the context of the OBI, expect to find the URI of classes in their declaration (and not as a secondary ID). MIREOT scripts are therefore unable to retrieve the external information in the GO merging case, resulting in a loss of terms on the importing ontology side, such as recently happened with some Phenotypic Quality Ontology (PATO) terms [16][9]. A common deprecation policy, following the example of what has been done regarding the ID policy, would help formalize expected behaviour, and guide tools developers. A review of the current reasons for obsolescence in the GO would be useful to perform to ensure adequacy between the instances defined by the IAO and the needs of the curators.

6 Evaluation

Most proposed policies have been adopted fairly recently, and evaluation is very preliminary. Although the relative costs and benefits could be difficult to quantify, a number of use cases illustrate the advantage of relying on numerical identifiers. For example, when choosing to use numerical IDs for terms, we know that some tooling issues will hinder adoption of those standards - nobody wants to type in OBI_0001234 when doing a SPARQL query. However, we believe that in the long term

[9] http://sourceforge.net/mailarchive/forum.php?thread_name=99D14FA3-9952-4C67-B892-41A8499A43C8\%40gmail.com&forum_name=obi-devel

(i) tooling issues will be resolved and (ii) using numerical IDs will be beneficial for maintenance of the resources and their necessary evolution. As illustration, see for example the recent threads mentioning how the (i) Protégé [17] team added a new menu "render by rdfs:label" to their interface[10] and (ii) issues faced by the developer of GoodRelations [18] to rename some classes.[11]

This paper is presented as a position paper/statement of interest. Our objective is to solicit feedback and interest from the community, and encourage participation in the development of current and future policies.

Acknowledgments

The authors' work was partially supported by funding from the Public Health Agency of Canada / Canadian Institutes of Health Research Influenza Research Network (PCIRN), and the Michael Smith Foundation for Health Research. The authors wish to acknowledge people who contributed comments to the OBI deprecation policy: Suzanne Lewis, James Malone, Daniel Schober, Allyson Lister.

References

1. Ashburner M. Smith B., Rosse C., Bard J., Bug W., Ceusters W., Goldberg L. J., Eilbeck K., Ireland A., Mungall C. J., Leontis N. OBI Consortium, Rocca-Serra P., Ruttenberg A., Sansone S. A., Scheuermann R. H., Shah N., Whetzel P. L., and Lewis S.. The OBO foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature biotechnology*, 25(11):1251–1255, 2007.
2. The Information Artifact Ontology (IAO), http://purl.obolibrary.org/obo/iao.
3. The Basic Formal Ontology (BFO), http://www.ifomis.org/bfo/.
4. B. Smith, W. Ceusters, B. Klagges, J. Kohler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. L. Rector, and C. Rosse. Relations in biomedical ontologies. *Genome biology*, 6(5):R46, 2005.
5. The OBO Flat File Format Specification, version 1.2, http://www.geneontology.org /OLS.format.obo-1_ 2.shtml.
6. Web Ontology Language (OWL), http://www.w3.org/2004/OWL/.
7. OBI Ontology, http://purl.obolibrary.org/obo/obi.
8. Simple Knowledge Organization System (SKOS), http://www.w3.org/TR/2009/REC-skos-referenc e-20090818/.
9. Dublin Core Metadata Element Set, http://dublincore.org/documents/dces/.
10. M. Courtot, F. Gibson, A. L. Lister, J. Malone, D. Schober, R. R. Brinkman, and A. Ruttenberg. Mireot: The minimum information to reference an external ontology term. *Applied Ontology*, 6(1):23–33, 2011.
11. The Vaccine Ontology, http://www.violinet.org/vaccineontology/.
12. The Adverse Even Reporting Ontology (AERO), http://purl.obolibrary.org/obo/aero.
13. Grau B.C., Horrocks I., Kazakov Y., and Sattler U. Extracting modules from ontologies: A logic-based approach. Proc. of the Third OWL Experiences and Directions Workshop, number 258 in CEUR, Innsbruck, Austria.
14. Gene Ontology Consortium. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(90001):D258–D261, 2004.
15. GO editorial style guide, http://www.geneontology .org/ GO.usage.shtml.
16. Phenotypic Quality Ontology (PATO), http://obofoundry.org/wiki/index.php/PATO:Ma in_Page.
17. The Protégé Ontology Editor and Knowledge Acquisition System, http://protege.stanford.edu/.
18. Martin Hepp. Goodrelations: An ontology for describing products and services offers on the web. In Aldo Gangemi and Jrme Euzenat, editors, *Knowledge Engineering: Practice and Patterns*, volume 5268 of *Lecture Notes in Computer Science*, pages 329–346. Springer Berlin / Heidelberg, 2008.

[10] https://mailman.stanford.edu/pipermail/p4-feedback/2011-May/003889.html

[11] http://lists.w3.org/Archives/Public/public-lod/2011Apr/0278.html