Metamodeling-Based Coherence Checking of OWL Vocabulary Background Models

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

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

Abstract. The surface (or, foreground) structure of linked data and their associated OWL vocabularies can be complemented by background models expressing valid ontological distinctions that may have become obscured by the modeling style chosen by the vocabulary designer. Background models can generally serve for debugging, visualization, matching, or even pattern-based design of operational ontologies such as linked data vocabularies. An example of a well-known background model language, primarily suited for taxonomic ontologies, is the system of OntoClean meta-properties. We present an alternative type of background model language, dubbed PURO, which is oriented towards linked data ontologies, and relies on particular–universal and relationship–object dichotomies. Typical 'foreground' manifestations of background language terms are then discussed. We demonstrate how a PURO background structure of OWL vocabularies can be itself modeled in OWL using a dedicated PURO ontology, and how reasoning over such a metamodel can verify ontological coherence of the original OWL vocabularies.

Keywords: Background model, ontology coherence, OWL.

1 Introduction

The Linked Data (LD) initiative represents one of the first truly functional and proliferating incarnations of the generic concept of the Semantic Web, as a network of machine-processable data/knowledge. LD vocabularies typically come in the form of lightweight ontologies expressed in OWL or some of its sub-languages, and their development has rarely been guided by rigorous ontology design methodologies; it was more often a result of a consensus of a group of domain practitioners, perhaps with a few academicians involved.

Although we agree that such spontaneous development has its positive side, as it subscribes to the open nature of the Web environment, the resulting vocabularies may not always be well-aligned with the exact ontological nature of things that are being modeled. As an example consider the Music Ontology (MO) [8] which features a prop-

erty mo:primary_instrument[3] and introduces individuals such as mo-mit:Cello and mo-mit:Violin which are expected as its values. Such individuals, however are not respective to particular physical instruments, but instead refer to instrument types. They are, in fact, a form of classes modeled by ontology individuals. This does not cause any apparent problems in the MO vocabulary by itself. However, when the vocabulary is then linked to multitude of data sets by a number of users, it may well happen that one of the datasets records that, e.g., a particular musician "Yo-Yo Ma" uses the "1712 Davydov Stradivari" as his primary instrument. An incoherence thus arises between the intended usage of the mo:primary_instrument property and its actual usage in the data set. Such an incoherence may later cause problems especially when several data sets are combined, or a data set is transformed from one vocabulary to another.

We hence feel that some aspects of solid ontological modeling deserve to be 'injected' back into linked data vocabularies, as they could have positive effect on subsequent handling of such vocabularies as well as underlying data. However, the structure of popular vocabularies is already followed by masses of data. There might even be good reasons for this structure being as it is with respect to efficient handling of this data in data management tools. Hence, direct refactoring of these vocabularies is usually unfeasible. We thus choose to rely on the *annotation* paradigm: assigning labels to the entities from linked data vocabularies that indicate useful ontological distinctions.

A well-known approach in this respect is OntoClean [3], which allows to assign a set of predefined meta-properties (e.g., rigidity, anti-rigidity, identity, and unity) to ontological classes. The meta-properties are then used in taxonomy refactoring, according to certain integrity constraints (e.g., a rigid class cannot be subclass of an anti-rigid class). In the rest of this paper, we denote the syntactical, visible model of a domain (an OWL vocabulary) as an *ontological foreground model* (OFM), while the structure of ontological distinctions behind that model (such as the associated structure of OntoClean meta-properties) as the corresponding *ontological background model* (OBM).

While OntoClean can be seen as a suitable OBM language for large taxonomy-oriented ontologies (e.g, such as those found in medical domains), we feel that a different approach is needed in case of LD vocabularies. These vocabularies are more fact-oriented (their taxonomies are often simplistic) and OntoClean labels such as rigidity or anti-rigidity are not easy to understand by their creators who are often domain practitioners and not ontology experts. In this paper we propose a novel OBM language, dubbed PURO, whose focus is especially on the distinction between *particulars* vs. *universals* and *relationships* vs. *objects* (hence its name). PURO introduces a terminology of labels such as *object*, *relationship*, and *type*, which should be intuitively understandable for users already acquainted with OWL.

We anticipate several kinds of benefits gained from associating appropriate PURO OBMs to linked data vocabularies (as OFMs): detection of conceptual incoherence within individual vocabularies; more principled design of new vocabularies; faster adoption of existing vocabularies thanks to insight into their inherent structure (otherwise obscured by syntactical details of the OFMs); and prevention of mismatches when using multiple vocabularies in combination.

[3] mo=http://purl.org/ontology/mo/
mo-mit=http://purl.org/ontology/mo/mit#

	***O**bject*	***R**elationship*	*Valuation*
***P**articular*	$\mathcal{B}$-object (individual)	$\mathcal{B}$-relationship (3 options)	$\mathcal{B}$-valuation (data prop. assert.)
***U**niversal*	$\mathcal{B}$-type (class)	$\mathcal{B}$-relation (3 options)	$\mathcal{B}$-attribute (data property)

Fig. 1. Basic terms of the PURO OBML (and their corresponding foreground notions)

The contribution of the paper is as follows: Section 2 introduces the PURO OBM language. Section 3 concentrates on checking coherence of OFMs w.r.t. a PURO OBM. PURO coherence constraints are defined here and the PURO model is formalized as an OWL ontology (a meta-ontology, in fact). The ontology can be associated with an OFM through annotation of OFM entities with a distinguished set of labels. The coherence constraints are formalized as OWL axioms. A metamodeling-based coherence checking procedure is then shown including a short example. Assumptions of the approach and its overall feasibility are discussed in Sect. 4. A short related work survey and concluding remarks follow in Sects. 5 and 6.

2 The Language of PURO Ontological Background Models

A PURO ontological background model makes two basic distinctions: the one between ontological *particulars* and *universals*, and the one between *objects* and *relationships*. Particular and universals are distinguished by the possibility of instantiation: A particular cannot have instances, while a universal possibly can. As for the R-O distinction, objects are singular entities with their own identity, while a relationship cannot be considered (and talked about) without also considering the entities on which it depends. LD vocabularies prominently feature assignments of (quantitative) data values to individuals. Such assignments are not proper relationships, and thus we distinguish them as a third option within the R-O distinction: *valuation*.

The P-U and R-O(-V) distinctions are orthogonal, thus creating a two-dimensional space of the size 2×3. The language of PURO ontology background models (PURO OBML) contains six basic terms (primitives) in each point of this space (cf. Fig. 1). Each of the PURO terms can be associated with an OFM entity in order to clarify its ontological background.

2.1 Basic Terms of the PURO OBML

We now describe PURO terms in detail. To avoid confusion with OFM terms, all OBM terms are prefixed with $\mathcal{B}$- and we intentionally use different words (object, type, relation, etc.) than OWL (which uses individual, class, property, etc., respectively). We start with the P-U distinction applied to objects:

$\mathcal{B}$-object refers to a particular object, typically a real-world one, which can be tangible (such as people, animals, things, etc.) or intangible (such as topics, events, processes, abstract collections of information, etc.). It is analogous to the notion of *individual* in the foreground model.

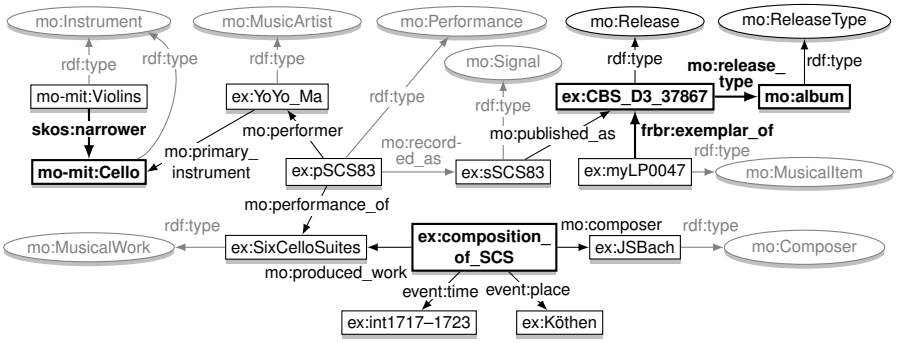

Fig. 2. Music Ontology data about myLP0047

$\mathcal{B}$-type refers to a universal, a set of background entities instantiating the same concept or sharing a common property. For simplicity, it also covers the notion of *quality* (e.g., red color), which is ontologically slightly different but plays the same role in the model structure. $\mathcal{B}$-types generalize the foreground notion of *classes*. A $\mathcal{B}$-type is *homogeneous* in the sense that its instances are PURO model entities of the same kind. Homogeneity leads to stratification of $\mathcal{B}$-types: instances of *1st-order $\mathcal{B}$-types* are $\mathcal{B}$-objects, instances of *n^{th}-order $\mathcal{B}$-types* are $(n-1)^{\text{th}}$-order $\mathcal{B}$-types for $n > 1$. Thus, unlike most description logics, the PURO admits also higher-order entities.

Example 1. Our music collection includes an LP from the CBS 1983 release of the same-year recording of YoYo Ma's performance of J. S. Bach's 'Six Cello Suites'. A possible Music Ontology foreground model of this situation is depicted in Fig. 2.

The foreground individual ex:myLP0047 is an instance of the class mo:MusicalItem. In the PURO background model of MO, ex:myLP0047 is a $\mathcal{B}$-object, and mo:MusicalItem is a 1^{st}-order $\mathcal{B}$-type. The individual ex:CBS_D3_37867 is an instance of mo:Release. As such, it describes common properties of all physical LPs, mo:MusicalItems, from the release. Furthermore, this particular release is assigned a mo:ReleaseType of mo:album, which is a foreground individual. Thus, mo:MusicalItem and ex:CBS_D3_37867 are both 1^{st}-order $\mathcal{B}$-types with different foreground representations; similarly, mo:Release and mo:album are 2^{nd}-order $\mathcal{B}$-types; and mo:ReleaseType is a 3^{rd}-order $\mathcal{B}$-type.

Let us now apply the P-U distinction to relationships:

$\mathcal{B}$-relationship refers to a particular relationship between two or more background entities (e.g., an object is produced by some producer; an object is of a certain type, one type of goods is a special case of another; or a vendor exclusively supplies a customer with some type of goods). $\mathcal{B}$-relationship generalizes several foreground notions: object property assertion, instantiation, and inter-class axioms. $\mathcal{B}$-relationships can also be n-ary for $n > 2$. We further discuss various kinds of $\mathcal{B}$-relationships below in Section 2.2.

$\mathcal{B}$-relation refers to a conceptual relation, the universal counterpart of $\mathcal{B}$-relationship. It is analogous to the foreground notion of *object property* without limiting the arity to two, and the domain and range(s) to $\mathcal{B}$-objects. Set-theoretically, a $\mathcal{B}$-relation is a subset of a Cartesian product of two or more sets of entities. Each of these sets must be homogeneous just like a $\mathcal{B}$-type.

Finally, applying the P-U distinction to valuations yields two more terms:

$\mathcal{B}$-valuation refers to a particular assignment of a quantitative data value to an entity (most often, but not exclusively, to a $\mathcal{B}$-object). $\mathcal{B}$-valuation is similar to $\mathcal{B}$-relationship, but it has a data value on the right hand side (e.g., the duration of a recorded signal is 7806 sec). $\mathcal{B}$-valuation is analogous to the foreground notion of *data property assertion*.

$\mathcal{B}$-attribute refers to a universal consisting of valuations of the same quantitative property. The analogous notion in the foreground model is *data property*.

Data values in a PURO background model are analogous to data values in foreground models. However, we only consider quantitative data values in the background model. Qualitative foreground data values may usually be reduced to $\mathcal{B}$-types as hinted above, e.g., a $\mathcal{B}$-type of $\mathcal{B}$-objects having red color.

2.2 Relationships in PURO OBMs

$\mathcal{B}$-relationships are the least uniform kind of entities in PURO background models, and require further discussion. We distinguish the following kinds of $\mathcal{B}$-relationships:

$\mathcal{B}$-instantiation is a relationship between an entity and a $\mathcal{B}$-type that the entity *instantiates*. A $\mathcal{B}$-instantiation intuitively means that the entity belongs to the given type, and it is a background analogue of an rdf:type statement. Unlike in foreground models either a $\mathcal{B}$-object, a $\mathcal{B}$-type, or sometimes even a $\mathcal{B}$-relationship can appear as the entity on the left-hand side. In the latter case, the right-hand-side entity is a higher-order $\mathcal{B}$-type.

A canonical foreground manifestation of $\mathcal{B}$-instantiation is an rdf:type statement. When a $\mathcal{B}$-type is represented by a foreground individual (e.g., ex:CBS_D3_37867 and mo:album in Fig. 2), $\mathcal{B}$-instantiation into this $\mathcal{B}$-type is manifested as an assertion of an object property (e.g., frbr:exemplar_of and mo:release_type). We remark that while in the former case the OFM and the OBM are analogous, in the latter case a clear dichotomy between the two models is apparent. Finally, some LD vocabularies allow, perhaps for increased flexibility, to assign categories to various objects via data properties with string or literal ranges (e.g., gr:category[4] in the GoodRelations (GR) vocabulary [4]). From the point of view of the PURO model, categories are $\mathcal{B}$-types, and so, in this case, $\mathcal{B}$-instantiation is manifested as a data property assertion.

$\mathcal{B}$-axiom is a $\mathcal{B}$-relationship between two $\mathcal{B}$-types, expressing a set-theoretic relationship between their extensions (such as subsumption or disjointness). A $\mathcal{B}$-axiom is canonically manifested in the foreground model using the corresponding foreground axiom, e.g., class subsumption expressed by an rdfs:subclassOf statement between two classes, each corresponding to a $\mathcal{B}$-type. If $\mathcal{B}$-types are represented as foreground individuals, their mutual relationship can be expressed using object properties. An example is found in the MO-recommended SKOS[5] taxonomy of instruments. Its fragment is shown in the left part of Fig. 2. The terms of this taxonomy represent $\mathcal{B}$-types of musical instruments. Hence, in this case,[6] the skos:narrower property is a manifestation of a subsumption $\mathcal{B}$-axiom.

[4] gr=http://purl.org/goodrelations/v1#

[5] skos=http://www.w3.org/2008/05/skos#

[6] Note that skos:narrower and skos:broader properties do not necessarily correspond to class subsumption since their meaning, as defined in the SKOS vocabulary, is slightly broader. How-

$\mathcal{B}$-fact is a $\mathcal{B}$-relationship that cannot be classified as either a $\mathcal{B}$-instantiation or a $\mathcal{B}$-axiom. One participant in a $\mathcal{B}$-fact is typically a $\mathcal{B}$-object. If the other is a $\mathcal{B}$-type, the $\mathcal{B}$-fact is called *heterogeneous*. $\mathcal{B}$-facts are canonically manifested in the foreground model as object property assertions. In Fig. 2, ex:pSCS83 mo:performer ex:YoYo_Ma represents a homogeneous $\mathcal{B}$-fact, while assertions ex:YoYo_Ma mo:primary_instrument mo-mit:Cello, and ex:sSCS83 mo:published_as ex:CBS_D3_37867 represent heterogeneous $\mathcal{B}$-facts. In some cases, $\mathcal{B}$-objects are represented as data values (e.g., regions can be encoded as strings "DE", "US-CA", etc.). A $\mathcal{B}$-fact involving such an object is manifested as an assertion of a data property (e.g., gr:eligibleRegions).

The most complex manifestation of a $\mathcal{B}$-fact is the reified relationship pattern. Here, a foreground individual reifies the $\mathcal{B}$-fact, and it is connected to the participants in the $\mathcal{B}$-fact by additional (object or data) property assertions. Reification is the only way of expressing n-ary relationships in LD vocabularies, since OWL does not support native n-ary constructs [12,6]. Figure 2 depicts an mo:Composition individual ex:composition_of_SCS, which can be seen as a reified $\mathcal{B}$-fact among a composer, his musical work, place, and time of composition. It should be noted that the boundary between (complex) relationships and (intangible) objects may not be sharp in some situations. For instance, a composition relationship can be also viewed as a composition event. Indeed, mo:Composition is a subclass of event:Event.[7] We discuss this issue in Section 4 below.

Due to the focus of this paper and space constraints, we have only described the various kinds of $\mathcal{B}$-relationships briefly. A detailed discussion with examples from LD vocabularies is available in [10].

3 PURO Model Coherence Checking

Clearly, the combination of OBM distinctions underlying a particular OFM cannot be arbitrary. For example, as mentioned in the introduction, if a class is labelled as rigid according to OntoClean, it cannot be subclass of a class labelled as anti-rigid. Violating such assumptions can be viewed as OBM-level incoherence. Note that the notion of OBM incoherence is only very loosely associated with that of logical inconsistency in principle. Yet, as we demonstrate later, the task of OBM coherence checking can be transformed into DL consistency checking, through metamodeling.

We will now define when a PURO OMB is coherent (Sect. 3.1), and formalize the PURO language and OBM coherence constraints in an OWL DL ontology (Sect. 3.2). That will enable us to define and demonstrate a method of automatic background model coherence checking of OFMs annotated with PURO OBML constructs (Sects. 3.3, 3.4).

3.1 Background Model Coherence

A PURO OBM model is said to be *coherent* if it satisfies all of the following three constraints:

ever in some LD vocabularies (such as mo-mit) which contain classes reifed as objects they are used to indicate subsumption.

[7] event=http://purl.org/NET/c4dm/event.owl#

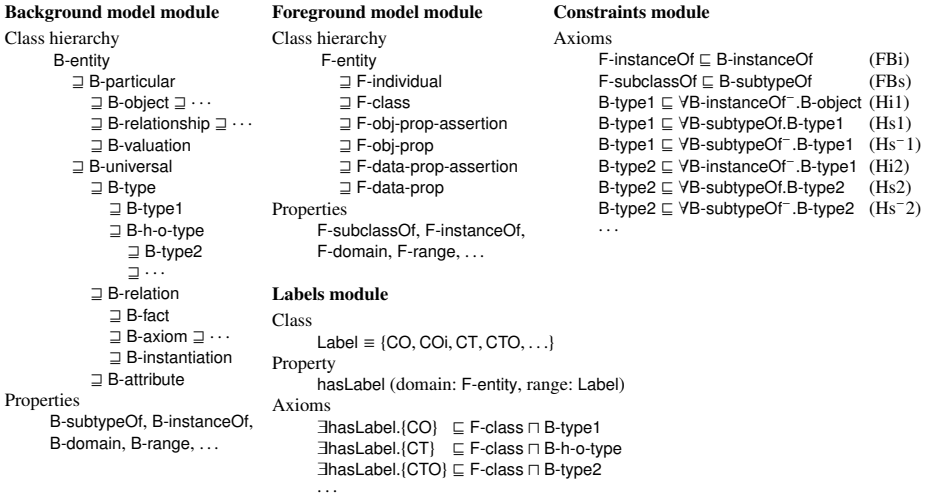

Fig. 3. Fragments of the PURO ontology modules.

entity coherence: the sets of $\mathcal{B}$-objects, $\mathcal{B}$-valuations, $\mathcal{B}$-relationships, n^{th}-order $\mathcal{B}$-types for each $n \geq 1$, $\mathcal{B}$-relations, and $\mathcal{B}$-attributes are pairwise disjoint;

type homogeneity: each $\mathcal{B}$-type is a homogeneous n^{th}-order $\mathcal{B}$-type, for some $n \geq 1$, as defined in Sect. 2.1;

relation homogeneity: the domain (the range) of each $\mathcal{B}$-relation is a homogeneous n^{th}-order (m^{th}-order) $\mathcal{B}$-type for some $n \geq 1$ ($m \geq 1$).

3.2 PURO Ontology

The language of PURO background models can be expressed as an OWL DL ontology. As this ontology talks about entities from different ontologies (i.e., the OFMs) it is in fact a meta-ontology. Within it, we can formally postulate coherence constraints introduced in the previous section. The main advantage of the PURO ontology lies in automation of incoherence detection: The ontology can be populated with entities resulting from metamodeling of some LD vocabulary and with annotations of those entities with PURO constructs. We are then able to detect incoherences in the vocabulary at the PURO background level using a regular ontology reasoner. The process of coherence checking is described in Sect. 3.3.

The ontology consists of four partially dependent modules: 1) background model, 2) foreground model, 3) labels, 4) constraints. Fragments of all modules are shown in Fig. 3 (we rely on the description logic syntax in order to improve readability).

The *background model* module defines the hierarchy of classes and properties representing the terms of the PURO OBML (e.g., B-object, B-type, etc.). Similarly, the *foreground model* module defines a much simpler hierarchy of concepts and properties representing the language terms of foreground models (e.g., F-individual, F-class, etc.). The *labels* module defines a set of annotation labels, represented as individuals grouped in the Label class. The labels have acronyms consisting of an initial letter denoting the

syntactic entity that is annotated (C for class, I for individual, Pr for property range, Pd for property domain) and of a sequence of letters determining the actual PURO concept.

Frequently used labels for classes thus are CO (Class whose instances correspond to $\mathcal{B}$-Objects), CR (Class whose instances correspond to $\mathcal{B}$-Relationships), CV (Class whose instances correspond to $\mathcal{B}$-Valuations), or CT (Class whose instances correspond to $\mathcal{B}$-Types), further refined to CTO (Class whose instances correspond to $\mathcal{B}$-Types of Objects). Analogously, property ranges can be labeled by PrO, PrR, PrV or PrT, and individuals by IO, IR, IV or IT.[8] The labels module also expresses the semantics of the labels in terms of axioms on top of the languages of background and foreground models. For instance, an entity annotated with the label CTO is a foreground class, and a $\mathcal{B}$-type of $\mathcal{B}$-types of $\mathcal{B}$-objects, i.e., a 2nd-order $\mathcal{B}$-type.

Finally the *constraints* module expresses conditions that hold in a background model. For instance, axioms (Hi1, Hs1, Hs$^-$1, ...) assert that $\mathcal{B}$-types are homogeneous. This module also connects properties conjoint in the background and the foreground models (FBi, FBs), such as instantiation or subsumption.

3.3 Vocabulary Meta-Modeling and Coherence Checking

We would like to be able to automatically check the background coherence of LD vocabularies, once they have been annotated by the labels respective to the PURO OBM. Welty [13] and Glimm et al. [2] rely on metamodeling and ontology reasoners for verification of OntoClean constraints. With use of the PURO ontology introduced in the previous section we are able to take a similar approach.

The inputs of coherence checking are an LD vocabulary and its annotation. This input needs to be pre-processed before checking as follows:

1. Obtain a deductive closure T of the checked vocabulary.
2. Create a metamodel T' of the closure T within the foreground model module, as follows: Let T_C, T_P, T_I be the sets of classes, properties, and individuals of T respectively. The metamodel contains individuals c_C, p_P, and i_j (as 'proxies' of the respective vocabulary entities), together with assertions F-class(c_C), F-obj-prop(p_P), and F-individual(i_j), for each $C \in T_\mathrm{C}$, $P \in T_\mathrm{P}$, and $j \in T_\mathrm{I}$ respectively.
   Each assertion $C(j)$ in T is then metamodeled as F-instanceOf($\mathsf{i}_j, \mathsf{c}_C$) in T', and each subsumption $C \sqsubseteq D$ as F-subclassOf($\mathsf{c}_C, \mathsf{c}_D$). Property domains, ranges, subproperty axioms, and property assertions can be metamodeled similarly.
3. Assert annotation labels in T' explicitly using the labels module: If an entity E of T is annotated with a label L, then we have hasLabel(e, L) in T' where e is the individual metamodeling E.

Coherence of the original vocabulary can now be checked by reasoning over the ontology obtained as the union of the metamodel T' and the four modules of the PURO ontology. The fragment of the PURO ontology shown in Fig. 3 enables detection of the following incoherences:

[8] In addition to labels corresponding to PURO concepts, there are also labels that indicate foreground data without ontological background relevant for a PURO model. We omit them here as they are not relevant for coherence checking; more information is in [10].

– If a foreground entity E (typically an individual) metamodeled as e is directly or indirectly labeled as representing both an ontological particular and a universal, then we recognize this incoherence due to membership of e in the meta-class:

$$\mathsf{Incoherent\text{-}PU} \equiv \mathsf{B\text{-}universal} \sqcap \mathsf{B\text{-}particular} \; .$$

– If a foreground class C is labelled as having both $\mathcal{B}$-objects (CO) and 1st-order $\mathcal{B}$-types (CTO) as instances, it cannot represent a proper homogeneous $\mathcal{B}$-type. This kind of incoherence can be detected due to axioms (Hi1) and (Hi2) implying

$$(\forall \mathsf{B\text{-}instanceOf}^{-}.\mathsf{B\text{-}object} \sqcap \mathsf{B\text{-}type1})(\mathsf{c}_C) \; .$$

  In fact, this incoherence is a special case of a more general situation, in which a class (or another foreground entity) is labelled as having both particulars and universals as its instances. Such an entity (e.g., c_C) falls into the meta-class:

$$\begin{aligned} \mathsf{Incoherent\text{-}T\text{-}PU} &\equiv \mathsf{B\text{-}type} \sqcap \forall \mathsf{B\text{-}instanceOf}^{-}.\mathsf{B\text{-}particular} \sqcap \mathsf{B\text{-}universal} \\ &\equiv \mathsf{B\text{-}type} \sqcap \forall \mathsf{B\text{-}instanceOf}^{-}.\mathsf{Incoherent\text{-}PU} \; . \end{aligned}$$

  Note that axioms (Hs1, Hs^{-}1, Hs2, Hs^{-}2) propagate the level of $\mathcal{B}$-types over subsumption. If a class has two subclasses with instances of different background kinds, it will be recognized as incoherent.

More kinds of incoherence (e.g., involving higher-order $\mathcal{B}$-types, or ranges of properties) can be detected with a suitable combination of constraints and definitions of meta-classes of incoherent entities.

Our approach not only *detects* that a foreground vocabulary is incoherent, but also allows to *explain* which kind of incoherence was detected, and which foreground model entities caused it, since they are instances of the respective Incoherent- meta-class.

3.4 Coherence Checking Demonstration

Let us now demonstrate coherence checking on the GR vocabulary. The vocabulary contains a class gr:ProductOrService with three subclasses: gr:Individual, gr:ProductOrServiceModel, gr:SomeItems. The subclass gr:Individual is easily recognized as a class of $\mathcal{B}$-objects and thus annotated as CO. The subclass gr:ProductOrServiceModel represents models, i.e., types of products, hence its annotation label is CTO.

If we feed this fragment of the GR vocabulary and the annotation into the coherence checking mechanism from Sect. 3.3, we obtain the following results: From the annotation of gr:Individual and gr:ProductOrServiceModel, the labels module allows us to derive $\mathsf{B\text{-}type1}(\mathsf{c}_{\mathsf{gr:Individual}})$ and $\mathsf{B\text{-}type2}(\mathsf{c}_{\mathsf{gr:ProductOrServiceModel}})$. Constraints module axioms (Hs1, Hs2) propagate the background meta-classification to the common superclass, and so we obtain $\mathsf{B\text{-}type1}(\mathsf{c}_{\mathsf{gr:ProductOrService}})$ and $\mathsf{B\text{-}type2}(\mathsf{c}_{\mathsf{gr:ProductOrService}})$. Since instances of a B-type1 are particulars ($\mathcal{B}$-objects), and instances of a B-type2 are universals (1st-order $\mathcal{B}$-types), the $\mathsf{c}_{\mathsf{gr:ProductOrService}}$ is an instance of Incoherent-T-PU.

The version of PURO ontology and the GR fragment used in the demonstration is available from `http://patomat.vse.cz/puro_v1.owl` and `http://patomat.vse.cz/gr_mm.owl`, respectively.

4 Discussion

The whole approach has several critical assumptions: 1) given an OFM properly annotated with PURO labels, incoherence can be detected via reasoning; 2) PURO OBM entities can be identified based on the structure and documentation of an OFM with reasonable accuracy; 3) the structure of the OFM and the respective PURO OBM significantly differs in a number of OWL vocabularies.

In the previous section, we have demonstrated the validity of the first assumption, though on a tiny example only. We however do not expect computational complexity issues even when processing complete LD vocabularies since mostly their size is rather small. As only the vocabulary needs to be annotated, the underlying datasets which are often large do not impact the effectivity of coherence checking.

To verify the remaining two assumptions we have surveyed a number of LD vocabularies (in business, government, geography, and other domains) and conducted an annotation experiment. We summarize the results as follows (for details see our technical report [10]): In 92 out of 94 syntactical classes in 3 popular vocabularies, two annotators (both expert ontologists) found a clear consensus on the PURO OBM category. Out of these 94 classes, only 59 were labeled as CO (i.e., matching the OFM structure).

We found that, in large majority, LD vocabularies anticipate facts about *particulars* (e.g., real persons, organizations, items of goods, documents, etc.) which can be reliably distinguished from the universals in the domain, whatever their syntactical way of modeling is. However, this distinction may be blurred in biomedical and other scientifically-biased ontologies, which often feature individuals acting as prototypes (e.g., cells, organs, chemical processes, etc.). As PURO model is specifically designed for LD vocabularies, the latter type of ontologies is not its primary application target.

The boundary between *relationships* and *objects* may not be as sharp in some situations. Due to absence of n-ary relations, LD vocabularies often feature reifying individuals, e.g., the instances of mo:Composition in MO (relation between musical work, its composer, place, and time) or gr:Offering in GR (relation between a seller, offered items, eligible regions, etc.). Such objects can be viewed as $\mathcal{B}$-relationships but often also as regular $\mathcal{B}$-objects – the event of composition, an abstract information record. The distinction is often a modeling decision (even at the background level) and it should be based on the criterion whether the (reifying) object would be meaningful even without explicitly considering the other participants in the relationship. In these terms, a composition event is clearly incomplete without knowing what was composed, similarly a business offering is incomplete without knowing the seller or the items offered. More detailed analysis of 'R vs. O' nuances is ongoing work.

5 Related Research

A prominent example of a different OBML (and in fact the only one known to us) is certainly OntoClean [3], which differs from our approach in its focus on annotating classes and repairing their taxonomic relationships. In contrast, PURO focuses on instantiations and facts, and its purpose is not necessarily repair. OntoClean also relies on conceptual

notions (such as rigidity) that are fundamentally different from what LD practitioners typically know from the foreground representation; this also holds for works that map ontological roles into OWL [9]. PURO, on the other hand, introduces new primitives which are more familiar (e.g., object, relationship, type, etc.).

Checking OntoClean constraints and similarly PURO coherence involves reasoning with higher orders. Though allowed to a certain extent in OWL Full, it is however not feasible in practice [5]. Motik [5] and De Giacomo et al. [1] proposed alternate semantics for higher-order description logics while maintaining decidability. Similarly Pan and Horrocks [7] proposed a variant of RDFS semantics (dubbed RDFS(FA)), in which types are divided into multiple 'strata', analogous to type orders in the PURO model.

Welty [13] and Glimm et al. [2] use metamodeling inside OWL in order to automatically verify OntoClean constraints. We have taken a similar approach in Sect. 3, where we have metamodeled the foreground ontology and expressed constraints that are necessary to check the PURO OBM model coherence, which can then be done using a regular OWL reasoner. Our approach is closer to the one of Welty, as we require a preprocessing step, in which the deductive closure of the foreground ontology is computed before the meta-level reasoning is applied. We are currently investigating the possibility of performing both levels of reasoning at the same time, similarly to Glimm et al.

The ontological distinctions recognized by PURO, albeit being rather simplistic, are also found in many foundational ontologies (e.g., DOLCE[9] or BFO[10]). Foundational ontologies are intended to be directly combined with ontologies at the same level of reasoning. The PURO model (and the whole OBML paradigm) significantly differs in that it also (largely) focuses on use cases in which we do not intend to restructure the associated OFM models, but we simply want to make the ontological background in them obvious so that their understanding and manipulation is facilitated. Therefore, the coupling methods between OFMs and OBMs differ from the ones of foundational ontologies (e.g., annotation and metamodeling, not direct 'injection' into the OFMs).

6 Conclusions and Future Work

PURO is a new OBM language aimed to aid in the process of development, debugging, visualization, and matching of operational ontological models. PURO is specifically aligned with LD vocabularies, focusing especially on the particular–universal and relationship–object distinctions, relying on constructs that should be rather familiar to users acquainted with basic OWL terminology. A pairing of a vocabulary and a PURO OBM is established using annotation with a predefined set of labels. The language also features a set of constraints, allowing for coherence checking w.r.t. the PURO model associated via annotation. With help of metamodeling an annotated LD vocabulary can be formally paired with the PURO meta-ontology and the coherence check can then be automated relying on available OWL reasoners in terms of a consistency check of the resulting ontology.

We see two main advantages of the outlined approach: It allows ontology developers 1) to verify the level of ontological relevance of their models, and 2) to annotate their

[9] `http://www.loa.istc.cnr.it/DOLCE.html`
[10] `http://www.ifomis.org/bfo`

vocabularies with ontological distinctions inexpressible in OWL, thus communicating intentions behind their modeling decisions. Although this paper has mainly focused on the former of these advantages, also the latter is relevant. It is not always practical, or even possible, to develop ontologically pure LD vocabularies, as there are justified design decisions and language constraints leading to simplified models. It is thus useful to keep track of the resulting ontologically ill-aligned model features (e.g., classes modeled as individuals, instantiations modeled as object properties, etc.), since they can gradually lead to incoherence, especially when a vocabulary is paired with various different data sets, which may each follow a slightly different modeling approach. The PURO models may turn into a useful tool to guide this process. In order to further aid its applicability, we are currently developing an annotation tool integrated with Protegé. This tool will allow for easy annotation of vocabularies, and also of data sets using a vocabulary, and automated coherence checking (see [10]).

Acknowledgements. This work was supported from the EU ICT FP7 under no. 257943 (LOD2 project), from the Slovak VEGA project no. 1/1333/12, and from the Czecho-Slovak bilateral project nos. 7AMB12SK020 (AIP) and SK-CZ-0208-11 (APVV).

References

1. De Giacomo, G., Lenzerini, M., Rosati, R.: On higher-order description logics. In: Proc. DL 2009.
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. Hepp, M. *GoodRelations Language Reference*. Version 1.0, released Oct 1, 2011. Available onilne: `http://purl.org/goodrelations/v1`
5. Motik, B.: On the properties of metamodeling in OWL. *J. Log. Comput.* 17(4):617–637, 2007.
6. Noy, N., Rector, A. (eds.): *Defining N-ary Relations on the Semantic Web.* W3C Working Group Note 12 April 2006, online at `http://www.w3.org/TR/swbp-n-aryRelations/`.
7. Pan, J.Z., Horrocks, I.: RDFS(FA): Connecting RDF(S) and OWL DL. *IEEE Trans. Knowl. Data Eng.* 19(2):192–206, 2007.
8. Raimond, Y., Giasson, F. (eds.): *Music Ontology Specification.* Released Nov 28, 2010. Available online: `http://purl.org/ontology/mo/`.
9. Sunagawa E., Kozaki K., Kitamura Y., Mizoguchi R.: Role organization model in Hozo. In: Proc. EKAW 2006.
10. Svátek, V., Homola, M., Kľuka, 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`
11. Svátek, V., Homola, M., Kľuka, J., Vacura, M.: Mapping Structural Design Patterns in OWL to Ontological Background Models. In: Proc. K-CAP 2013.
12. W3C OWL Working Group (eds.): *OWL 2 Web Ontology Language Document Overview.* W3C Recommendation, 2009.
13. Welty, C.: OntOWLClean: Cleaning OWL ontologies with OWL. In: Proc. FOIS 2006.