B-Annot: Supplying Background Model Annotations for Ontology Coherence Testing

Vojtěch Svátek[1], Simone Serra[1], Miroslav Vacura[1],
Martin Homola[2] and Ján Kľuka[2]

[1] Univ. of Economics, Prague, W. Churchill Sq.4, 130 67 Prague 3, Czech Republic
{svatek,vacuram}@vse.cz, serrazimone@gmail.com
[2] Comenius University in Bratislava, Mlynská dolina, 842 48 Bratislava, Slovakia
{homola,kluka}@fmph.uniba.sk

Abstract. The demo paper presents *B-Annot*, a Protégé plugin for annotation of ontologies and linked data vocabularies by background model distinctions. In addition, it briefly demonstrates the subsequent use of the annotations created by *B-Annot*, for verifying the ontological coherence of the ontologies/vocabularies at the level of meta-models. Finally, possible further extensions of the tool and its role in the (background-model-driven) ontological engineering workflow are briefly discussed.

1 Introduction and Motivation

With the growing popularity of the semantic web, a large portion of new ontologies, such as Linked Data (LD) vocabularies, has been directly authored in OWL and thus influenced from the beginning by its inventory of constructs, and also by particular application needs. Let us call such an operational artefacts *ontological foreground model* (OFM). On the other hand, by giving priority to mimicking as much as possible (at least, in some aspects) what is observed in the real-world, we arrive at an *ontological background model* (OBM). For instance:

- OWL classes may sometimes represent permanent *types* of objects and sometimes just *roles* played by these objects in a certain phase of their existence;
- OWL individuals may represent true *individual objects* ('particulars'), but also *universal* entities (types), or even *relationships* whose existence fully depends on the participating objects.

When the OFMs are, e.g., visualized, reused, matched or transformed, such 'hooding' may cause troubles. For example, in an OFN it can happen that a class (i.e. role) *Student* becomes superclass of classes *Human* and *Robot*; an object then may stop being member of the superclass while remaining member of the subclass. For another example see the upper part of the diagram (adapted from [8]) in Fig. 1, depicting the complex fact of a business entity (resource 3) offering exemplars (i.e., 'some items') of a certain musical album (resource 1) as product for sale, in a certain region. The fact refers to two LD vocabularies: the e-commerce ontology GoodRelations (GR)[3] and the Music Ontology (MO).[4] The

[3] http://purl.org/goodrelations/v1
[4] http://purl.org/ontology/mo/

remaining two instance-level resources in the diagram (2 and 4) are the 'offering' itself and the value '90' (minutes) understood as 'typical' and thus modeled as a resource rather than literal.[5] In the lower part of the diagram we approximate the *ontological background* of this fragment (omitting the entities that would be *types* in both diagrams, for easier readability). Among other things we see that notion of 'album', originally being the value of the object property `mo:release_type`, now becomes an additional type of the product offered, and that the 'Offering' object becomes absorbed by the 'offers' relationship (now with arity >2).

Obviously, modelling the ontological background for each individual data fragment is infeasible. The mapping between the 'foreground view' of the domain (as contained in the vocabulary) and the corresponding 'background view' thus has to be established at the level of entity *types*, which means, indirectly (note that especially less expressive vocabularies are just collections of unlinked entities whose connection is only established at the level of instance data). On the one side of the mapping is an *ontological foreground model* (OFM), i.e., the structure of an RDFS/OWL ontology; on the other side is an analogous *ontological background model* (OBM). OBM models should be represented in a suitable *OBM language* of modelling primitives (OBML). Two such languages are

- *OntoClean* [3], which labels OFM classes with the ontological notions of *essentiality*, *rigidity* (e.g., in the first example mentioned, 'permanent' classes *Human* and *Robot* would be rigid while the 'temporary' class *Student* would be anti-rigid), *identity* and *unity*.
- The recently designed *PURO* OBML [9],[6] aiming to capture the background distinctions of OFM entities as in the bottom part of Fig. 1: that between *objects* ('particulars') and their *types* ('universals') and that between *relationships* (or 'valuations' by a quantitative value) and self-standing objects.

OntoClean has proven useful for taxonomy-centric ontologies that dominate, e.g., in bioinformatics. On the other hand, PURO has been specifically designed for 'relation-centric' ontologies/vocabularies [8], which are prominent in LD. Another important phenomenon in LD is that an existing entity might be systematically used with a different background distinction than foreseen in the vocabulary specification; for example, a property that is assumed to have categories of objects in its range might refer to individual objects in some dataset. Therefore, 'generic' annotation of vocabularies might not be sufficient; we should also be able to annotate vocabularies 'as they are used' in a specific dataset.

By their capacity of underlying the entities from various operational (typically, domain-restricted) knowledge models with background ontological distinctions, OBMLs are analogous to *foundational ontologies*. The difference is in the way the 'surface' and 'deep' model are interconnected. A foundational ontology provides root concepts upon which the 'surface model' concepts are grafted; both models thus share the same space. In contrast, OBMs reside in their own 'layer';

[5] Such kind of modeling is not common in MO, but, rather, in GR-compliant ontologies, cf. `http://www.ebusiness-unibw.org/ontologies/opdm/#ontologies`.

[6] A more extensive description is in [7].

when connecting an OFM with an OBM, we thus need to 'inject' a 'proxy' of one model to the other model, in order not to let the one interfere with the formal semantics of the other. Two alternatives for creating such a 'proxy', assuming both layers are to be expressed in OWL-DL, are as follows:

- OBM entities could become values of specific *OWL annotation properties*, and be saved as unobtrusive part of (a copy of) the OFM.
- OFM entities (classes, properties and individuals) could be uniformly *meta-modelled* as syntactical instances to be inserted as an A-Box into a meta-modelling ontology, where their mapping to OBM can be captured.

The first alternative is favourable for visibility of the OBM distinctions to a human when working with the OFM. The second alternative, in turn, allows to carry out conceptual coherence checking according to constraints defined in the meta-modelling ontology, via a generic OWL DL reasoning mechanism. This approach has been previously tested for the OntoClean OBML in [2, 10], and later for the PURO OBML by us [9].

In this system/demonstration paper we present *B-Annot*: a Protégé plugin[7] that allows to create and save meta-models of a selected vocabulary with respect to either OntoClean or PURO, and (especially for the latter) in two modalities, 'generic' and 'dataset-specific'. (Storage of OBM distinctions in annotation property values, as well as other enhancements, is forthcoming.) We also briefly demonstrate how the annotations can be used for conceptual coherence checking; in contrast to PURO-only coherence checking described in [9], we nowadays rely on a modular set of ontologies that also includes an OntoClean module.

2 *B-Annot* Functionality

In summary, the tool allows the user, for the vocabulary to be annotated already loaded into the Protégé editor,

- to select the meta-ontology (either OntoClean or PURO) and decide whether generic or dataset-level annotation is going to take place;
- *for dataset-specific annotation*, to inspect the statistics of presence of entities from the given vocabulary in different datasets, fetched online from LOD-Stats [1], to select an appropriate dataset, and to view the list of entities from the vocabulary that occur in this dataset;
- *for dataset-specific annotation*, to browse a pre-computed summary of the dataset (inspired by [4]), with entities from the vocabulary highlighted;
- select an entity (in one of the Protégé tabs) and annotate it with a background model distinction;
- save the whole annotation set to an RDF file, and load it back.

[7] Available from `http://patomat.vse.cz/cz.vse.bannotation.plugin.view.jar`.

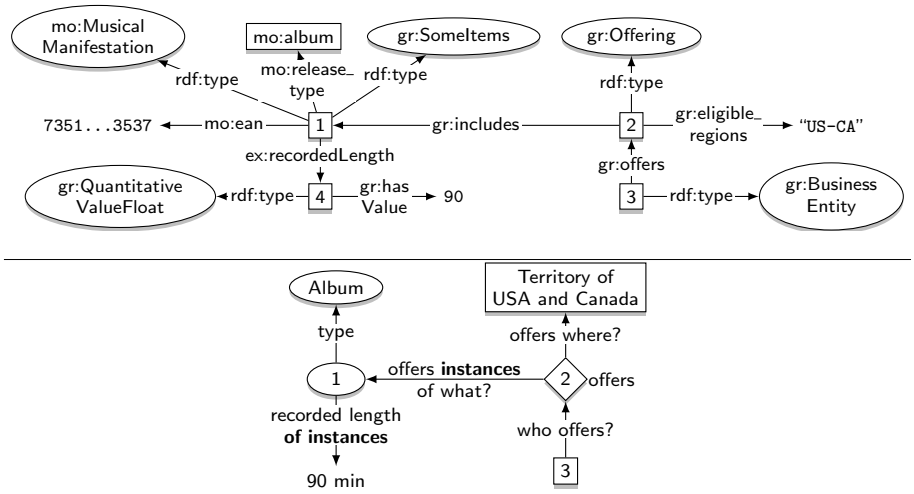

Fig. 1. RDF data fragment and its ontological background

We will now describe the scenario of dataset-specific annotation, since generic annotation is essentially a subset of it. Furthermore, we will use the PURO meta-ontology as more relevant in the dataset-specific mode. (OntoClean would be applied in the same way.) Fig. 2 shows the *B-Annot* interface after the choice of PURO and dataset-specific annotation mode (FOAF has been previously loaded into Protégé as ontology to be annotated). The user can see that of the 14 datasets for which the statistics has been fetched, 10 use some number of FOAF entities, ranging from 1 to 23; these are relevant to the annotation session. After clicking at the 'summary' button for the Geospecies dataset, an ordered listing of frequent 'class-property-class' is displayed, a part of which is in Fig. 3.[8] FOAF entities, here the properties *depiction*, *isPrimaryTopicOf*, *primaryTopic* and *topic*, are displayed in red. Finally, the actual annotation takes place. In Fig. 4 we see that the user, based on the observation that *foaf:topic* is usually valued by biological taxa[9] in this dataset, assigns this property the PURO label 'PrT' ('property whose range is a type') from the pull-down menu (with items picked from the meta-ontology depending on the entity type to be annotated: class, property or individual). Entity annotations are subsequently listed in the bottom part of the window, and can, eventually, be saved (and reloaded) in bulk, as a set of *hasLabel*[10] triples.

[8] We also experiment with 'class-property-class-property-class' paths, but they are not implemented in the current version of the system.

[9] For the sake of this example, we omit the philosophical discussion whether and for what purpose a taxon should indeed be understood as a universal.

[10] Every meta-modelling ontology has its own *hasLabel* property; here it is the one from the PURO ontology.

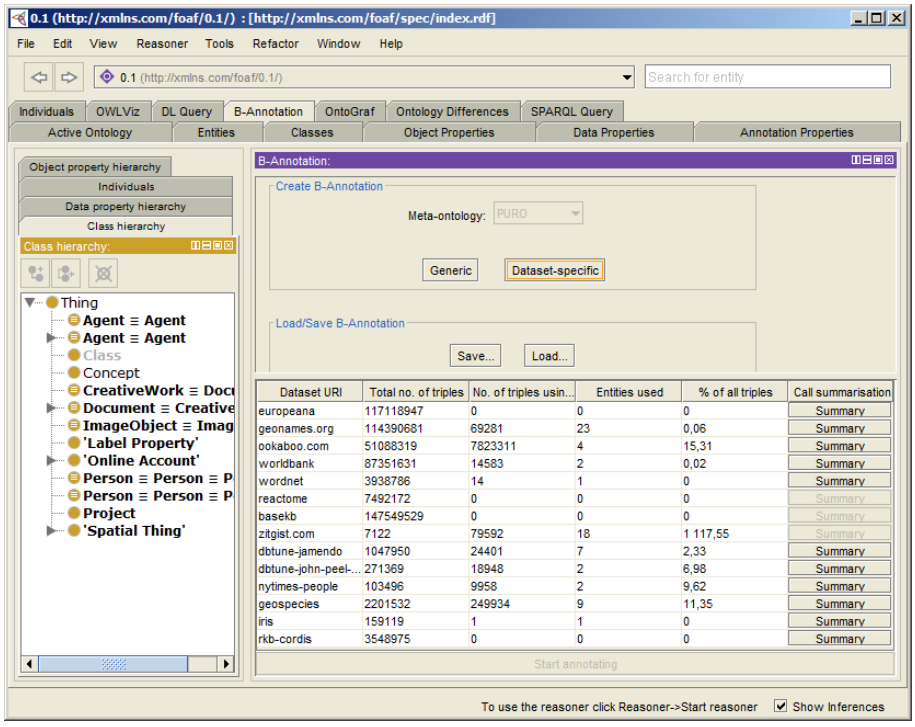

Fig. 2. Dataset choice in dataset-specific annotation

3 Coherence Checking Examples

For each OBML considered, the distinctions underlying a particular OFM can be compared to a predefined set of coherence rules. For OntoClean there are four standard coherence rules [3]: Given two properties, p and q, when q subsumes p then: a) if q is anti-rigid then p must be anti-rigid, b) if q carries an identity criterion then p must carry the same criterion, c) if q carries a unity criterion then p must carry the same criterion, and d) if q has anti-unity then p must also have anti-unity. The PURO OBML, in turn, specifies three constraints: for a) entity coherence, b) type coherence, and c) relation coherence (for details see [9]). We demonstrate the coherence checking on two example annotations.

The first is a fragment of the GR ontology annotated with PURO OBML,[11] containing class *ProductOrService* with subclasses *Individual* and *ProductOrSer-*

[11] `http://patomat.vse.cz/gr_mm.owl`

Summary for dataset

Type 1	Property	Type 2	Count
geosp:Continent	geosp:hasExpectationOf	http://rdf.geospecies.or...	64
geosp:SpeciesConcept	foaf:depiction		62
http://rdf.geospecies.org/ont/families/...	geosp:hasSubgenusName		62
geosp:PhylumConcept	rdfs:seeAlso	geosp:Bio2RDFtaxon	60
geosp:PhylumConcept	rdfs:seeAlso	geosp:UniprotTaxon	60
geosp:KingdomConcept	foaf:isPrimaryTopicOf		60
geosp:Bio2RDFtaxon	skos:closeMatch	geosp:PhylumConcept	60
geosp:UniprotTaxon	skos:closeMatch	geosp:PhylumConcept	60
geosp:PhylumConcept	skos:closeMatch	geosp:Bio2RDFtaxon	60
geosp:PhylumConcept	skos:closeMatch	geosp:UniprotTaxon	60
geosp:OrderConcept	geosp:hasGBIF		60
http://rdf.geospecies.org/ont/families/...	geosp:isUnexpectedIn	geosp:Country	58
geosp:Country	geosp:hasLowExpectationOf	http://rdf.geospecies.or...	58
geosp:PhylumConcept	geosp:hasCommonName		55
geosp:PhylumConcept	skos:altLabel		55
http://rdf.geospecies.org/ont/families/...	geosp:hasITIS		53
geosp:ClassConcept	dcq:modified		50
	foaf:primaryTopic	geosp:ClassConcept	50
geosp:ClassConcept	dcq:isPartOf	void:Dataset	50
geosp:ClassConcept	dcq:title		50
geosp:ClassConcept	skos:prefLabel		50
	foaf:topic	geosp:Bio2RDFtaxon	50

Fig. 3. Dataset summary for Geospecies, with FOAF entities emphasized

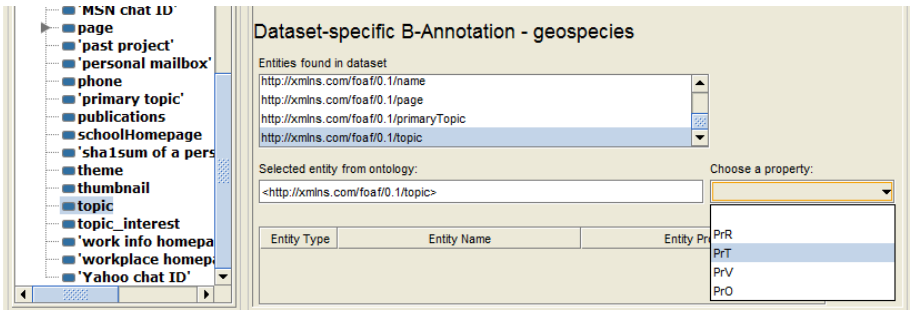

Fig. 4. Annotation of *foaf:topic* by a PURO label, for Geospecies

viceModel. Using DL consistency checking over the PURO meta-ontology[12] and this fragment leads to inferred membership of class *ProductOrService* to a special 'diagnostic' class of the PURO ontology: *Incoherent-TPU*. This class which contains meta-models of classes that 'do not have homogeneous instances' in terms of PURO, specifically, whose instances can be both particulars and universals.

The second example is annotation of a fragment of the ontology used to demonstrate OntoClean inconsistencies in [3]. This fragment[13] includes meta-entities representing six classes of the original ontology annotated with OntoClean labels. The OntoClean meta-ontology used for coherence checking[14] allows for validation of all four coherence rules. The ontology contains four classes (Incoherence-Antiunity, Incoherent-Identity, Incoherent-Rigidity, Incoherent-Unity) that are – as result of inference – filled with individuals that represent classes in meta-model that are incoherent with regard to respective OntoClean rules. For example, the class *AmountOfWater* was annotated with OntoClean labels **+O ∼U +R**. Its subclass *LivingBeing* was annotated with OntoClean labels **+O +U +R**. The defect of the model is that a class with anti-unity label (simply said, class of objects whose arbitrary 'section' is again an instance of the same class) cannot subsume a class with unity label (i.e., containing objects that have 'strict boundaries around themselves'). Therefore it is inferred that the individual meta-modelling the class *LivingBeing* belongs to the diagnostic class *Incoherent-Antiunity*.

4 Conclusions and Future Work

The *B-Annot* plugin represents the first proof-of-concept implementation of annotation technology for ontologies and vocabularies that is (1) not restricted to a single theoretical framework but supports multiple OBMLs, and (2) interconnects the browsing/editing of ontologies (as supported by common ontological editors) with LD summaries. It is a part of a prospective eco-system of tools (other existing ones include, e.g., pattern-based ontology transformation tools [6]) supporting (informed rather than merely intuitive) reuse and design of ontologies on the semantic web.

Serious usability tests and requirement collection for *B-Annot* is only planned after some of the envisaged enhancements will have taken place.

A straightforward extension of *B-Annot* will be the possibility to also store annotations in OWL annotation properties of a copy of the annotated ontology. This will allow for easy browsing of the annotations in their original context.

Background annotation by distinctions referring to notions like 'rigid' (in OntoClean) or 'particular' (in PURO) risks to discourage even reasonably experienced ontological engineers without philosophical background. The threshold should thus be set as low as possible in the future, via operationalized annotation guidelines. For OntoClean's rigidity alternatives, a promising approach has

[12] `http://patomat.vse.cz/puro_v1.1.owl`
[13] `http://patomat.vse.cz/ontoclean-coherence-check-1.owl`
[14] `http://patomat.vse.cz/ontoclean-v.1.0.owl`

already been shown by Seyed, who designed a wizard relying on common-sense verbalization of the meaning of these alternatives [5]. For the PURO OBML distinctions, textual guidelines with examples have already been designed and tested in an classroom assignment; the experience gained will be used to design verbalisation templates similar to those from [5].

As the amount of mature vocabularies and their stable entities is still low[15] their purely manual annotation via *B-Annot* is feasible. In long term, however, partial automation could be achieved by leveraging on two different sources: (1) via linguistic parsing of associated texts, especially the values of `rdfs:comment`, and, (2) via logically inferring the most likely annotations based on previously assigned annotations of interrelated entities, e.g., from superclasses to subclasses.

This work has been supported from the EU ICT FP7 under no. 257943 (LOD2 project), from the VSE IGA project no. 34/2014, from the Slovak VEGA project no. 1/1333/12, and from project APVV-0513-10.

References

1. Auer, S., Demter, J., Martin, M., Lehmann, J.: LODStats An Extensible Framework for High-Performance Dataset Analytics. In: EKAW 2012, Galway, Springer LNCS 7603.
2. Glimm, B., Rudolph, S., Völker, J.: Integrated metamodeling and diagnosis in OWL 2. In: Proc. ISWC 2010.
3. Guarino, N., Welty, C.: An Overview of OntoClean. In: Staab, S., Studer, R., eds.: *The Handbook on Ontologies*, pp. 151–172, Springer-Verlag, 2009.
4. Presutti, V., et al.: Extracting core knowledge from Linked Data. In: Proceedings of the Second Workshop on Consuming Linked Data, COLD2011. (2011)
5. Seyed, P.: A Method for Evaluating Ontologies – Introducing the BFO-Rigidity Decision Tree Wizard. In: FOIS 2012: 191–204.
6. Šváb-Zamazal, O., Dudáš, M., Svátek, V.: *User-Friendly Pattern-Based Transformation of OWL Ontologies.* In: Proc. EKAW'12, Galway.
7. Svátek, V., Homola, M., Kĺuka, J., Vacura, M.: Ontological Distinctions for Linked Data Vocabularies. Technical Report TR-2013-039. Comenius University, Bratislava, 2013. Available online: `http://kedrigern.dcs.fmph.uniba.sk/reports/display.php?id=54`
8. Svátek, V., Homola, M., Kĺuka, J., Vacura, M.: Mapping Structural Design Patterns in OWL to Ontological Background Models. In: Proc. K-CAP 2013, ACM.
9. Svátek, V., Homola, M., Kĺuka, J., Vacura, M.: Metamodeling-Based Coherence Checking of OWL Vocabulary Background Models. In: Proc. OWLED 2013, online `http://ceur-ws.org/Vol-1080/owled2013_6.pdf`.
10. Welty, C.: OntOWLClean: Cleaning OWL ontologies with OWL. In: Proc. FOIS 2006.

[15] The statistics at `http://lov.okfn.org/dataset/lov/stats/` reveals that out of the several thousand entities referenced in LD, there are only about 150 that are at the same time reused by more than one other vocabulary and instantiated by at least 100 LOD instances.