Semantic Web 0 (0) 1
IOS Press

PURO: Capturing the Background of OWL Ontologies

Vojtěch Svátek [a,*], Martin Homola [b], Ján Kľuka [b], Marek Dudáš [a] and Miroslav Vacura [a,c]
[a] *Dept. of Information and Knowledge Engineering, University of Economics, Prague, Nám. W. Churchilla 4, 130 67 Praha 3, Czech Republic*
E-mail: {svatek, marek.dudas}@vse.cz
[b] *Department of Applied Informatics, Comenius University in Bratislava, Mlynská dolina, 842 48 Bratislava, Slovakia*
E-mail: {homola, kluka}@fmph.uniba.sk
[c] *Dept. of Philosophy, University of Economics, Prague, Nám. W. Churchilla 4, 130 67 Praha 3, Czech Republic*
E-mail: vacuram@vse.cz

Abstract. Many modeling decisions in OWL ontologies are enforced by formal constraints of the language or by pragmatic application concerns, and thus deviate from faithful modeling of reality in terms of the Universal/Particular and Relationship/Object distinctions. The proposed PURO language aims to complement OWL in this respect via providing a conceptual layer relaxing some of the OWL constraints, while it remains simple in terms of its basic repertory of constructs. We formulate the notion of ontological background model (OBM) and demonstrate how models expressed in PURO can serve as OBMs for OWL ontologies. We also provide a full first-order axiomatization of the stable core of PURO, interrelate and exemplify its primitives, and enumerate their typical OWL manifestations. Further, we analyze a number of PURO use cases in various stage of maturity, and elaborate on several evaluation experiments corroborating the usability of the PURO language. Connections to related research such as OntoClean or OntoUML are also established.

Keywords: Ontological Background Language, PURO, OWL, Ontology Design, Pattern, Transformation

1. Introduction

While several highly expressive ontological formalisms have been proposed in the knowledge representation (KR) community in the last few decades, currently the most widespread ontology language is OWL [62,12], which arose in the context of the semantic web movement as a means to providing context for RDF [13] data. The repertory of description logic (DL) axioms in OWL is relatively large; yet, a large proportion of OWL ontologies, particularly, what is called *linked data vocabularies* [83], only use a small fragment of OWL expressiveness going little beyond RDF Schema; see the study underlying the OWL LD fragment [31].

The rapid and wide spreading of OWL may be largely due to the simplicity of its core inventory of building blocks: the 'class – (object or datatype) property – individual' triad. The ontology designer is thus not forced to contemplate about the deeper nature of the modeled entities: in contrast to full-blown KR and conceptual modeling languages (e.g., OntoUML [37]), its metamodel does not explicitly distinguish between constructs such as partonomic relationships, roles, qualities, events, etc., which are only modeled in OWL, upon discretion, at the level of particular ontologies. Nothing beyond classes, properties, and individuals thus can be chosen at the atomic level of discernment. Intuitively, classes should model *universals* (corresponding to 'types of things'), individuals should model *particulars* (corresponding to 'individual things' that do not themselves have instances), object

*Corresponding author. E-mail: svatek@vse.cz

properties should model *relationships* holding among either universals or particulars, and data properties should ideally focus on capturing *quantitative values* (since qualitative 'values' are typically objects that may deserve their own IRIs as individuals), or data items such as labels (which, in turn, do not have their 'tangible' referent in the real world).

However, encoding the intended model in OWL often suffers from more severe constraints than those incurred purely by the limited triad as such. The decidability and computational effectiveness of description logic reasoning operations led the designers of the OWL standard to disallowing metaclassing ('types of types') in the mainstream version of the language (OWL DL), and, in particular, relationships ('properties') with arity higher than 2 are completely outside its scope. Ontology designers thus not only have to content themselves with that triad, but cannot even choose within it freely; possibly, e.g., an individual has to be used for modeling a universal or a relationship. In a sense, OWL DL thus stands 'two steps away' from languages that both (1) provide a richer inventory of basic constructs, and (2) allow to assemble them in a more flexible way and thus, ideally, more faithfully to reality.

To illustrate this phenomenon, let us consider two interconnected examples of situations to be modeled as RDF data fragments referring to OWL entities from relevant ontologies.

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 OWL model of this situation, based on the Music Ontology (MO)[1] and individuals from the DBpedia dataset,[2] is depicted in Fig. 1 together with some context.

Example 2 EMI offers unspecified exemplars of the LP records from Example 1 identified by its EAN code 8707464378677 for sale in the United States. The records' total play time of 130 minutes is specified in the offering. An OWL model of this situation using a combination of the GoodRelations (GR)[3] ontology and MO, plus again DBpedia, is depicted in Fig. 2.

The emphasized portions of the figures indicate where the modeling is potentially skewed by the constraints of OWL or other factors:

1. The left-hand side of Fig. 1 refers to Cello being the performer's primary instrument. The nature of the Cello entity here is that of a universal (abstract 'instrument', a type) rather than of a particular (physical) instrument. However, in OWL DL, this universal, essentially, has to be modeled by an *individual*, `mo-mit:Cello`, so that it could appear as the object of a property.
2. In the right-hand side of Fig. 1 we see the `mo:album` entity, which, again, is clearly a universal, yet, modeled by an *individual*, since otherwise it wouldn't be possible to state that an album is a 'release type'. The relationship between a given album, `ex:CBS_D3_37867`, and the 'album concept', which should canonically be expressed via the `rdf:type` property, is circumscribed using a custom property, `mo:release_type`.
3. The bottom part of Fig. 1 primarily expresses the fact that 'Six cello suites' has been composed by Bach; however, some additional details (time and place) are specified about the composition process, and thus the relationship between the composer and the work has to be *reified* to an individual. Now the structure is however syntactically entirely different from just stating, via a property assertion, the original simple fact.
4. Analogously, the right-hand side of Fig. 2 aims to express that the company offers the album for sale at some territory. Offering goods for sale is in most contexts an n-ary relationship (with $n > 2$) and thus needs to be, again, *reified* (here as the `ex:o` individual).
5. Note the fact that the notion of 'territory of US' is, in this fragment, expressed using a *literal* (data value) rather then individuals with IRI identifiers, since `gr:eligibleRegions` is a data property. This modeling is not enforced by the OWL structures as such but by the preference of markup-oriented vocabularies such as GR to directly store human-readable values in property objects, to avoid indirection.

[1] See [65], prefixes `mo=http://purl.org/ontology/mo/`, `mo-mit=http://purl.org/ontology/mo/mit#`, `frbr=http://purl.org/vocab/frbr/core#` and `skos=http://www.w3.org/2008/05/skos#`.

[2] See `https://dbpedia.org`, prefix `dbr=http://dbpedia.org/resource/`

[3] See [43], prefix `gr=http://purl.org/goodrelations/v1#`.

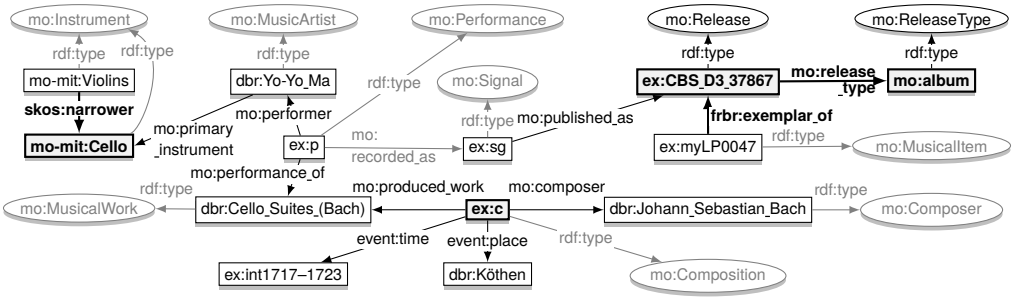

Fig. 1. Music-Ontology-based data about myLP0047.

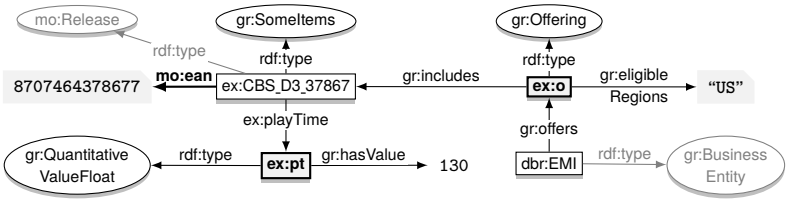

Fig. 2. GoodRelations-based data about an offer of some LPs for sale.

6. The bottom-left of the figure states the play time of the album. However, since the preferable way of modeling quantitative values in the GoodRelations vocabulary is that via instances of classes such as `gr:QuantitativeValueFloat`, such an *intermediate individual* is introduced there.
7. On the left-hand side of Fig. 2 we see the EAN code as a literal value. This is relatively OK, except that the EAN is, ontologically, not truly a 'property' of `ex:CBS_D3_37867` but rather its alternative *identifier*.

The first major challenge for OWL modeling, that of 'referring to classes as if they were instances' (items 1 and 2 above), can be technically handled using the so-called *punning*, namely, OWL in its version 2 allows for a class and an individual to share the same IRI identifier while being treated as separate at the inferential level.[4] Even this is far from elegant, since the IRI of the 'class' used as the property value is indeed, syntactically, that of an individual. The second OWL challenge, that of n-ary relationships (from items 3 and 4 above) can only be treated using design patterns [61], which are not a canonical part of the language.

The remaining issues are not caused by OWL as language but by application concerns. Inherently ontological entities may be expressed as mere 'data' (item 5), while true quantitative data values may require introduction of artificial objects (item 6). Finally, various properties expressing mere names or identifiers may populate the graph structure of data (item 7); while a data property is used here, there are even patterns that elaborate on codes, notations, etc., in the role of first-order citizens of ontologies, i.e. through object properties.[5]

It is perfectly possible to build ontologies using such techniques. However, the ease of use for the ontology designer somewhat fades out, and the ontological nature of the entities, in spite of the simplicity of the underlying 'triad' metamodel, becomes blurred. In the situation when the OWL code is 'its own model', ontologies risk to become hackers' business rather than a principled vehicle for communicating semantics among machines and humans alike. This caveat is what spurred us towards introducing a new KR language for the semantic web, which should not replace OWL by any means, but rather complement it in order to make ontology design and processing more transparent and (a tiny step) more faithful to deeper ontological distinctions.

Our proposed language, *PURO*, first informally drafted in 2013 [79], is analogous to OWL in its 'core triad', and thus potentially easy to grasp for developers already familiar with OWL, yet provides the flexibility needed for natural modeling of situations that

[4] The introduction of punning to OWL 2 has rendered the originally proposed design patterns for 'Classses as Property Value' [60] less relevant in most situations.

[5] See, e.g., SKOS, `http://www.w3.org/2008/05/skos#`.

are only encoded via patterns (such as 'reified relationship') or semantically problematic tricks (such as class-individual punning) in OWL. An obvious downside of this flexibility would be a high cost of inferential operations; PURO (despite its grounding in first-order logic) is however not meant as a fully operational language but primarily as a tool allowing for 1) better *human interaction* with semantic web ontologies, and 2) automatic checking of certain aspects of *conceptualization coherence*.

From the methodological point of view, PURO is intended to be applied to modeling not only at the schema level (TBox) but also to examples at the *instance level* (ABox), in a single representational artifact. This allows to express ontological structures in terms of *interconnected graphs* rather than mere lists/trees of classes and properties.

Aside its core metamodel, the design of PURO also anticipates a library of *alignment patterns* allowing, among other, to transform a PURO model to alternative OWL encodings in different styles (i.e., using different patterns).

The PURO eco-system already comprises a *suite of tools* for PURO authoring, PURO transformation to OWL, as well as annotation of OWL models with PURO distinctions. Specific studies have been carried out for different *use cases* of PURO, and in different subject domains. The understandability and usability of PURO has been *evaluated* in small user studies.

A number of research papers on partial aspects of PURO have been published between 2013 and 2016. What is however missing is a comprehensive text describing the core model in extenso, including a formal apparatus, and the modalities of aligning PURO patterns with OWL ontology patterns. As regards the tools, applications and evaluations, there are also several yet unpublished results, and the different components and studies lack a systematic, comparative overview. Delivering a coherent union of all this is the main purpose of the current manuscript.

The rest of the text is structured as follows. Section 2 introduces the general notion of *ontological background model* of which PURO is a particular instance, for a broader context. Section 3 formally defines the *PURO language* primitives (PURO terms), the notion of a *PURO model*, and also informally discusses the typical cases of PURO entity manifestations in OWL ontologies. Section 4 analyzes different modalities of the PURO structure and OWL structure *alignment*. Section 5 summarizes and systematically compares the different *use cases* for PURO considered so far. Section 6 reports on several complementary efforts in *evaluating* the viability of PURO-based ontological engineering. Section 7 discusses the crucial issues of PURO observed to date. Section 8 surveys the most important *related research*. Finally, Section 9 wraps up the paper and outlines prospects for future work.

2. Ontological background models

The PURO language as such can be described entirely generically, as a KR language based on an underlying formal model. However, the motivation for inventing a new, relatively simple, KR language when hundreds of others are already available, can be better explained by first introducing the notion of *ontological background model*, as well as that of *ontological background modeling language* of which PURO is a particular instance. Referring to this context, we will then, in Section 3, not only formally define the language itself but also outline its mapping to OWL as its primary *foreground* language. Namely, not only the PURO language itself but also its OWL alignment are both crucial components of the proposed approach. In this section we thus explain the notion of ontological background (as well as foreground) model.

Throughout its whole history, KR has sought balance between ontological validity and practical usability, the latter with respect to both efficient processing by machines and accessibility to people who are not particularly 'ontologistically minded'. The introduction of formalized ontologies into information systems put, to some degree, emphasis on the ontological validity, especially through the apparatus of foundational (or, upper-level) ontologies, which started to blossom after 2000 [55]. On the other hand, with the growing popularity of the semantic web, a large portion of new ontologies, such as linked data (LD) vocabularies, have been directly authored in OWL and thus influenced from the beginning by its inventory of constructs and limitations. Additionally, such vocabularies are often primarily designed with focus on particular application needs, and thus deviate from the most ontologically faithful patterns. We saw illustrations of both these effects (language limitations and pragmatic choices) in the motivation examples in Section 1.

Whether we start from a more 'foundational' or a more pragmatical model, we might eventually need *both*: the former for careful domain analysis, and the latter for achieving data structure simplicity. Under

specific conditions, we would then call the two models *ontological background model* (OBM) and *ontological foreground model* (OFM), respectively. The relationship between the two, $\mathcal{B}$-modeling (for 'modeling the background of'), can be semi-formally defined as follows:

Definition 1 ($\mathcal{B}$-modeling) *An ontological model*[6] M_B *of a domain* D $\mathcal{B}$-models *an ontological model* M_F *of the same domain* D *if all the following conditions hold:*

1. M_B *captures more closely than* M_F *the inherent state of affairs in domain* D*: the set of its biases is a (typically, proper) subset of that of* M_F.
2. *There exists at least a partial alignment* A *from the elements of* M_B *to the elements of* M_F.
3. M_B *is* not *integrated with* M_F *in a single representation space comprising* M_B, M_F, *and* A.

The definition is obviously far from operational and can be only used as approximate guidance.

The first condition refers to *biases* such as those exemplified in Section 1 (incurred either by the language in which M_F is expressed, or by some usage concerns). Their explicit enumeration may not always be within the reach of some deterministic algorithm; however, we can at least assume that M_B should not introduce an additional bias (distorting the faithful modeling of reality) that would not be present in M_F. The $\mathcal{B}$-modeling relationship also does not mean that M_B would have to preserve all the information present in M_F.

The second condition assumes an *alignment*, which provides a grounding for the $\mathcal{B}$-modeling relationship. The alignment A is not a part of either M_B or M_F as such. Formally, A, if viewed as a unidirectional mapping from M_B to M_F, need not be surjective ('onto'): typically, M_F might capture information that is not an inherent part of the state of affairs in the domain (e.g., assignment of identifiers to objects, for the sake of managing data on them) and would not thus be captured by M_B. On the other hand, M_B may cover information that is not explicitly present in M_F, typically due to the representational constraints of its language; therefore even the inverse mapping from M_F to M_B is not surjective.

The third condition has been added to draw a subtle distinction between OBMs and foundational/upper ontologies [55] such as DOLCE or UFO. A foundational ontology (FO) typically provides root concepts upon which the domain ontology concepts are grafted using logical structures of the ontological language itself (such as subclassing); both models thus share the same space and form, together with their alignment, a single (although modularized) homogeneous ontology. In contrast, OBMs reside in their own 'layer'; consequently, they do not influence reasoning inside the OFM, as FOs typically do. Informally, a FO also aims to capture the meaning of entities *at a more general level* (the revelation and subsequent elimination of biases only being a possible side-effect), while an OBM aims to capture the meaning of entities and their structures *more faithfully* (even at a similar level of generality). As regards the upper-level categories of FOs, they may be rather similar to the primitives used in OBMs; a slightly more detailed comparison is in Section 8.2.

Definition 2 (ontological background model) *An ontological model* M_B *is an* ontological background model *if it* $\mathcal{B}$*-models an* existing *or just* possible *ontological model* M_F .

Terminologically, we will also use the wording 'is an OBM of' interchangeably with '$\mathcal{B}$-models' from Def. 1, provided there is no risk of confusion with the unary notion of OBM from Def. 2. The inverse relationship then would be 'is an OFM of'.

We can imagine that an OBM $\mathcal{B}$-modeling some OFM might itself be $\mathcal{B}$-modeled by an even less biased (standing 'closer to the reality') model, which would be its OBM, in turn. (In this vein, we for example foresee the possible OBM role of OntoUML [37], with its broad repertory of foundational distinctions, wrt. PURO. More on this in the future work plan in Section 9.) However, if we simplify our view to two levels of ontological modeling, it can be schematized as in Fig. 3, with the amount of bias increasing from the inner part to the outer part. There are, typically, 1:n relationships between

- the modeled reality and the different OBMs: the n should, ideally, be small, but different views cannot be completely avoided even when aiming at maximally unbiased modeling,
- OBMs and OFMs,

[6] We intentionally leave the notion of *ontological model* (aka ontology) without formal definition, to avoid exclusion of less common KR languages. Intuitively, an ontological model should be a set of logical formulas expressing how entities (corresponding to something occurring in the real, or some imaginary, world) are interconnected with relationships; at least some of the relationships would interconnect universals and not just particulars (otherwise it should rather be called a fact base, knowledge graph, or the like).

– OFMs and use cases: even if OFMs, say, in OWL, tend to be tuned for being efficiently handled by particular kinds of applications, such as reasoners, query systems, graph visualizers, and the like, they are mostly not designed to be specific for a single use case.

The bold arrows between OBMs and OFMs correspond to the '$\mathcal{B}$-models' relationship.

OBMs should be represented in a suitable KR language, an *OBM language* (OBML), which would define the modeling primitives and possibly constraints. The presented PURO OBML is one such language. As mentioned above, a more powerful OBML (although not explicitly labeled as such) that has been previously mapped to OWL [2,38] is *OntoUML* [37]. Aspects of OBML can also be found in the *OntoClean* [36] metamodeling method, although no fully fledged OBMs are constructed in it; OntoClean primitives are rather assigned to the OFM entities in the form of 'metaproperties'. A more detailed comparison with these languages is included in Section 8.

3. The language of PURO

In this section we describe all elements of the PURO OBML, and we specify the coherence constraints associated with them. We express the coherence constraints on PURO background models as a theory in first-order logic. Finally, we illustrate the use of the language on the motivation examples from Section 1.

3.1. The meta-level language and sorts

Our formalization of the PURO OBML expresses properties of and constraints on the terms of the PURO language with respect to ontological entities and values described by these terms. We have chosen ordinary first-order logic as the meta language of this formalization for the sake of simplicity. The terms of the PURO OBML and the entities described by them will be all treated as objects of the domain of discourse. First-order predicates will express the properties of these objects and relationships among them. This view allows for more expressivity and flexibility than the alternative of seeing only the entities as objects and PURO terms as predicates.

Since the domain of discourse contains elements of diverse nature, we need to explicitly divide them among mutually disjoint sorts. We have settled on the following four sorts:

1. *Ontological entities* are the objects, relationships, types, relations, etc. existing in the modeled domain.
2. *Terms of the PURO OBML* (also *PURO terms*, for short), such as 'particular', 'universal', '$\mathcal{B}$-relationship', and others introduced below, refer to the ontological categories recognized by the PURO OBML. They can be *associated* with ontological entities in order to express their categorization. PURO terms are denoted by constants in the first-order formalization, which are intended to fully enumerate this sort.
3. *Quantitative values* (also *values*, for short) are elements such as the length '180 cm', the date 'December 6, 2017', or the time interval '20$^{\text{th}}$ century'. These are not considered ontological entities, but they are often used in ontologies to describe proper entities. We thus need to include them in the description of the PURO OBML, treating them as a distinct sort.
4. *Roles* are the parts that entities play in relationships, i.e., author, seller, supervisor, etc. For a given relationship, the elements of this sort serve to distinguish its participants. They also enable linking roles of entities in *n*-ary relationships in background models to the corresponding OWL object properties in foreground models.

Unary predicates Entity, PuroTerm, Value, and Role denote, respectively, the four sorts in the first-order formalization. The infix binary predicate : (colon) associates an entity with a term of the PURO OBML (i.e., Person : Universal means that the entity Person is categorized as a universal in the PURO model).

The disjointness of sorts and the constraint on the arguments of the predicate : are axiomatized as follows:

$$\begin{aligned}&\neg(\mathsf{Entity}(x) \wedge \mathsf{Value}(x)) \wedge \\ &\quad \neg(\mathsf{PuroTerm}(x) \wedge \mathsf{Entity}(x)) \wedge \\ &\quad \neg(\mathsf{PuroTerm}(x) \wedge \mathsf{Value}(x)) \wedge \\ &\quad \neg(\mathsf{Role}(x) \wedge \mathsf{Entity}(x)) \wedge \\ &\quad \neg(\mathsf{Role}(x) \wedge \mathsf{Value}(x)) \wedge \\ &\quad \neg(\mathsf{Role}(x) \wedge \mathsf{PuroTerm}(x))\end{aligned} \tag{1}$$

$$x : t \rightarrow \mathsf{Entity}(x) \wedge \mathsf{PuroTerm}(t) \tag{2}$$

Note that in the first-order notation we consider $\wedge$ and $\vee$ to be of a higher precedence than $\rightarrow$ and $\leftrightarrow$. We omit parentheses accordingly. We consider axioms to be closed formulas, and omit outermost universal quantifiers.

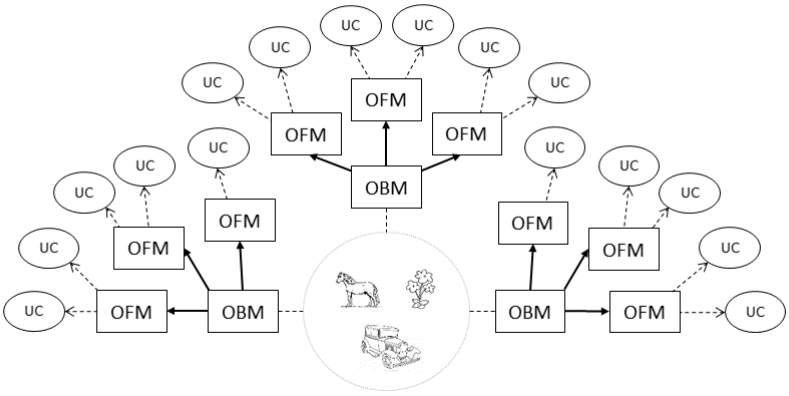

Fig. 3. Schema of relationships between reality, OBMs, OFMs and use cases

3.2. PURO distinctions and basic terms

The PURO OBML is built on the top of two basic ontological distinctions. The first one is the P–U distinction, between ontological *particulars* and *universals*, which are distinguished by the possibility of instantiation. Particulars are singular objects (e.g., representing a person or an object of the physical world). So, a distinguishing feature of particulars is that they cannot have instances. Universals, on the other hand, have the nature of categories (e.g., the category of all persons, or the one of all animals); so universals possibly may have instances. As a consequence of this distinction, particulars and universals are disjoint.

The P–U distinction is based on the relation of *instantiation*. In our first-order axiomatization of PURO, this relation is the intended interpretation of the binary predicate instanceOf. It associates two ontological entities, the first one being an instance of the second one; hence the axiom:

$$\mathsf{instanceOf}(x, y) \rightarrow \mathsf{Entity}(x) \wedge \mathsf{Entity}(y) \tag{3}$$

The essential consequences of the P–U distinction can be then expressed as follows:[7]

$$\begin{array}{l}\textsc{PuroTerm}(\mathsf{Particular})\\ \textsc{PuroTerm}(\mathsf{Universal})\end{array} \tag{4}$$

$$\textsc{Disjoint}(\mathsf{Particular}, \mathsf{Universal}) \tag{5}$$

$$\exists x\, \mathsf{instanceOf}(x, y) \rightarrow y : \mathsf{Universal} \tag{6}$$

The set of all instances of a universal is called its *extension*.

The R–O distinction, on the other hand, is the one between *objects* and *relationships*. Objects are singular entities with their own identity (e.g., each entity representing a person from the physical world is an object). A relationship cannot be considered (and talked about) without also considering the participating entities (e.g., the parent-of relationship between a mother and her child).

The R–O distinction is orthogonal to the P–U distinction, that is, it enables to distinguish between the particulars which are objects and those which are relationships, but as well between the universals which are categories of objects, which correspond to the foreground notion of classes and we call them *types* (e.g., again, the category of all persons), and those which are categories of relationships, which correspond to the foreground notion of properties and we call them *relations* (e.g., the universal entity respective to the parenthood relation).

Besides objects and relationships, RDF datasets prominently feature assignments of (quantitative) values to individuals. Ontologically, such assignments are not proper relationships, and thus we distinguish them as a third option within the R–O distinction which we call *valuations* and their universal counterparts are called *attributes*.

This orthogonality between the twofold P–U axis and the threefold R–O(–V) axis partitions all entities into one of the six kinds shown in Table 1. In the ta-

[7] $\textsc{Disjoint}(P_1, P_2, \ldots, P_k)$ abbreviates the formalization $\forall x \bigwedge_{1 \leq i \leq k, i < j \leq k} \neg(x : P_i \wedge x : P_j)$ of pairwise disjointness of PURO terms $P_1, \ldots, P_k$.

ble and throughout the paper, the PURO terms that refer to these kinds are preceded with $\mathcal{B}$- to indicate that they are terms of an OBM that presumably $\mathcal{B}$-models some OFM, whether existing or prospective. Each of these PURO terms then can be associated with one or more OFM entities in order to clarify their ontological background. Note that the 'typical corresponding foreground notions' in OWL are merely the structurally closest counterparts of the PURO terms, but not the only possible ones; they may even be inapplicable for various reasons, as we discussed in Section 1.

Table 1
Basic terms of the PURO OBML (and their typical corresponding foreground notions)

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

The consequences of the R–O distinction further add to our axiomatization:

$$\begin{aligned}&\textsc{PuroTerm}(\mathsf{BObject})\\&\textsc{PuroTerm}(\mathsf{BRelationship})\\&\textsc{PuroTerm}(\mathsf{BValuation})\end{aligned} \tag{7a}$$

$$x : \mathsf{Particular} \leftrightarrow x : \mathsf{BObject} \vee x : \mathsf{BRelationship} \vee x : \mathsf{BValuation} \tag{7b}$$

$$\textsc{Disjoint}(\mathsf{BObject}, \mathsf{BRelationship}, \mathsf{BValuation}) \tag{7c}$$

$$\begin{aligned}&\textsc{PuroTerm}(\mathsf{BType})\\&\textsc{PuroTerm}(\mathsf{BRelation})\\&\textsc{PuroTerm}(\mathsf{BAttribute})\end{aligned} \tag{8a}$$

$$x : \mathsf{Universal} \leftrightarrow x : \mathsf{BType} \vee x : \mathsf{BRelation} \vee x : \mathsf{BAttribute} \tag{8b}$$

$$\textsc{Disjoint}(\mathsf{BType}, \mathsf{BRelation}, \mathsf{BAttribute}). \tag{8c}$$

In the following subsections we describe the PURO terms in detail, including their illustration using the examples from Section 1.

3.3. Objects and types

We will now describe the P–U distinction applied to objects in more detail:

$\mathcal{B}$-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, etc.). It is analogous to the notion of *individual* in the foreground model.

$\mathcal{B}$-type refers to a universal whose instances are entities belonging to the same concept or sharing a common property. $\mathcal{B}$-types generalize the foreground notion of *classes*. 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.

The most common $\mathcal{B}$-types are those whose instances are $\mathcal{B}$-objects. Such $\mathcal{B}$-types correspond to what is most typically meant by classes in OFMs. We refer to such $\mathcal{B}$-types as to *level-1* $\mathcal{B}$-types.

In the OBM of Example 1, `ex:myLP0047` is[8] a $\mathcal{B}$-object, and `mo:MusicalItem` is a level-1 $\mathcal{B}$-type. Similarly, `dbr:Johann_Sebastian_Bach` and `dbr:Yo-Yo_Ma` are $\mathcal{B}$-objects, and their respective classes `mo:Composer` and `mo:MusicArtist` are level-1 $\mathcal{B}$-types in the background model. In the OBM of Example 2, the seller `dbr:EMI` is a $\mathcal{B}$-object, and `gr:BusinessEntity` is a level-1 $\mathcal{B}$-type.

Unlike OWL, or most description logics, the PURO OBML also admits *higher-level* $\mathcal{B}$-types.

For example, in GR, there is a level-2 $\mathcal{B}$-type `gr:BusinessEntityType`, whose instances, such as `gr:PublicInstitution`, are level-1 $\mathcal{B}$-types. One can find a more complex system of higher-level $\mathcal{B}$-types in MO, which is also illustrated in Example 1: 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; in particular, `ex:myLP0047` is an instance of `ex:CBS_D3_37867`. 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 level-1 $\mathcal{B}$-types with different foreground representations; similarly, `mo:Release` and `mo:album` are level-2 $\mathcal{B}$-types; and `mo:ReleaseType` is a level-3 $\mathcal{B}$-type.

$\mathcal{B}$-types are usually *homogeneous* in the sense that instances of a $\mathcal{B}$-type are entities of the same kind. For example, instances of `gr:PublicInstitution` are all $\mathcal{B}$-

[8] More precisely, we should say 'corresponds to' or 'is a manifestation of' rather than just 'is', since an OFM term is not identical to its corresponding an OBM term. In commenting the examples we omit this distinction, for brevity.

objects representing the actual organizations. Instances of gr:BusinessEntityType are all level-1 $\mathcal{B}$-types representing types of legal entities such as gr:PublicInstitution or gr:Reseller; one does not expect it to have $\mathcal{B}$-objects representing actual organizations as instances.

Homogeneity leads to stratification of $\mathcal{B}$-types: instances of *$\mathcal{B}$-types of level 1* are $\mathcal{B}$-objects, instances of *$\mathcal{B}$-types of level n* for $n > 1$ are $\mathcal{B}$-types of level $n-1$. However, in metamodelling there are also $\mathcal{B}$-types which are heterogeneous by nature, e.g., the type Deprecated, which one may use to classify all entities in an ontology which should not be used any more.

The following relationships among $\mathcal{B}$-objects and $\mathcal{B}$-types are axiomatized below for all integers $n \geq 1$ and $n' > n$: Every entity which is a BType_n for $n \geq 1$ is also a BType (9b), and no entity is both a BType_n and a $\mathsf{BType}_{n'}$ (9c). Each entity y which is a BType_n for $n \geq 1$ is homogeneous, i.e., all instances of y are either BObjects in the case of BType_1 (9d), or of the kind BType_{n-1} for $n > 1$ (9e). Similarly, if an instance of a universal y is a BType_n (resp. a BObject), then y is either a BType_{n+1} (resp. a BType_1) or it is heterogeneous (9f, 9g).

$$\textsc{PuroTerm}(\mathsf{BType}_n) \tag{9a}$$

$$x : \mathsf{BType}_n \rightarrow x : \mathsf{BType} \tag{9b}$$

$$\textsc{Disjoint}(\mathsf{BType}_n, \mathsf{BType}_{n'}) \tag{9c}$$

$$\mathsf{instanceOf}(x, y) \wedge y : \mathsf{BType}_1 \rightarrow x : \mathsf{BObject} \tag{9d}$$

$$\mathsf{instanceOf}(x, y) \wedge y : \mathsf{BType}_{n+1} \rightarrow x : \mathsf{BType}_n \tag{9e}$$

$$\mathsf{instanceOf}(x, y) \wedge x : \mathsf{BObject} \rightarrow y : \mathsf{BType}_1 \vee (y : \mathsf{BType} \wedge \mathsf{Heterogeneous}(y)) \tag{9f}$$

$$\mathsf{instanceOf}(x, y) \wedge x : \mathsf{BType}_n \rightarrow y : \mathsf{BType}_{n+1} \vee (y : \mathsf{BType} \wedge \mathsf{Heterogeneous}(y)) \tag{9g}$$

Any actual PURO model will only need finitely many levels of $\mathcal{B}$-types. Hence, there is always some bound m such that the axioms (9c, 9e, 9g) are only needed for $n, n' \leq m$. The axiomatization can thus be finite. This assumption is also supported by some previous works [64,44].

3.4. Relationships and relations

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., a musical work is composed by a composer, an object is produced by some producer; an object is of a certain type; one type of goods is a special case of another; a vendor exclusively supplies a customer with some kind of goods).

$\mathcal{B}$-relationship generalizes several foreground notions. The most basic of them is that of object property assertion. However, $\mathcal{B}$-relationships can also be n-ary for $n > 2$. PURO models also explicitly deal with a few finer aspects of relationships, which we introduce alongside their axiomatization below.

Each entity participating in a $\mathcal{B}$-relationship fulfills a particular *role*. For example, in a commercial transaction relationship, one entity participates as the transferred subject (an instance of goods or a service), another as the supplier, and yet another as the purchaser. The participant's role in a $\mathcal{B}$-relationship is explicit. This is similar to the reified foreground manifestation of relationships, in which the relationship is an individual connected to participants by multiple object property assertions corresponding to roles, but unlike the basic foreground manifestation of a binary relationship, in which participants are distinguished by being the subject or the object of a single object property assertion.

We thus axiomatize the properties of $\mathcal{B}$-relationships in FOL with the help of the sort of roles (see Sect. 3.1) and a ternary predicate hasParticipant. The intended meaning of $\mathsf{hasParticipant}(r, \rho, x)$ is that the relationship r has the entity x acting as a participant in the role ρ. For instance, a formal background model of the composition relation from Example 1 can include the fact hasParticipant(c, composer, Johann_Sebastian_Bach). We postulate that everything that has a participant is a $\mathcal{B}$-relationship, while the participant is an ontological entity (but not a value) and it has a role in the relationship:

$$\mathsf{hasParticipant}(r, \rho, x) \rightarrow r : \mathsf{BRelationship} \wedge \mathsf{Entity}(x) \wedge \mathsf{Role}(\rho) \tag{10a}$$

Similarly to the upper bound on the number of $\mathcal{B}$-type levels, we also assume there is a finite number

of relationship roles in any given background model, and, finitely many entities participate in every $\mathcal{B}$-relationship.

Note that some relationships have several participants with the same role. Examples include a shared ownership of a property or being siblings. In foreground models, one such relationship can manifest as several binary object property assertions (some of them possibly inferred from the symmetry of the object property). This should not occur in a background model, e.g., the siblings Ann, Ben, and Clara should all participate in a single relationship, all in the same role (i.e., sibling).

Each $\mathcal{B}$-relationship is at least binary, in the sense that it either has two different participants (possibly with the same role), or it has two different roles, each fulfilled by a participant (which is possibly the same entity):

$$\begin{aligned} &r : \mathsf{BRelationship} \\ &\quad \rightarrow \exists x_1\, \exists x_2\, \exists \rho_1\, \exists \rho_2 \\ &\qquad ((x_1 \neq x_2 \vee \rho_1 \neq \rho_2) \wedge \\ &\qquad\ \mathsf{hasParticipant}(r, \rho_1, x_1) \wedge \\ &\qquad\ \mathsf{hasParticipant}(r, \rho_2, x_2)) \end{aligned} \tag{10b}$$

Note that relationships with equal participants are *not* necessarily equal. This is in line with the intuition that, e.g., the relationships 'marriage of Liz and Rich' and 'business partnership of Liz and Rich' differ. They are instances of different relations, they differ in the roles for Liz and Rich, and they may also have different values of attributes, such as the time when the relationship was established (see below). In fact, Liz and Rich might also be in two distinct marital relationships, since they might marry, then divorce, only to remarry later. Relationships are thus a richer concept than n-tuples. We gave three examples of relationships of Liz and Rich but there is only one set-theoretical pair (Liz, Rich).

As we noted, $\mathcal{B}$-relationship generalizes also other foreground notions such as instantiation, and interclass axioms. We further discuss various kinds of $\mathcal{B}$-relationships below in Section 3.7.

$\mathcal{B}$-relation refers to a conceptual relation, the universal counterpart of $\mathcal{B}$-relationship. It is analogous to the (much more limited) foreground notion of *object property*. Therefore, instances of $\mathcal{B}$-relations are $\mathcal{B}$-relationships (11a), and conversely $\mathcal{B}$-relationships are typically instances of $\mathcal{B}$-relations, while they can also be classified under heterogeneous $\mathcal{B}$-types (11b).

Unless the relation is known to be heterogeneous, the set of all their participants with the same role is homogeneous (11c). It is, however, not required that all relationships in a $\mathcal{B}$-relation have the same number of participants with a given role or the same set of roles. A $\mathcal{B}$-relation of shared ownership can thus have instances with differing numbers of owners, and an offering-for-sale relation may have some instances in which an eligible region participates and other instances without such a participant.

$$\begin{aligned} &\mathsf{instanceOf}(x, y) \wedge y : \mathsf{BRelation} \\ &\quad \rightarrow x : \mathsf{BRelationship} \end{aligned} \tag{11a}$$

$$\begin{aligned} &\mathsf{instanceOf}(x, y) \wedge x : \mathsf{BRelationship} \\ &\quad \rightarrow y : \mathsf{BRelation} \vee \\ &\qquad (y : \mathsf{BType} \wedge \mathsf{Heterogeneous}(y)) \end{aligned} \tag{11b}$$

$$\begin{aligned} &\mathsf{instanceOf}(r_1, R) \\ &\qquad \wedge \mathsf{instanceOf}(r_2, R) \\ &\qquad \wedge \mathsf{hasParticipant}(r_1, \rho, x_1) \\ &\qquad \wedge \mathsf{hasParticipant}(r_2, \rho, x_2) \\ &\qquad \wedge x_1 : T \\ &\quad \rightarrow x_2 : T \vee \mathsf{Heterogeneous}(R) \end{aligned} \tag{11c}$$

3.5. Valuations and attributes

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). Unlike $\mathcal{B}$-relationships, $\mathcal{B}$-valuations are always binary and their second participants are quantitative data values (e.g., the duration of a recorded signal is 7806 sec), rather than entities. An inherent part of most data values is also the *unit*.

$\mathcal{B}$-valuation is analogous to the foreground notion of *data property assertion*, which is also the usual foreground manifestation of a $\mathcal{B}$-valuation. However, it is not always the case the other way around: The background entity corresponding to a data property assertion of a non-quantitative value is not a $\mathcal{B}$-valuation, as demonstrated in the next section.

We axiomatize $\mathcal{B}$-valuations with the help of predicates hasValuation and hasValue. We postulate that: (12a) every valuation is a valuation of some subject (that has this valuation), and (12b) assigns it a value; (12c) the subject is a PURO entity, whereas (12d) the value is a quantitative value; (12e, 12f) the subject of a particular valuation and the value assigned are both uniquely determined. As is the case with $\mathcal{B}$-

relationships, two $\mathcal{B}$-valuations with equal subjects and values are not necessarily equal.

$$v : \mathsf{BValuation} \rightarrow \exists x\, \mathsf{hasValuation}(x, v) \tag{12a}$$

$$v : \mathsf{BValuation} \rightarrow \exists w\, \mathsf{hasValue}(v, w) \tag{12b}$$

$$\mathsf{hasValuation}(x, v) \rightarrow \mathsf{Entity}(x) \wedge v : \mathsf{BValuation} \tag{12c}$$

$$\mathsf{hasValue}(v, w) \rightarrow \mathsf{Value}(w) \wedge v : \mathsf{BValuation} \tag{12d}$$

$$\mathsf{hasValuation}(x_1, v) \wedge \mathsf{hasValuation}(x_2, v) \rightarrow x_1 = x_2 \tag{12e}$$

$$\mathsf{hasValue}(v, w_1) \wedge \mathsf{hasValue}(v, w_2) \rightarrow w_1 = w_2 \tag{12f}$$

A simple example of foreground manifestations of a $\mathcal{B}$-valuation can be seen in the following birthdate assignment to a person:[9]

```
dbr:Johann_Sebastian_Bach
  foaf:name "Johann Sebastian Bach"^^xsd:string ;
  foaf:birthdate "1685-03-31"^^xsd:date .
```

Note that while both foaf:name and foaf:birthdate are used with literal values, only the latter corresponds to a valuation. On the other hand, the string 'Johann Sebastian Bach' is merely an alternative name of the given $\mathcal{B}$-object, besides the IRI dbr:Johann_Sebastian_Bach.

In this case the valuation is assigned to a $\mathcal{B}$-object, but more complex cases are possible. For instance, while values are not allowed as direct participants in relationships, valuations can be assigned to relationships. That is, if we want to express that Bach composed the Six Suites in a certain time interval – two $\mathcal{B}$-objects and one value being involved here – we are dealing with a binary relationship with an additional valuation. Also note that in OFMs valuations referring to a value with a unit may often be transformed (reified) to an OWL individual from which properties would lead both to the unit and to the data value ('magnitude'); this is what we saw in the assignment of the play time in Fig. 2. A similar reification effect would arise if a valuation participated in a relationship, or had itself another valuation, e.g., if a price (such that of an offered product) had some temporal validity period.

$\mathcal{B}$-attribute refers to a universal consisting of valuations of the same quantitative property. Thus $\mathcal{B}$-attributes are analogous to the foreground model notion of *data properties* (and, in fact, to the notion of attributes in ER schemas).

$\mathcal{B}$-attributes are homogeneous in the usual sense (13a, 13b). Similarly to $\mathcal{B}$-relations, the domain of a $\mathcal{B}$-attribute, i.e., the set of subjects of all its $\mathcal{B}$-valuations, is expected to be homogeneous unless known otherwise (13c). The range of a $\mathcal{B}$-attribute should also be a single type of values (we have excluded types of values from the formalization for simplicity).

$$\mathsf{instanceOf}(x, y) \wedge y : \mathsf{BAttribute} \rightarrow x : \mathsf{BValuation} \tag{13a}$$

$$\mathsf{instanceOf}(x, y) \wedge x : \mathsf{BValuation} \rightarrow y : \mathsf{BAttribute} \vee (y : \mathsf{BType} \wedge \mathsf{Heterogeneous}(y)) \tag{13b}$$

$$\mathsf{instanceOf}(v_1, A) \wedge \mathsf{instanceOf}(v_2, A) \wedge \mathsf{hasValuation}(x_1, v_1) \wedge x_1 : T \wedge \mathsf{hasValuation}(x_2, v_2) \rightarrow x_2 : T \vee \mathsf{Heterogenous}(A) \tag{13c}$$

3.6. Types of relations and attributes

Similarly to classification of types into higher-level types, it may be useful to classify relations and attributes. Examples of classifying roles[10] occur in the DL literature, e.g., both SNOMED and NCI Thesaurus enable roles to be organized into *role groups* [71,40, 69]. We therefore introduce and axiomatize below two $\mathcal{B}$-types for this purpose, $\mathsf{BType}_\mathsf{R}$ and $\mathsf{BType}_\mathsf{A}$ (14a, 14b), disjoint from each other and also from every BType_n for $n \geq 1$ (14c). Both are homogeneous and of level 2 (14d, 14e, 14f, 14g). We are not aware of any use cases that would justify types of relations and attributes of higher levels, therefore we leave them out of PURO terms.

$$\mathsf{PuroTerm}(\mathsf{BType}_\mathsf{R}) \quad \mathsf{PuroTerm}(\mathsf{BType}_\mathsf{A}) \tag{14a}$$

$$x : \mathsf{BType}_\mathsf{R} \vee x : \mathsf{BType}_\mathsf{A} \rightarrow x : \mathsf{BType} \tag{14b}$$

[9] The example uses the popular Friend-of-a-Friend (FOAF) vocabulary, http://xmlns.com/foaf/spec/.

[10] This term roughly corresponds to the union of 'relation' and 'attribute' in our PURO terminology and should not be confounded with that of PURO roles. We do not use the DL meaning of 'role' any further in this paper.

$$\textsc{Disjoint}(\mathsf{BType}_R, \mathsf{BType}_A, \mathsf{BType}_n) \tag{14c}$$

$$\mathsf{instanceOf}(x, y) \wedge y : \mathsf{BType}_R \rightarrow x : \mathsf{BRelation} \tag{14d}$$

$$\mathsf{instanceOf}(x, y) \wedge x : \mathsf{BRelation} \rightarrow y : \mathsf{BType}_R \vee (y : \mathsf{BType} \wedge \mathsf{Heterogeneous}(y)) \tag{14e}$$

$$\mathsf{instanceOf}(x, y) \wedge y : \mathsf{BType}_A \rightarrow x : \mathsf{BAttribute} \tag{14f}$$

$$\mathsf{instanceOf}(x, y) \wedge x : \mathsf{BAttribute} \rightarrow y : \mathsf{BType}_A \vee (y : \mathsf{BType} \wedge \mathsf{Heterogeneous}(y)) \tag{14g}$$

3.7. 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 three kinds of $\mathcal{B}$-relationships: $\mathcal{B}$-instantiations, $\mathcal{B}$-axioms, and $\mathcal{B}$-facts. These three kinds are mutually disjoint and every relationship in the background model belongs to one of them. These basic facts are axiomatized as follows:

$$\begin{array}{l}\textsc{PuroTerm}(\mathsf{BInstantiation})\\ \textsc{PuroTerm}(\mathsf{BAxiom})\\ \textsc{PuroTerm}(\mathsf{BFact})\end{array} \tag{15a}$$

$$r : \mathsf{BRelationship} \leftrightarrow r : \mathsf{BInstantiation} \vee r : \mathsf{BAxiom} \vee r : \mathsf{BFact} \tag{15b}$$

$$\textsc{Disjoint}(\mathsf{BInstantiation}, \mathsf{BAxiom}, \mathsf{BFact}) \tag{15c}$$

Let us now discuss the kinds of relationships in detail:

$\mathcal{B}$-instantiation is a relationship between an entity and a universal that the entity *instantiates*. A $\mathcal{B}$-instantiation intuitively means that the entity belongs to extension of the given universal, and it is a background analogue of an `rdf:type` statement. Unlike in foreground models, any ontological entity can play the role of the instance, as is apparent from the examples and the formalization below. Foreground manifestations of $\mathcal{B}$-instantiation identified so far in LD vocabularies are:

- *`rdf:type` statement.* This is a canonical manifestation of $\mathcal{B}$-instantiation. In the simplest case, a `rdf:type` triple such as

```
dbr:Yo-Yo_Ma rdf:type mo:MusicArtist .
```

  from Example 1 asserts that a $\mathcal{B}$-object foreground-modeled as an individual is a $\mathcal{B}$-instance of a 1st-level $\mathcal{B}$-type foreground-modeled as a class. As a more interesting example, take the individual `mo:album` which instantiates the class `mo:ReleaseType` in MO, as asserted by:

```
mo:album rdf:type mo:ReleaseType .
```

  As explained in Sect. 3.3, `mo:album` is perceived as a $\mathcal{B}$-type. The above statement is nonetheless viewed as $\mathcal{B}$-instantiation, one between a 2nd-level $\mathcal{B}$-type and a 3rd-level one.
- *Object property assertion.* If a $\mathcal{B}$-type is represented as a foreground individual, $\mathcal{B}$-instantiation is expressed as an object property of the type's instances. Recall the individual `ex:CBS_D3_37867` from Example 1. In MO, we express that the 1st-level $\mathcal{B}$-type represented by this individual instantiates the 2nd-level $\mathcal{B}$-type `mo:album` (also represented as a foreground individual) as:

```
ex:CBS_D3_37867 mo:release_type mo:album .
```

  Note that commonly the object property (`mo:release_type`), and sometimes also the class in its range, exhibits a suffix such as -type or -class in LD vocabularies.
- *Data property assertion.* If a $\mathcal{B}$-type is represented as a foreground data value (string), $\mathcal{B}$-instantiation is expressed as a data property, as in this GR example:

```
dbr:EMI gr:category
  "MusicSeller/VinylSeller"^^xsd:string .
```

  In GR, the individual `dbr:EMI` is primarily classified in the class `gr:BusinessEntity`. Its classification in the $\mathcal{B}$-type identified by the string above can be seen as of a different kind, it is nonetheless an $\mathcal{B}$-instantiation. The designer of GR might have wanted to give more flexibility to the user with such textual categories.
- As regards the $\mathcal{B}$-instantiation between a relationship and a relation (or, a valuation and an attribute), they are not explicitly manifested in OWL, since a property assertion simply refers to the IRI of the property without a dedicated construct.[11] However, in a situation when a PURO re-

[11] Actually, the same short notation is also used in the PURO graphical language used by the tools overviewed in Section 5.

lationship, e.g., one with a higher arity, is reified to an OWL individual, it may actually be manifested through an rdf:type statement.

```
ex:c rdf:type mo:Composition .
```

$\mathcal{B}$-instantiation is formalized below with the help of two mutually distinct roles – the instance role and the universal role (16a). A $\mathcal{B}$-instantiation is then a binary $\mathcal{B}$-relationship whose participant playing the role of the instance is actually an instance of the participant playing the role of the universal and there are no other participants (16b).

$$\begin{aligned}&\mathsf{Role}(\mathsf{instance})\\&\mathsf{Role}(\mathsf{univ})\\&\mathsf{instance} \neq \mathsf{univ}\end{aligned} \tag{16a}$$

$$\begin{aligned}&r:\ \mathsf{BInstantiation}\\&\quad\rightarrow \exists x\, \exists U\\&\qquad\big(\mathsf{hasParticipant}(r, \mathsf{instance}, x) \wedge\\&\qquad\ \mathsf{hasParticipant}(r, \mathsf{univ}, U) \wedge\\&\qquad\ \mathsf{instanceOf}(x, U) \wedge\\&\qquad\ \forall y\, \forall \rho(\, \mathsf{hasParticipant}(r, \rho, y)\\&\qquad\qquad\rightarrow \rho = \mathsf{instance} \wedge y = x \vee\\&\qquad\qquad\ \rho = \mathsf{univ} \wedge y = U)\big)\end{aligned} \tag{16b}$$

The various object/type properties in the examples above (rdf:type, mo:release_type, gr:category) are all manifestations of type roles.

$\mathcal{B}$-axiom is a $\mathcal{B}$-relationship of universals, expressing a set-theoretic relationship between their extensions (such as subsumption or disjointness). Foreground manifestations of a $\mathcal{B}$-axiom identified so far are:

- *Corresponding axiom.* This is a canonical manifestation of a $\mathcal{B}$-axiom, e.g., class subsumption expressed by rdfs:subclassOf between two classes, each corresponding to a $\mathcal{B}$-type.
- *Object property assertion.* In this case both $\mathcal{B}$-types are 'metamodeled' as individuals in the foreground representation. Their mutual relationship can be expressed using properties such as skos:narrower and skos:broader from the SKOS vocabulary.[12] A typical example is the MO-recommended SKOS version of MusicBrainz Instrument Taxonomy. A fragment of the taxonomy is shown in the left part of Fig. 1. Its individuals, terms such as mo-mit:Cello, refer to $\mathcal{B}$-types—whole classes of physical instruments. Thus foreground assertions such as:

```
mo-mit:Cello
    skos:broader mo-mit:Violins ;
    skos:narrower mo-mit:ElectricCello .
```

reflect subsumption $\mathcal{B}$-axioms.

We axiomatize $\mathcal{B}$-axioms below, starting with the fact that they relate universals (17a). There are possibly multiple mutually disjoint kinds of $\mathcal{B}$-axioms. However, we will only axiomatize those mentioned above: $\mathcal{B}$-subsumption and $\mathcal{B}$-disjointness (17b, 17c, 17d), which are more likely to occur.

$$\begin{aligned}&r:\ \mathsf{BAxiom} \wedge \mathsf{hasParticipant}(r, \rho, U)\\&\quad\rightarrow U:\ \mathsf{Universal}\end{aligned} \tag{17a}$$

$$\begin{aligned}&\textsc{PuroTerm}(\mathsf{BSubsumption})\\&\textsc{PuroTerm}(\mathsf{BDisjointness})\end{aligned} \tag{17b}$$

$$\begin{aligned}&\textsc{Disjoint}(\mathsf{BSubsumption},\\&\qquad\qquad\mathsf{BDisjointness})\end{aligned} \tag{17c}$$

$$\begin{aligned}&r:\ \mathsf{BSubsumption} \vee r:\ \mathsf{BDisjointness}\\&\quad\rightarrow r:\ \mathsf{BAxiom}\end{aligned} \tag{17d}$$

$\mathcal{B}$-subsumption is a binary $\mathcal{B}$-axiom. One of its participants plays the role of a sub-universal, the other plays the role of a super-universal. These roles are mutually distinct (18a). The extension of the sub-universal is a subset of extension of the super-universal and there are no other participants (18b).

$$\begin{aligned}&\mathsf{Role}(\mathsf{subuniv})\\&\mathsf{Role}(\mathsf{superuniv})\\&\mathsf{subuniv} \neq \mathsf{superuniv}\end{aligned} \tag{18a}$$

$$\begin{aligned}&r:\ \mathsf{BSubsumption}\\&\rightarrow \exists U\, \exists V\\&\quad\big(\mathsf{hasParticipant}(r, \mathsf{subuniv}, U) \wedge\\&\qquad\mathsf{hasParticipant}(r, \mathsf{superuniv}, V) \wedge\\&\qquad\forall x(\mathsf{instanceOf}(x, U)\\&\qquad\qquad\rightarrow \mathsf{instanceOf}(x, V)) \wedge\\&\qquad\forall w\, \forall \rho(\, \mathsf{hasParticipant}(r, \rho, w)\\&\qquad\qquad\rightarrow \rho = \mathsf{subuniv} \wedge w = U \vee\\&\qquad\qquad\ \rho = \mathsf{superuniv} \wedge w = V)\big)\end{aligned} \tag{18b}$$

$\mathcal{B}$-disjointness is a $\mathcal{B}$-axiom (n-ary for $n \geq 2$ by default) that relates universals with pairwise mutually

[12] Note that skos:narrower and skos:broader properties do not *necessarily* correspond to class subsumption. Their meaning, as defined in the SKOS vocabulary, is slightly more general.

disjoint extensions (19c). All these universals have the same role (19a, 19b).

$$\text{Role(disjunct)} \tag{19a}$$

$$r : \text{BDisjointness} \wedge \text{hasParticipant}(r, \rho, U) \rightarrow \rho = \text{disjunct} \tag{19b}$$

$$r : \text{BDisjointness} \wedge \text{hasParticipant}(r, \rho, U) \wedge \text{hasParticipant}(r, \rho, V) \wedge U \neq V \wedge \text{instanceOf}(x, U) \rightarrow \neg\text{instanceOf}(x, V) \tag{19c}$$

$\mathcal{B}$-fact is a $\mathcal{B}$-relationship that cannot be classified as either a $\mathcal{B}$-instantiation or a $\mathcal{B}$-axiom. Participants in a $\mathcal{B}$-fact are typically (but not necessarily) $\mathcal{B}$-objects. Foreground manifestations of $\mathcal{B}$-facts identified so far are:

- *Object property assertion.* The canonical foreground manifestation of a binary $\mathcal{B}$-fact is a property assertion connecting a pair of individuals. In the most common case, the individuals are manifestations of $\mathcal{B}$-objects. But if we express that Yo-Yo Ma's primary instrument is the cello using the MO vocabulary (where $\mathcal{B}$-types of instruments manifest as individuals, as already explained), we end up with a $\mathcal{B}$-fact in which a $\mathcal{B}$-object and a $\mathcal{B}$-type participate:

  ```
  dbr:Yo-Yo_Ma
     mo:primary_instrument mo-mit:Cello .
  ```

- *Data property assertion.* In this case, one of the $\mathcal{B}$-objects is reduced to a data value (a string) in the foreground representation. In Example 2, the offering is restricted to a region of eligibility, which is a geographical region, hence a $\mathcal{B}$-object. However, the GR vocabulary encodes regions as string codes, as apparent from the example:

  ```
  ex:o rdf:type gr:Offering ;
       gr:eligibleRegions "US-CA"^^xsd:string .
  ```

  Hence the second triple, a foreground data property assertion, corresponds to a $\mathcal{B}$-fact.
- *Reified relationship pattern.* $\mathcal{B}$-facts can appear *reified* in the foreground model. The foreground manifestation of a single reified $\mathcal{B}$-fact is a complex structure. It consists of a foreground individual, and of (object or data) property assertions that connect that individual to the manifestations of participants in the $\mathcal{B}$-fact. Reification is the only way to encode n-ary facts in LD, given that OWL does not support native n-ary constructs [61].
  In Example 1, the relationship between a musical work and its composer, with a further specification of the time interval and place when and where the composition took place, is expressed by the following structure:

  ```
  ex:c
    rdf:type mo:Composition ;
    mo:composer dbr:Johann_Sebastian_Bach ;
    mo:produced_work dbr:Cello_Suites_(Bach) ;
    event:time ex:int1717–1723 ;
    event:place dbr:Köthen_(Anhalt) .
  ```

  We see that the $\mathcal{B}$-objects `dbr:Johann_Sebastian_Bach`, `dbr:Cello_Suites_(Bach)` take part in a $\mathcal{B}$-relationship of type `mo:Composition`. It should be however noted that the boundary between (complex) relationships and objects may not be sharp in some situations. The reifying individual `ex:c` is of type `mo:Composition`, and hence according to the description in the MO vocabulary it is a 'composition event'. Hence, in this kind of modeling, the respective $\mathcal{B}$-relationship is confounded with a true $\mathcal{B}$-object (the composition event), which established the relationship. Similarly instances of the `gr:Offering` class in the GR vocabulary can be viewed as reifying a complex relationship, but at the same time as abstract information objects – $\mathcal{B}$-objects.
  The distinction between whether a particular entity should be perceived as an object or as a relationship (or both) 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 without explicitly considering the other participants in the relationship. (In these terms, information about a 'composition event' is clearly incomplete without knowing what was composed.)
- *Counting pattern.* A particular kind of $\mathcal{B}$-fact manifestations is one (similar to the 'shortcuts' explained next in Sec. 3.9) allowing to merely record the existence of the fact using a counter, without specifying the object on its other side. Consider a PURO model of a computer configuration, comprising a $\mathcal{B}$-fact connecting a computer processor with its core. While for a computer configuration ontology we would presumably transform such a fact to an object property, the same

PURO model might also produce a computer sales ontology with a data property such as ex:numberOfCores. Note that this is a completely different kind of 'manifestation by data property' than in the second bullet above.

Since $\mathcal{B}$-facts have no special semantics, we give no further axiomatization beyond the one already presented at the beginning of this section (15a–15c).

3.8. Originating participants in relationships

The subject-object distinction of participants in object property assertions, and the related perception of the relationship as *originating* in the subject and directed towards the object are sometimes significant. For instance, in an ontology concerned primarily with products, the production relationship is oriented from the product towards the producer, and called produced_by rather than produces. This distinction can be achieved in the PURO role-based model of (n-ary) relationships by designating the role ρ of the subject as the *originating role* in a relationship r. This designation is formally written as $\mathsf{hasOrigRole}(r, \rho)$. Each participant in the relationship with this role is then an *originating participant*. There may be several originating participants in a single relationship with one or several originating roles. Thus, if 'owner' is an originating role in a shared ownership relationship, then all co-owners are originating participants, and the shared property is a target participant. In a parenthood relationship, the roles 'mother' and 'father' and their respective participants can be seen as originating. However, relationships are not required to have originating roles.

The designation of a role as originating can influence its visualization (as an arrow directed from the participant), the translation of the relationship to a foreground model or its verbalization (the originating participant becomes the subject of an object property assertion or of the sentence describing the relationship).

We place only two weak constraints on originating roles: There must be a participant fulfilling each role designated as originating in a relationship (20a). In a homogeneous relation, whenever two relationships have participants fulfilling a given role, this role must be either originating in both or neither of them (20b).

$$\mathsf{hasOrigRole}(r, \rho) \wedge r : \mathsf{BRelationship} \rightarrow \exists x\ \mathsf{hasParticipant}(r, \rho, x) \tag{20a}$$

$$\begin{aligned} &\mathsf{instanceOf}(r_1, R) \\ &\quad \wedge \mathsf{instanceOf}(r_2, R) \\ &\quad \wedge \mathsf{hasParticipant}(r_1, \rho, x_1) \\ &\quad \wedge \mathsf{hasParticipant}(r_2, \rho, x_2) \\ &\quad \wedge \mathsf{hasOrigRole}(r_1, \rho) \\ &\rightarrow \mathsf{hasOrigRole}(r_2, \rho) \vee \mathsf{Heterogenous}(R) \end{aligned} \tag{20b}$$

3.9. OWL shortcuts of PURO structures

In the enumerations of likely OWL manifestations of PURO structures, we so far only considered those corresponding to a single PURO entity, sometimes reifying it to more complex patterns. However, the opposite approach is also possible: parsimonious OWL vocabularies might shrink chunks of PURO models to 'shortcut' structures. Shortcutting is known as a common pattern used inside OWL ontologies, especially in the context of efficiently publishing large amounts of linked data [50]. This is particularly useful if the 'intermediate' entities are likely to remain anonymous and would only be 'skolem constants' in an OWL knowledge base. For example, the MO allows to build a sequence such as

```
ex:p rdf:type mo:Performance ;
     mo:produced_sound ex:s .
ex:s rdf:type mo:Sound ;
     mo:recorded_in ex:r .
ex:r rdf:type mo:Recording ;
     mo:produced_signal ex:sg .
```

but there is also a shortcut predicate for connecting the start of the chain with its end:

```
ex:p mo:recorded_as ex:sg .
```

The complex part of this dual structure can however be expressed using a PURO model; a single OWL property would then be the manifestation of multiple PURO entities. This would help keep the OWL ontology and RDF data small, but possibly even model (at PURO level) the background structures that would be difficult to express in OWL. Note that an automatic application of a PURO2OWL transformation (with some linguistic heuristics) might create an OWL property called something like *producedSoundInRecordingThatProducedSignal*, which an ontologist could then refine to a short form such as *recordedAs*.

3.10. Multi-object placeholders in PURO

Aside the basic PURO entities that are 'background counterparts' to OWL entities, we might also need enti-

ties that slightly deviate from the intuitive semantics of universals vs. particulars, and the like. As a representative of such special-status entities,[13] we will discuss what can be denoted as 'multi-object placeholders'.

A common requirement in data modeling is to refer to the existence of entities whose *identity*, but possibly also *number*, remain unspecified; we only know their common *type*. This modeling problem most typically appears when the original relationship refers to something possibly happening in the future: for example, a company *offers to sell* an unspecified number of physical products of some (catalog) type, or some activity *is scheduled to be carried out* by multiple agents of a certain type (i.e., whose identity and number is unknown when making the statement). A variation of the setting is obtained by replacing the common type with a common (individual) entity to which the unspecified entities are *related*. E.g., the company would be offering an unspecified number of products *produced by* a certain manufacturer, or the activity would be scheduled to be carried out by agents *employed by* a certain department.

In PURO we represent such a multi-object by a specific kind of $\mathcal{B}$-object, which does not have any particular features in the PURO formalization, but has a specific name, *Some objects*, and is assumed to be handled differently in terms of the OFM generation. The problem of expressing this kind of conceptualization in OWL has been tackled in our previous study [81], where a family of three patterns (called MISO, for 'Multiple Indirectly Specified Objects') has been proposed:[14]

1. *Existential restriction* the filler of which is either a class (for multi-objects specified by their $\mathcal{B}$-type) or a value restriction (for multi-objects specified by their $\mathcal{B}$-relationship to a $\mathcal{B}$-object); possibly the filler can also be a conjunction of multiple expressions (classes and/or value restrictions).
2. *Placeholder individual*, declared as instance of a class (for multi-objects specified by their $\mathcal{B}$-type) or linked to an individual via an object property (for multi-objects specified by their $\mathcal{B}$-relationship to a $\mathcal{B}$-object); again, multiple instantiations and/or property assertions can be specified for the placeholder individual. To distinguish the placeholder individual from 'normal' individuals, is should further be declared as instance of a dedicated 'utility' class.
3. 'Folding' the multi-object structure into a *custom property* directly shortcutting to either the entity representing the specifying $\mathcal{B}$-type (which now has to be metamodeled by an individual in the OFM) or the individual representing the specifying $\mathcal{B}$-object. This is a particular case of a shortcut structure discussed in Section 3.9. This pattern however only allows to model one single aspect (type, or relationship to an object) of the multi-object.

In Example 2, the assertion of the gr:includes property that connects the offering ex:o and the release ex:CBS_D3_37867 essentially corresponds to the use of the MISO pattern of the second type. The unspecified instances of the release are the individual physical records that are for sale. They are represented using a placeholder, which is declared as instance of gr:SomeItems (the utility class of multi-object placeholders).

The GR ontology also provides another example, which demonstrates the third type of MISO pattern. An offering of a product or service can be restricted to (undistinguished) possible customers from a certain category, namely to public institutions, as follows:

```
ex:o231 rdf:type gr:Offering ;
   gr:eligibleCustomerTypes gr:PublicInstitution .
gr:PublicInstitution
   rdf:type gr:BusinessEntityType .
```

The foreground individual gr:PublicInstitution reflects the target $\mathcal{B}$-type, hence the gr:eligibleCustomerTypes property is a foreground representation of a MISO pattern with the background meaning that the instances of gr:PublicInstitution (and not the metamodeled type itself) are the eligible customers.

3.11. PURO model

After formally defining the PURO OBML, it remains to define what we see as a concrete PURO model.

Definition 3 (PURO model) *A PURO model is a pair* $PM = (Met, Ext)$*, where Met is the model's metadata set, and Ext is the model's extension, which is a finite set of* $\mathcal{B}$*-relationships and* $\mathcal{B}$*-valuations.*

[13] Another one, which we however have not yet thoroughly addressed in our research, would be *prototypes* (or, possibly, defaults).

[14] All three patterns additionally feature components – OWL2 annotations – allowing to approximately quantify the number of objects represented by the multi-object. We leave this part out of our overview.

We do not elaborate on the notion of metadata set any further, it is however obvious that a model is more than its extension; the metadata may, for example, contain the name, provenance or versioning information on the model, analogously to the metadata part of OWL ontologies. As regards the extension, note that only these two types of PURO entities ($\mathcal{B}$-relationships and $\mathcal{B}$-valuations) can be interpreted as statements, similarly to axioms in OWL; other entities are merely part of the model signature:

Definition 4 (PURO model signature) *A signature of a PURO model $PM = (Met, Ext)$ is the union of the sets of all*

- *$\mathcal{B}$-objects and $\mathcal{B}$-types that are participants in some $\mathcal{B}$-relationship $r \in Ext$*
- *$\mathcal{B}$-objects and $\mathcal{B}$-types that have a $\mathcal{B}$-valuation $v \in Ext$*
- *$\mathcal{B}$-relations R such that there is a $\mathcal{B}$-relationship $r \in Ext$ that is an instance of R*
- *$\mathcal{B}$-attributes V such that there is a $\mathcal{B}$-valuation $v \in Ext$ that is an instance of V.*

In terms of the first-order interpretation from this paper, the notion of PURO model is further refined to a set of formulas over the following first-order predicates (PURO statement predicates):

- : (association of an entity to its PURO term)
- instanceOf
- subTypeOf
- hasParticipant
- hasOrigRole
- hasValuation
- hasValue

From the practical point of view, it should be recalled that explicitly represented PURO models, as considered in many of the use cases in Section 5 of this paper, should indeed contain not only $\mathcal{B}$-axioms (roughly corresponding to TBox axioms in OWL, but being much more lightweight) but also $\mathcal{B}$-instantiations, $\mathcal{B}$-facts and $\mathcal{B}$-valuations (roughly corresponding to ABox axioms in OWL). However, since PURO models are meant to be manageable by humans, their 'ABox' should not be extensive: it should only serve for interconnecting the 'TBox' entities into a single graph and exemplifying their usage, in a similar way as sample RDF graphs are frequently included into OWL ontology specifications (but with the advantages of formal modeling).

We also assume, though not as a definitional condition, that PURO models should have a contiguous structure (even if possibly through the 'ABox' only) rather than being sets of completely disparate sets of statements.

Definition 3 also does not postulate the existence of any OFM, i.e., a PURO model is not necessarily an OBM in the sense of Def. 1, although the main purpose of the PURO language is to build OBMs.

3.12. Motivation examples revisited

Figure 4 graphically depicts a possible PURO OBM of the motivation examples from Section 1 (covering the whole structure of the second example and a part of the first one, in one diagram). The notation is that used in our PURO Modeler authoring tool [21], in which the diagram was created: $\mathcal{B}$-objects are denoted by rectangles, $\mathcal{B}$-types of all levels by ovals, $\mathcal{B}$-relationships by diamonds, $\mathcal{B}$-valuations by hexagons, and their values by parallelograms. Relations and attributes are represented via their names inside the relationship/valuation diamond/hexagon; role names appear next to the connecting edges. Instantiations and axioms are drawn by solid lines while facts and valuations by dashed lines. An originating role has the arrow suppressed, in order to read the relationship more easily as a 'branching arrow' starting in the originating participant.

We see that the offering now becomes a ternary relationship, and that *CBS_D3_37867*, *Album* and *Release type* together form a chain of (meta-)types. The fact that *CBS_D3_37867* is a *Release* is not stated explicitly, but is hinted (assuming very simple lexical pattern matching) by the fact that its type, *Album*, is a *Release type*. Such an inference could be supported, e.g., via the use of the Multi-Level Theory (MLT), which we refer to in Section 8.

CBS_D3_37867 is instantiated by both a simple instance *myLP0047* and a set of multiple (MISO) instances 'indirectly specified' by it (labeled as *Some objects*).

The play time is now a simple valuation de-reified from the `ex:pt` individual; its typing by `gr:QuantitativeValueFloat` was a technicality of the GR vocabulary, not needed at the PURO level. Ontologically slightly arguable is the assignment of this valuation to the $\mathcal{B}$-type rather than to instances. We propose to directly connect $\mathcal{B}$-types both with valuations or facts that relate to these types *themselves* (to their intension), such as the fact about $\mathcal{B}$-type *Cello* being the primary instrument of a certain person (Section 3.7)

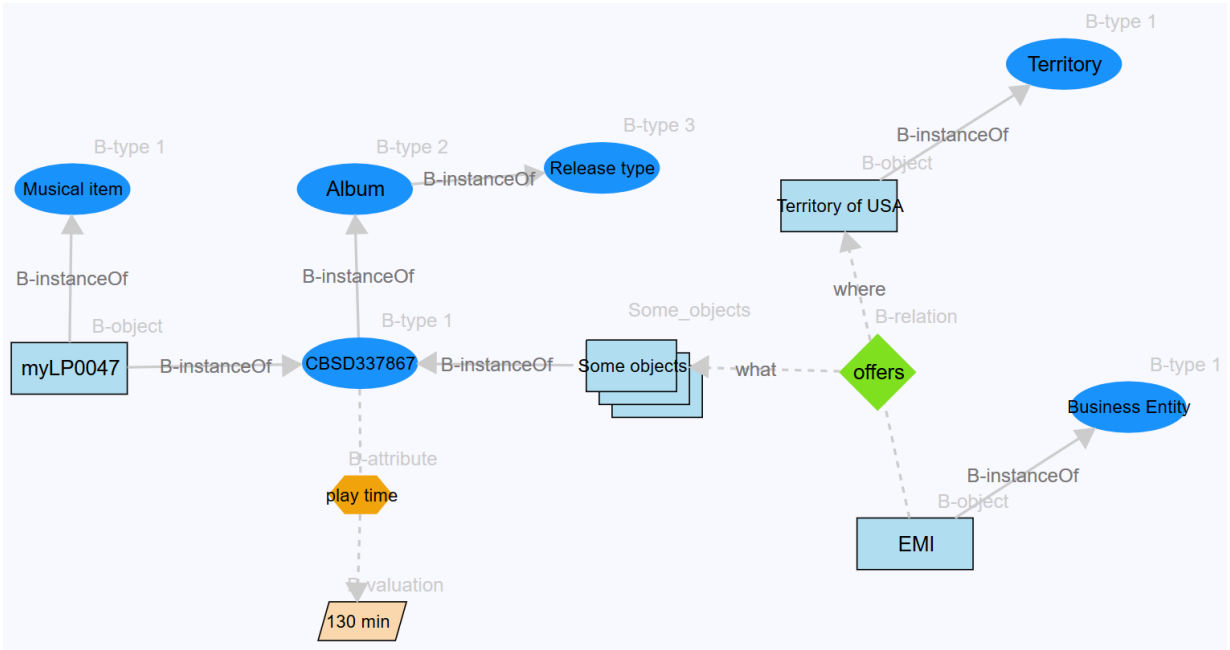

Fig. 4. PURO model diagram describing an offer of some LPs for sale and a specific LP exemplar

and with valuations or facts that relate to *all instances* of these types (their extension). Consequently, only *incidental* valuations of (and, analogously, facts including) particular instances should be directly connected with these instances. The finer-grained distinction of a PURO 'linking' to the intension/extension of a $\mathcal{B}$-type should however be part of an extra, optional layer of the language in the future.

Note also that the EAN code, as an alternative identifier of *CBS_D3_37867*, is suppressed in the PURO model (it could possibly be encapsulated inside the *CBS_D3_37867* node).

Considering the differences between this model and the OWL structures from Fig. 1 and Fig. 2, and the variety of alternative OFM manifestations of OBM structures, enumerated earlier in this section, we argue that such a model:

1. Captures the underlying reality more faithfully than the particular OWL models from Fig. 1 and 2, thus satisfying the *first* condition of being an OBM in Def. 1
2. May give rise to multiple pragmatic variants of OWL models.

In Section 4 we will further discuss how the alignment between PURO models and OWL models, as the *second*[15] condition of being an OBM in Def. 1, can be implemented.

4. Gluing PURO with OWL: transformations and alignments

PURO has been, from the beginning, primarily designed as a language complementary to OWL (even if nothing precludes its core from being used as a KR language on its own). In the previous section we treated the relationship between PURO and OWL models in terms of OWL entity types that typically appear as foreground manifestations of particular (atomic) PURO constructs. Now we will discuss the different ways the alignment between models in the two languages can be achieved, so that the PURO model would (at least, in some cases) satisfy the second condition of being an OBM, from Definition 1. Such an alignment can arise either for a pair of pre-existing models, via an *alignment process*, or when one model arises by transformation from the other, via a *transformation process*.

[15] Remember that the *third* condition of being an OBM in Def. 1 merely requires that the OBM is expressed using a different language than the OFM (mainly to technically distinguish OBMs from upper-level ontologies), which is trivially satisfied for PURO models being in the OBM role wrt. OWL ontologies.

In the following we will first formally define what we mean by a PURO2OWL alignment, and then start to investigate the different processes that may yield such an alignment either as their main outcome or as their by-product.

4.1. PURO2OWL alignments and their types

Formally, we can define the notion of alignment, starting from its constituent atomic alignment links (also called 'correspondences' in ontology matching literature [22]), as follows:

Definition 5 (PURO2OWL link) *A PURO2OWL link is a triple* (p, o, ls) *such that*

- *p, the (PURO)* source*, is either*
  1. *a (possibly empty) set of PURO entities*
  2. *a (possibly empty) set of PURO terms*
  3. *a non-empty set of PURO models*
- *o, the (OWL)* target*, is either*
  1. *a (possibly empty) set of OWL entities*
  2. *a non-empty set of OWL ontologies*
- *ls is a link structure, which defines the way p and o are interconnected.*

The definition is a rather abstract one, overarching multiple different types of links observed in the use cases described later in this paper, with highly variable granularity. Link participants can be entities (i.e., elements of PURO or OWL models), terms (i.e., elements of the PURO or OWL metamodel[16]), or whole models (PURO models or OWL ontologies).

Also note that the set of PURO entities or even terms in the source may be empty, since OWL entities such as identifiers may not have their counterpart in PURO. This also holds on the other side, i.e. the set of OWL entities in the target may be empty. The latter would however probably mean that the OWL encoding is burdened by some information loss wrt. the given PURO model. Such an information loss should not be necessary at the theoretical level,[17] but may have some pragmatic reason; for example, if the design of the OWL representation aims to avoid reification, secondary roles in relationships or their valuations may be eliminated.

Also note that we denote the two sides of a link as 'source' and 'target' for easier reference only. This does not imply that the links should be designed or traversed in the direction from the source to the target.

Definition 6 (PURO2OWL alignment) *A PURO2OWL alignment is a non-empty set of PURO2OWL links,* $A = \{(p_1, o_1, ls_1), ..., (p_n, o_n, ls_n)\}$*, such that all* p_i *are of the same kind (i.e., they are all sets of PURO entities, or all sets of PURO terms, or all sets of PURO models), and, similarly, all* o_i *are of the same kind (i.e., they are all sets of OWL entities, or all sets of OWL ontologies).*

The definitions thus allow for different kinds of alignments (which we will now describe in detail), but do not allow to mix entities, terms or models/ontologies on the same side of an alignment, for simplicity. Yet, if the alignment refers to a set of PURO/OWL entities on any side, they need not all belong to (the signature of) the same PURO model or OWL ontology, respectively.

Since by Definition 5 there are three possible types of PURO2OWL link participants on the PURO side and two on the OWL side, there are six types of links in terms of their participant types, and, consequently, six types of PURO2OWL alignments, between:

1. PURO entities and OWL entities (E2E)
2. PURO entities and OWL ontology (E2O)
3. PURO term and entities (T2E)
4. PURO term and ontology (T2O)
5. PURO model and entities (M2E)
6. PURO model and ontology (M2O).

Following our running example, let *MusicalAlbumSale* denote the whole PURO model from Fig. 4 and let `gr:` and `mo:` denote the whole of GoodRelations and Music Ontology, respectively. Let us also simplify, for this display only, the link notation from (p, o, ls) to $p \leftrightarrow o$ (setting aside the link structure ls). Links of the mentioned types then could be:

- E2E: { offers } ↔ { `gr:offers`, `gr:Offering`, `gr:includes`, `gr:eligibleRegions` }
- E2O: { offers } ↔ { `gr:` }
- T2E: {$\mathcal{B}$-relation } ↔ { `gr:Offering` }
- T2O: {$\mathcal{B}$-relation } ↔ { `gr:`, `mo:` }

[16] Note that since the metamodel of OWL is itself defined in OWL, it is also possible to put an OWL metamodel entity ('OWL term', analogous to a PURO term) to the right-hand side of the link, while technically it would be no different from linking to any other OWL entity.

[17] At least, we conjecture that any PURO entity in any model can be completely transformed to OWL entities; a formal proof is a future work.

- M2E: { *MusicalAlbumSale* } ↔ { gr:offers, gr:Offering, gr:includes, gr:eligibleRegions, mo:album, mo:ReleaseType, mo:MusicalItem, ... }
- M2O: { *MusicalAlbumSale* } ↔ { gr:, mo: }

As regards the link structure *ls* as the third element of the links, for the moment we do not constraint its shape. Its role is particularly important in E2E links having a set with higher cardinality on at least one side, since then the meaning of the link does not trivially follow from the content of its source and target. We assume that such an E2E link structure, depending on the way of use, could typically be represented as either:

- A 'ground' alignment local to the given link; for example, in the E2E example above it may directly specify that the domain of gr:offers corresponds to the originating role of the PURO relation 'offers', gr:Offering is the class of its instance-reifications, and gr:includes and gr:eligibleRegions correspond to its 'what' and 'where' roles.
- A set of instantiations of a reusable *alignment pattern*; here, a reification pattern template that would have slots for the relation and its roles on the one side and for the reification class and the auxiliary (role-modeling) properties on the other side.

Intuitively, there is a partial granularity ordering on the PURO side and a complete one on the OWL side: the links (and thus also whole alignments) involving sets of PURO entities are finer-grained than those involving sets of PURO terms or PURO models, and links (and alignments) involving sets of OWL entities are finer-grained than those involving sets of OWL ontologies.

Thus, although in practical tasks we so far only considered the first three link types of this list, the remaining three can be derived from them (and, actually, from the E2E type alone), since every PURO entity instantiates some PURO term and is in the signature of one (or, possibly, more) PURO models, and every OWL entity is in the signature of one (unless we consider imports and cross-vocabulary links) ontology. Coarser-grained alignments can be constructed from finer-grained ones using (e.g.) set operations, and provide, at least, a useful summary of them. Of the coarse-grained alignment types, only M2E and M2O however seem to be of practical importance, allowing to gather OWL entities (and corresponding RDF data) or whole ontologies corresponding to a domain described by a PURO model. On the other hand, the T2O type is unlikely to be useful, since a reasonable-sized OWL ontology would typically contain entities of various kinds of 'background nature' and would thus always link to all or nearly all PURO terms.

Also note that only an E2E alignment forms a grounding for a PURO model being an OBM for an OWL ontology containing the respective OWL entities, since Definition 1 requires an alignment 'from the elements of M_B to the elements of M_F', i.e. a fine-grained one. Thus, even if a coarser-grain alignment can be designed independently, without an underlying E2E alignment the PURO model is technically not an OBM. Presumably, also, only an E2E an alignment can be (though not necessarily) 'tight' in the sense of allowing to construct a complete fragment of one model based on the fragment of the other model, unidirectionally or even bidirectionally. Such a reconstruction would rely on the link structures *ls* within the constituent links of the alignment.

A more technical aspect is whether the alignment is being maintained on the side of the PURO model, of the OWL model, or independently of both (i.e. what is the 'alignment container'). This choice does not fully depend on the type of links, even if it is influenced by it. We will discuss some 'container' options in connection with the use cases in Section 5.

4.2. Transformation and Alignment Processes

The diagram in Fig. 5 depicts the transformation and alignment processes. On its left-hand side we see the 'PURO world' and on its right-hand side the 'OWL world'. The upper portion of the diagram contains the language metamodels, while the lower schematizes a concrete PURO and OWL model, the latter being further divided into an ontology and a knowledge base (a TBox and an ABox, respectively). Note that a PURO model should by default comprise both these levels (those of universals and particulars, in this context) in an interconnected structure.

In the following we will discuss the horizontal links between the PURO and OWL 'worlds'. The thin arrows correspond to *alignment processes*, and the thick ones to *transformation processes* (though possibly having alignments as their side-effect). We present the different options together with the PURO suite tools that support them, though only briefly, since the tools have already been described in more detail in previous publications.

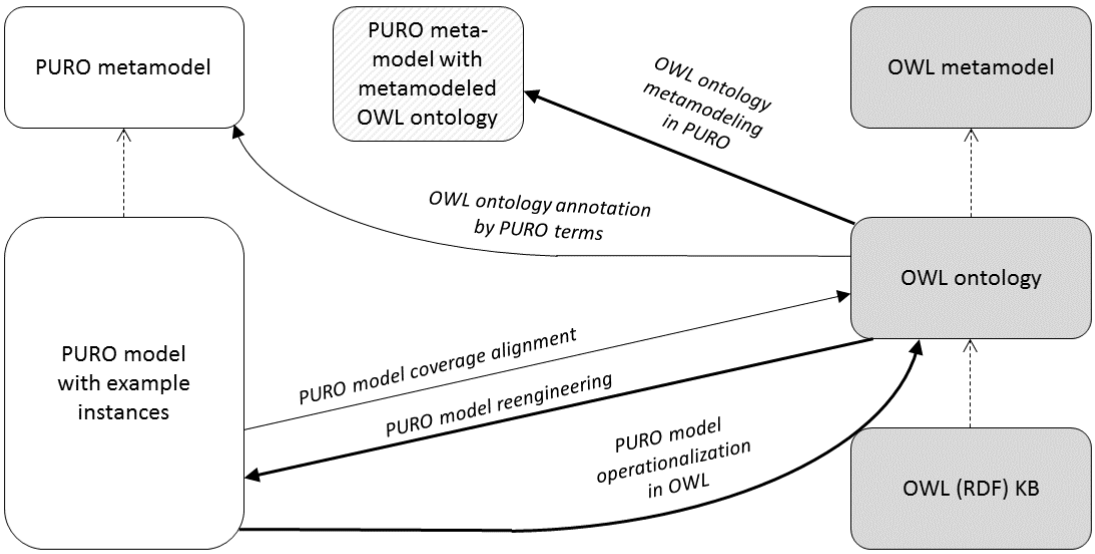

Fig. 5. Transformation processes among PURO and OWL (meta-)models

Note that the scope of the processes is far from exhaustive, both not to clutter the diagram and to stay close to the use cases from Section 5. Beside the processes listed below, we could for example consider direct creation of E2E links between PURO models and OWL ontologies. (In our research, we actually considered this scenario indirectly, through first operationalizing the PURO entities to tentative OWL entities and then automatically aligning these OWL entities with entities from pre-existing OWL ontologies. This is the essence of the use case no. 2 from Section 5.)

4.2.1. PURO model operationalization in OWL

Since PURO is not primarily intended for operations such as reasoning or querying over large instance bases, we view the transformation of a PURO model to an OWL ontology as an 'operationalization', see the bottommost arrow in Fig. 5. Schematically, the arrow 'touches' the OWL KB box, since the example instances from the PURO model, if practically relevant, can be retained for the OWL KB, too.[18] The by-product of the transformation is an E2E alignment between the PURO model and the newly created ontology, at entity level.

Our *OBOWLMorph* transformation tool [18] (see an illustrative screenshot of its UI in Figure 6) allows to generate a fragment (or, skeleton) of an OWL ontology from a PURO model (developed in the PURO Modeler authoring tool [21]), possibly in different encoding styles. The process is carried out by means of transformation patterns [19] translated to SPARQL Update queries, and is tracked by means of simple annotations added to the OWL model in the form `<PURO entity IRI> puro:transformedTo <OWL entity IRI>`; on the side of the PURO model the alignment is currently only stored programmatically inside OBOWLMorph. The transformation cannot be, in the current implementation, viewed as a tight alignment, since E2E links connecting isolated entities would not allow to reconstruct one complete model from the other due to dependencies between structures of statements/axioms. For this a PURO2OWL variant of *alignment patterns*, previously designed for the OWL2OWL setting by Scharffe et al. [67] and by Fillottrani and Keet [24], would be required. Adopting such a tight-alignment mechanism should however not be too difficult even

[18] For the future we envisage that the PURO tools should allow to either manually select those instances ($\mathcal{B}$-objects), $\mathcal{B}$-relationships) and $\mathcal{B}$-valuations) that should form an OWL ABox 'seed', or retain/eliminate them in bulk.

within the current tools, since the patterns would be structurally nearly the same as the already used transformation patterns.

4.2.2. PURO model reengineering

The opposite direction of the transformation would have the character of *reengineering*, since the constructed PURO model should capture the reality at a more abstract level than an OWL ontology. Specific instances gluing the PURO model together could be either designed by hand, or picked from the ABox of the ontology. A by-product would again be an E2E alignment.

The research on such re-engineering is merely ongoing[19] and no computational support for tracking the PURO2OWL links is provided, i.e., the PURO model straightforwardly derived from an OWL ontology is, in the end of the reengineering process, manually authored in the same way as a brand-new PURO model.

4.2.3. PURO model coverage alignment

For an existing PURO model we can also consider an inherently looser (E2O) alignment in the sense of only specifying the whole, existing, OWL ontology covering (in whatever way) the notion of a certain PURO entity, see the corresponding link in Fig. 5. This kind of alignment (which does not involve a transformation) is supported by the PURO Modeler tool as an extra functionality (beyond PURO model authoring), see the use case no. 4 in Section 5.3.

4.2.4. OWL ontology annotation and metamodeling by PURO

A different kind of loose alignment (now, T2E) occurs when the goal is not to construct a PURO model proper but to associate entities from an OWL ontology with PURO metamodel elements (PURO terms). For example, an OWL model can be *annotated* with labels indicating that a certain class maps to a level-1 $\mathcal{B}$-type, to a level-2 $\mathcal{B}$-type, or to some other PURO term. See the 'OWL ontology annotation by PURO terms' link in Fig. 5.

Next, when the goal is to reason about an OWL model in the context of PURO distinctions, it is useful to transform it to a logical space in which this connection could be made accessible for reasoners. Within this transformation, the OWL ontology is *metamodeled* (following the mentioned T2E links), see the 'OWL ontology metamodeling in PURO' link in Fig. 5. The transformation does not create any new alignment.

The first step of the two-phase process (possibly also usable as stand-alone), OWL ontology annotation, can be carried out using our *B-Annot* plugin for Protégé [80]. It allows the user to specify the PURO term with respect to entities of the displayed ontology (leveraging on various information, including the usage patterns of the annotated entities mined from linked datasets) and store them in annotation properties of the OWL ontology and/or as a separate annotation file.

As regards the second, transformation, phase, in our published study [79] the OWL ontology was metamodeled as the instance level of a *PURO meta-ontology*,[20] which operationalizes the PURO metamodel in OWL.[21] Since the OWL ontology entities now become instances of the 'owlified' PURO metamodel, the set of these OWL individuals (PURO meta-ontology instances) can be viewed as a kind of PURO model, in turn, although typically not a contiguous one due to the missing 'instance glue', in contrast to directly authored PURO models.[22]

The transformation can be carried out fully automatically using a pair of *SPARQL Construct queries*, since both its input (OWL ontology with annotations) and output (instances of the PURO meta-ontology) conform to the OWL/RDF data model. This pair of SPARQL queries, one for the OWL ontology and one for the annotations[23] is universally applicable – independent of the ontology to be transformed. It need not be modified unless the PURO metamodel itself or the design choices of the meta-ontology are altered.

[19] There is a set of provisional guidelines used for the first reengineering experiments, available at `http://protegeserver.cz/files/OWLtoOBM_cookbook.pdf`.

[20] `http://patomat.vse.cz/puro_v1.owl`

[21] We might thus add an extra 'operationalization' link to the diagram: one from the PURO metamodel to this special-purpose OWL ontology derived from it. However, this transformation was a one-shot effort, and the choice of OWL as metamodeling language is not mandatory for the whole approach anyway; the meta-ontology could also be represented in a different logically-grounded language without altering the essence of the approach.

[22] Note that the PURO metamodel instances derived from OWL ontology entities are almost exclusively PURO universals and not PURO particulars, since even most syntactic individuals defined in OWL ontologies are rather 'codes' with the semantics of universals, such as `mo:album` from our example. Thus, unless we extract ABox instances from external RDF fact bases or add them manually, the PURO 'proto-models' derived from OWL ontologies would not contain particulars.

[23] `http://patomat.vse.cz/onto-tsf.rq`, `http://patomat.vse.cz/annot-tsf.rq`.

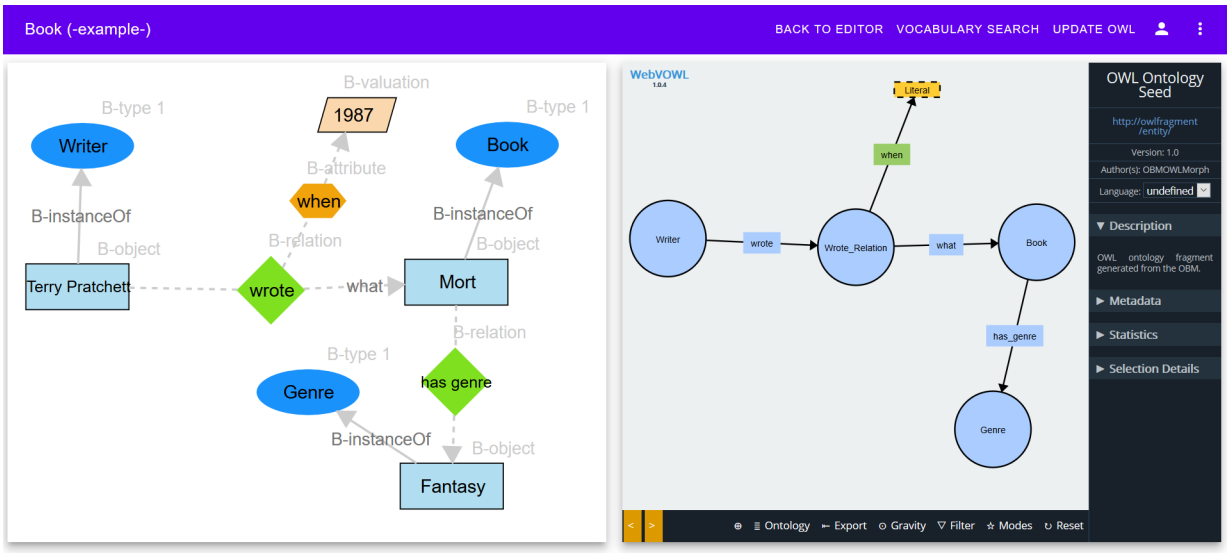

Fig. 6. An illustrative screenshot of OBOWLMorph UI showing an OBM and the result of its transformation to OWL.

Benefits of this kind of alignment and transformation sequence are exemplified by the 'OWL ontology coherence testing' use case described in Section 5.4.

4.2.5. *Summary of alignment and transformation processes*

The tabular summary of the described processes is in Table 2. The 'Type' column distinguishes a transformation process producing an alignment (T→A), a standalone alignment process (A) and a standalone transformation process (T); the rest of the columns and their content are self-explanatory.

5. PURO use cases

PURO is a rather generic tool for which candidate use cases are gradually emerging. Although it could possibly be used as a standalone knowledge representation language, we only consider use cases in which its alignment to OWL models comes into play. Namely, all the use cases described below are built upon the generic transformation and/or alignment processes from Section 4, and the respective subsections thus refer to the names of these processes (from Fig. 5 and Table 2), as well as to the enabling tools.

The majority of the use cases has already been described in previous (conference/workshop) publications. The role of this section is thus to demonstrate the *versatility* of PURO and to outline a high-level *comparison* of the diverse use cases rather than to explain them in depth or bring convincing evidence on the superiority of PURO to alternative approaches in handling each particular problem.

5.1. *Ontology skeleton generation and entity reuse*

When designing an operational ontology or linked data vocabulary, among the crucial requirements typically is that the ontology should properly reflect the reality it models and at the same time fit well the purpose of the foreseen applications. The use of PURO in the initial phase of ontology design, then followed with operationalization (as indicated in Fig. 5) and subsequent elaboration in OWL, looks favorable to both. In PURO the designer is less constrained than in OWL (especially as regards faithful metamodeling and native representation of n-ary relationships or relationships participating on other relationships) and thus can better align the model with the real state of affairs, and OWL ontologies tuned for different purposes can be obtained from a single PURO model. The nature of PURO models as contiguous graphs might also make the designer less prone to forget about some 'essential' segment of the modeled reality. Compared to a list- or tree-based ontology authoring environments, in the graph paradigm of PURO it is not quite convenient to mint families of mutually similar entities such as subtypes of the same supertype; however, in the two-phase approach such 'broadening' can be supplemented in the second, OWL-based, authoring phase, after the architectural skeleton of the ontology has been set up.

Table 2
PURO and OWL alignment and transformation processes

Process	Type	Input	Output	Tools
PURO model *operationalization* in OWL	T → A	PURO model	OWL ontology (possibly also seed of RDF dataset); E2E alignment	OBOWLMorph [18]
PURO model *reengineering*	T → A	OWL ontology	PURO model; E2E alignment	not yet implemented
PURO model *coverage alignment*	A	PURO model; OWL ontology	E2O alignment	PURO Modeler (on top of its authoring functionality) [17]
OWL ontology *annotation* by PURO terms	A	OWL ontology	T2E alignment, as annotation file, or as OWL ontology with annotation properties	B-Annot (Protégé plug-in) [80]
OWL ontology *metamodeling* in PURO	T	annotation file	PURO metamodel and metamodeled OWL ontology as a logical KB	set of SPARQL Construct queries (universally applied to all cases)

We conducted several rounds of experiments, each consisting in making two groups of users create a structured representation of a textually described conceptualization of simple domains; one group started the design in PURO and then automatically converted the model to OWL (using PURO Modeler and OBOWLMorph) while the other used a conventional OWL editor (the desktop version of Protégé). The results (summarized in Section 6.2 below) indicate that the approach involving PURO may lead to somewhat better coverage of the domain, and in some cases may be perceived as more user-friendly.

In the OWL generation phase the entities need not be created from scratch but can also be *reused* from existing OWL ontologies (because of this additional input we view this approach as an additional 'sub-use-case'). The enabling technique is *ontology alignment*: the OWL fragments generated from PURO, possibly in different style variants, are matched to existing ontologies, and the reuse candidates are presented to the user upon selecting a particular node/edge in PURO Modeler [16].

5.2. Ontology (design) pattern analysis and education

Since PURO maps background models of reality to syntactic patterns, it appears as a suitable tool for studying the nature of OWL ontology patterns, be it explicit 'best-practice' patterns or implicit structural patterns used on the fly.

Current best-practice patterns for OWL ontologies are normally expressed in OWL. Namely, ontology content patterns [26] are a kind of directly reusable mini-ontologies expressed as an OWL TBox,[24] while logical/structural ontology patterns, describing the ways the designers can overcome language expressiveness problems, are often illustrated on fragments containing both TBox and ABox statements, see in particular the patterns proposed by the W3C SWBPD workgroup[25] [60,61,66].

PURO seems particularly suitable for studying the nature of the latter kind of patterns. Actually, the W3C patterns are closely related to the expressiveness limitations of OWL that are relaxed in PURO: the binarity of relationships, lack of higher-order relationships (both dealt with by the 'n-ary relation' pattern [61]), and the difficulty of assigning a class as a participant of a relationship (addressed by the 'classes as property values' pattern [60]). Consequently, it is possible to directly model the underlying conceptualizations in PURO and then systematically analyze the ways they can be 'squeezed' back to OWL. For the 'classes as property values' pattern [60] we thus revealed [78] that

- the surface problem of putting a class into the value of a property may correspond to three alternative background states of affairs
- aside the four W3C-endorsed pattern alternatives, at least three further (intuitive) ones can be applied, thus providing several foreground alternatives for each underlying background state of affairs.

In terms of the processes from Section 4, this can be viewed as a special kind of *reengineering* (of both the TBox and ABox of the pattern examples), in which however the result is not a reusable PURO model for a particular domain but a PURO model structure explaining the particular OWL pattern; in such a PURO

[24] See, e.g., `http://ontologydesignpatterns.org/wiki/Category:ContentOP`.

[25] `https://www.w3.org/2001/sw/BestPractices/`

model not only the particulars but even the universals are mere examples, in turn.

A generic analysis of published best-practice patterns is a one-shot endeavor that needs to be carried out by experienced ontologists (familiar both with PURO and ontology patterns) and that only *indirectly* leads the designer towards a suitable expression of his/her modeling problem. We however also worked towards *directly* supporting the designer by allowing him/her to view the different alternative OWL encodings of the same underlying conceptualization in an explicit form. This support corresponds to an 'open-ended' variant of the PURO2OWL transformation from the first use case (Section 5.1): the designer first creates a PURO model (possibly a merely tentative or bogus one, setting up a 'pattern education sandbox') and then compares its OWL style alternatives returned by the transformation mechanism [19]. Note that the foreground alternatives are typically instantiations of the same kind of choices described, at a general level, by logical/structural patterns as the aforementioned W3C ones.

5.3. Analysis of a thematic set of ontologies

The scenario has common traits to some of the previous ones. It is of analytical nature (similarly to design pattern analysis), but instead of patterns, sets of thematically related ontologies[26] are being studied. Analogously to the analysis of patterns, it is possible to start either from an explicit PURO model delimiting the domain in question, or from the OWL ontologies.

The first approach, in which the ontologies are put on a common ground via an explicit PURO model, has been labeled as *local coverage* analysis [17]. It is similar to the entity reuse scenario from 5.1 in the sense that the individual elements of the PURO model are equipped with IRI links to relevant pre-existing OWL ontologies. However, the assignment of these links is manual rather than automatic and its granularity is that of whole ontologies: it is an E2O *coverage alignment* in the sense of Section 4. The graphical rendering of the links (as a kind of Venn diagrams surrounding the PURO elements) in PURO Modeler allows the designer to identify the complementarity/supplementarity of the ontologies to capture data about situations like the one exemplified by the given PURO model (see an example in Fig. 7, corresponding to that from Fig. 4).

[26] Studies of ontology *content* patterns, as mini-ontologies, could actually be ranged under this use case rather than the previous one, which deals with logical patterns without a TBox representation.

The second approach has been applied on a large collection of OWL ontologies that deal with *events* [39]. The different situations foreseen by these ontologies have first been ranged under four categories, denoted as 'Actions', 'Happenings', 'Planned (social) events' and 'Structural components of temporal entities'. Then a (non-deterministic) decision structure for choosing an adequate PURO representation for each category has been proposed; it leads, for example, to 'Actions' being preferably modeled as $\mathcal{B}$-relationships and 'Planned (social) events' as $\mathcal{B}$-objects.

5.4. Coherence checking of ontologies

The motivation for undertaking a journey from an OWL ontology through its annotation by PURO terms to its operational metamodel, as described in the last (two-phase) process from Section 4, may typically be some analytical task. In our study [79] we aimed, in a sense, at an analogy of conceptual constraint verification over 'metaproperties' in the well-known OntoClean method [36]. While OntoClean, e.g., postulates that a rigid class (such as 'Woman') must not be a subclass of an anti-rigid one (such as 'Student'), in PURO we can similarly verify, at least partially, the constraints related to its core distinctions, most notably the $\mathcal{B}$-type layering.

Similarly to Glimm et al. [30] we use metamodeling to represent and interlink both the foreground and the background model in a single ontology. We start from an OWL ontology representing the original foreground model, in which the T2E alignment is stored in the form of annotation labels which are expressed via OWL annotation properties. For example the label 'CTO' means that the annotated foreground entity is a foreground class whose instances correspond to background types whose instances are objects (i.e., the class annotated with 'CTO' corresponds to a 2$^{\text{nd}}$-order $\mathcal{B}$-type).

Throughout the whole task we apply the PURO meta-ontology, which is composed of four loosely coupled modules: (a) the *background module* contains classes classifying the entities according to their background nature, e.g., `B-object`, `B-type1`, `B-type2`, `B-relation`, `B-attribute`, etc.; (b) the *foreground module* contains classes classifying the entities according to their foreground nature, e.g., `F-individual`, `F-class`, `F-obj-prop`, `F-data-prop`, etc; (c) the *labels module* that provides the `hasLabel` property, defines the individuals respective to annotation labels and encodes the semantics of the T2E alignment represented by the la-

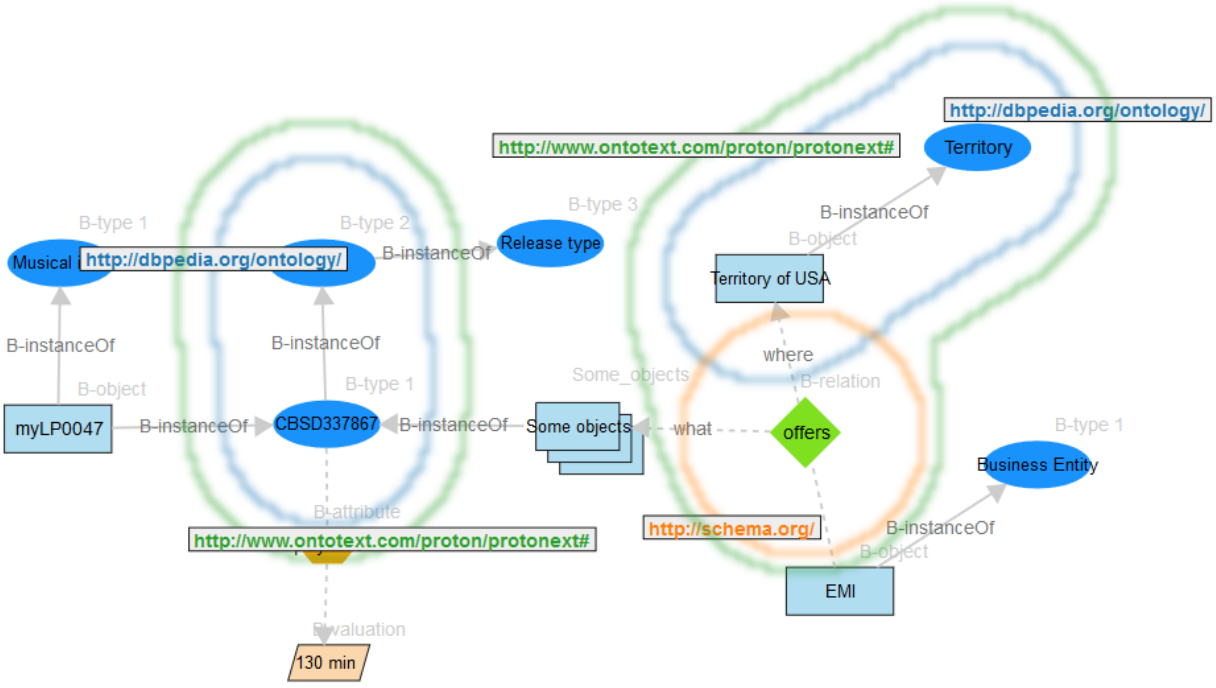

Fig. 7. OBM in PURO Modeler with parts covered by existing ontologies highlighted.

bels, for example the following axiom (for simplicity expressed in DL syntax) takes care of the 'CTO' label:

$$\exists \texttt{hasLabel}.\{\texttt{CTO}\} \sqsubseteq \texttt{F-class} \sqcap \texttt{B-type2}$$

Finally, (d) the *constraints module* derives the background B-instanceOf and B-subclassOf properties from their foreground counterparts, enriches these properties by implications derived from the reconstructed background model, and includes additional axioms reflecting the PURO coherence constraints. For the latter task, the module introduces new diagnostic meta-classes that classify the entities which are derived to be incoherent (e.g., Incoherent-PU classifies entities which are found to be both particulars and universals, Incoherent-T-PU classifies entities which are found to contain both particulars and universals as instances,[27] etc.). The constraints respective to these two diagnostic classes are as follows:

$$\texttt{Incoherent-PU} \equiv \texttt{B-particular} \sqcap \texttt{B-universal}$$

$$\begin{aligned}\texttt{Incoherent-T-PU} \equiv\ & \texttt{B-type} \\ & \sqcap \exists \texttt{B-instanceOf}^{-}.\texttt{B-particular} \\ & \sqcap \exists \texttt{B-instanceOf}^{-}.\texttt{B-universal}\end{aligned}$$

Employment of diagnostic meta-classes for coherence checking allows not only to detect that the foreground vocabulary is incoherent, but also to identify which entities cause the incoherence and to explain what kind of incoherence occurred.

When an OWL (i.e. foreground) ontology is to be coherence-checked, its deductive closure is first computed and stored as T. Then its metamodel T' is built, containing individuals c_C for each class C from T', p_P for each property P from T', and i_j for each individual j from T'. These individuals are then, according to the T2E alignment, associated with appropriate labels (also expressed as individuals, e.g., CTO for the label 'CTO') using the hasLabel property. Also, the (foreground) instance-of and the subclass-of relations from the closure T are explicitly encoded in T' using the properties F-instanceOf and F-subclassOf (from the foreground module of the meta-ontology), respectively.

Our study [79] is preliminary and partial w.r.t. the full axiomatization of coherence constraints specified in Section 3. Exploitation of this full axiomatization or

[27] Note that in our previous report [79] the axoim corresponding to Incoherent-T-PU contains an error. The correct version is stated here and it is also found in the PURO meta-ontology at http://patomat.vse.cz/puro_v1.owl.

its tractable subset for automated coherence checking is a matter of future work and is beyond the scope of this paper.

5.5. Extended versions of coherence checking

There are also two variants of the base use case of single-ontology coherence analysis, which have both been investigated in the context of the EU LOD2 project.[28] The focus of this project was on large RDF datasets and on interlinking at the level of both data and ontologies; therefore, the sub-use-cases aimed at leveraging on interlinked ontologies and on ontologies populated by RDF datasets. Both sub-use-cases are described in sections (2.7 and 4, respectively) of a project deliverable [77].

First, the combination of two ontologies *aligned* using equivalence links can be examined. This modification is straightforward: the entities from both ontologies are merged and the entities (classes or properties) linked with equivalence links in the (OWL2OWL) alignment are metamodeled by individuals that are declared as being the same (`owl:sameAs`).

Second, the ontology to be annotated may be considered not only at the level of its generic specification but also with respect to its *usage in a certain dataset*. The B-Annot plugin provides, for datasets it has previously summarized, lists of frequent paths, in which the entities from the current ontology are highlighted. The displayed paths are currently of length 1 only: connecting a class, a property and another class, such that instances of the classes frequently appear together with the property. Examples of such paths from different datasets may be `{mo:MusicArtist, foaf:made, mo:Record}` and `{gr:BusinessEntity, ex:sells, mo:Record}`. We may then expect that the tendency of the annotator to associate the same foreground entity, `mo:Record`, as a background particular ($\mathcal{B}$-object) rather than a background universal ($\mathcal{B}$-type), would be higher in the context of the first path (artists only make a record once) than in the context of the second path (individual exemplars of a record are sold many times).

The *projection* of an ontology to a dataset may influence its annotation and subsequent coherence checking in at least two ways:

- entities suspected for incoherence according to their PURO term labeling assigned based on the ontology specification might *not be used* in a particular dataset, thus the projection may be coherent even if the ontology as such is not
- the entity co-occurrence obtained from the dataset may indicate its usage differently from that indicated by the specification, which may lead to annotation by *a different PURO term*, thus affecting the coherence checking result, too.

With this functionality in effect, the task of coherence testing may be transferred from the testing (for overall quality assessment, or caveat identification) of ontologies to the testing of RDF datasets, though with a 'schema flavor'.

As mentioned, B-Annot only provides paths of length 1, since longer paths are impractical to serve in a tabular format due to proliferation of partially overlapping paths. However, external dataset summarization tools can also be applied that heuristically integrate frequent entity paths into frequent graphs. Of these, *LODSight* [20] (which partly evolved from the B-Annot summarization component) offers not only a graph over frequent paths but also various filtering options, of which the most relevant for the discussed use case is the filtering by ontology. This way, not only a dataset summary but also a projection of a single ontology (or a subset thereof) to the dataset can be visualized in terms of the frequent type co-occurrence graph.

5.6. Extraction of 'semantic' concise bounded descriptions (CBD)

The notion of *concise bounded description* of an RDF resource within its graph [72] has been proposed as an 'optimal unit of specific knowledge about that resource to be utilized by, and/or interchanged between, semantic web agents'. The motivation is that naïve extraction of the graph neighborhood of the resource leads to inclusion of auxiliary resources that fully depend on external resources (that are not included in the description). The initial kinds of 'dependent resources' are blank nodes and instances of `rdf:Statement` (i.e., RDF reifications). However, the same treatment would be deserved by instances of OWL classes that correspond to a $\mathcal{B}$-relationship – either via a link to an *explicit* $\mathcal{B}$-relationship in a PURO model (from which the ontology has been derived, see Section 5.1) or due to being annotated (as explained in Section 5.4) by an annotation referring to the PURO term BRelationship. This could prevent the extraction of semantically incomplete RDF descriptions, e.g., of a person being de-

[28] `http://lod2.eu`

scribed as involved in a marriage without mentioning the spouse.

This use case is still in the phase of an initial idea. However, given the popularity and importance of huge linked datasets, it might turn one of the most important ones in the future.

5.7. Tabular summary of the use cases

The characteristics of the different use cases are summarized in Table 3. The number of rows is higher than that of the use case sections above, since there are separate rows for use case variations described together with their base use case. Detailed use cases no. 1 and 2 correspond to Section 5.1, no. 3 and 6 to Section 5.2, no. 4 and 5 to Section 5.3, no. 7–9 to Sections 5.4 and 5.5, and no. 10 to Section 5.6.

The columns contain, in turn:

- The use case number
- Its verbal description
- The distinction whether an *explicit* PURO OBM appears in the process underlying the use case, or whether mere PURO terms are used for annotation, i.e. the model is only *implicit* (this column is only provided for better readability: it is fully correlated with the PURO side of the alignment in the 'Alig.' column)
- The distinction whether a *real* OWL ontology is involved in the process (or only temporary/artificial OWL fragments), and whether this OWL ontology pre-dates the process (i.e. is *prior*) or is rather its *posterior* result; in some use cases specific types of OWL-compliant knowledge/data also appear such as alignments or RDF datasets
- The PURO2OWL alignment type (as mentioned, only three of the six theoretical options appear here)
- The processes (cf. Table 2) names of software tools developed or reused to support the use case
- Some expected benefits of using PURO compared to addressing the same (or analogous) use case without PURO
- The maturity (or, degree of elaboration) of the use case, ranging from a merely hypothetical one through proof-of-concept cases tested on a handful of problems to relatively mature cases addressed via a kind of systematical analysis: either at the theoretical level or experimentally
- References to papers (published, or in preparation) and reports fully or partially devoted to the given use case.

Table 3
Summary of the use cases

No.	Use case description	PURO	OWL	Alig.	Processes and tools	Expected benefits of PURO usage	Maturity	Publ.
1	OWL ontology skeleton generation from PURO (Sec. 5.1)	Explicit	Real ontology, posterior, possibly multiple styles	E2E	Authoring in PURO Modeler; *operationalization* in OBOWLMorph	More complete and correct OWL models; convenience of graph editing; amenability to re-engineering the OWL model to a different style	Partially evaluated on real data	[18,21]
2	…with entity reuse	Explicit	Real ontologies, prior	E2E	As in no. 1, plus OWL2OWL alignment	Reuse sensitive to style pattern adequacy	Proof of concept	[16]
3	Exemplification of OWL style patterns (part of Sec. 5.2)	Explicit	Temporary ABox and TBox, posterior	E2E	As in no. 1	Educational impact: better awareness of style patterns	Proof of concept	[19]
	Analysis of a thematic set of ontologies (Sec. 5.3)							
4	…wrt. local coverage	Explicit	Real ontologies, prior	E2O	*Coverage alignment* in PURO Modeler	More effective reuse of ontologies	Proof of concept	[17]
5	…wrt. state of affairs	Implicit	Real ontologies, prior	T2E	Manual *annotation* of entity *categories*	More adequate reuse of ontologies	Completed study	[39]
6	Analysis of OWL design patterns (part of Sec. 5.2)	Explicit	Pattern examples (TBox + ABox), prior	E2E	Manual *annotation* of patterns; reengineering of examples	More adequate reuse of design patterns	Completed study	[78]
	Coherence testing (Sec. 5.4+5.5)							
7	…of a specific ontology	Implicit	Real ontology, prior	T2E	*Annotation* of entities in B-Annot; *metamodeling* via SPARQL	Ontology quality evaluation; awareness of issues possibly arising after ontology reuse	Proof of concept	[79,77]
8	…of aligned ontologies	Implicit	Real ontologies and their alignments, prior	T2E	As in no. 7	Improvement of ontology alignment	Proof of concept	Sec. 2.7 of [77]
9	…of an ontology projection to a dataset	Implicit	Real ontology, prior; RDF dataset	T2E	As in no. 7, plus dataset summarization (possibly by LODSight)	RDF dataset quality evaluation	Proof of concept	[77]
10	Semantic CBD generation (Sec. 5.6)	Explicit or implicit	Real ontology, prior or posterior; RDF dataset	T2E or E2E	*Annotation* of entities in B-Annot, or as in no.1; plus CBD generation	Better completeness/conciseness ratio of RDF data descriptions	Hypothet.	None

6. PURO evaluations

The huge scope of use cases and settings for applying PURO has so far been only very partially covered by empirical evaluation efforts. Two different kinds of evaluation studies took place so far.

The primary research questions of the first evaluation study, associated with the 'annotation' mode of PURO-to-OWL mapping (cf. Sec. 4.2.4 and 5.4), were:

- **Q1** Does the foreground structure of real, commonly used OWL ontologies significantly differ from that of their underlying PURO models, in terms of their metamodel 'isomorphism'?
- **Q2** Are expert ontologists able to find a consensus on the type of a PURO entity underlying an existing OWL entity?
- **Q3** Can minimally trained ontology engineers correctly classify the type of a PURO entity underlying an existing OWL entity?

A positive answer to the first question is important for demonstrating that PURO brings an added value and is not just a contrived philosophical exercise. The answer to the second and third question is crucial wrt. the possibility to reengineer PURO models from OWL models, whether the aim is (only) to test the coherency of the latter, or (which is more ambitious) to develop PURO models that can be further reused for building alternative OWL models with different encoding.

The second evaluation study, or, rather, a suite thereof, is linked to the scenario of bootstrapping OWL ontology skeletons from PURO models (cf. Sec. 4.2.1 and 5.1). Its primary research questions were:

- **Q4** Can minimally trained ontology engineers identify the type of PURO entity mentioned in a free text, and compose the respective PURO entities into a complete PURO model describing the given situation?
- **Q5** Can the resulting PURO models be transformed into usable skeletons of OWL models that have quality comparable to models directly authored in OWL based on the same described situation?

This second suite of experiments has been closely tied to the reference implementation of PURO in PURO Modeler and OBOWLMorph.

The raw data for both evaluation studies are available from `http://bit.ly/SWJ2019PURO`.

Table 4
Meaning of PURO labels of OWL classes

Label	Instances of the class correspond to
CO	$\mathcal{B}$-objects
CR	$\mathcal{B}$-relationships
CV	$\mathcal{B}$-valuations
CTO	$\mathcal{B}$-types ('types of objects')
CTR	$\mathcal{B}$-relations ('types of relationships')
CTV	$\mathcal{B}$-attributes ('types of valuations')

6.1. Evaluation 1: Labeling OWL entities by PURO terms

The experiment consisted from two phases. The first phase aimed at answering (at least provisionally, wrt. a manageable data sample) the research questions Q1 and Q2. It also generated ground truth data for the second phase.

6.1.1. First phase: labeling by expert ontologists

Three popular, lightweight but nontrivial OWL ontologies, already mentioned in this paper, were examined by two experienced knowledge engineers (coauthors of this paper, VS and MV): FOAF, MusicOntology (MO) and GoodRelations (GR). Each class from these ontologies was categorized according to the PURO meta-type of its *instances*, with support of previously set up brief *annotation guidelines*. For example, the category "class of types of objects" (labeled as CTO, see also Section 5.4) was defined in the guidelines as follows: "Instances of such a class refer to *types*, which can have their own instances." On the other hand, the category "class objects" (labeled as CO) was defined as follows: "Instances of such a class refer to true *instances*, i.e. objects that cannot be understood as *types* and cannot have their own instances (unlike type CTO). At the same time they are truly objects and not *relationships*: they can be considered standalone, and not just in connection with other objects (unlike type CR, i.e. class of relationships)". A brief summary of the label meaning is in Table 4.

The two annotators assigned the categories partly independently; they however picked up tricky cases and discussed them, which led to partial evolution of the annotation guidelines. Subsequently they reviewed the results in bulk and tried to arrive at a consensus in the remaining less clear cases. In the end, in a vast majority of cases the adequate PURO label was determined

either as 'obvious' or at least as 'dominant';[29] in class annotation, a single label was seen as 'obvious' in 82 cases (87.2 %) and 'dominant' in another 10 cases (10.6 %), out of 94. In Table 5 we present the summary results: the total numbers of classes[30] in the ontologies, and the corresponding PURO category label frequencies (only considering the first-choice, i.e. 'obvious' or 'dominant' labels).

The first phase of the study thus corroborated the assumptions behind the relevant research questions: Q1 since only 57 classes (61 %) were labeled as CO[31] (i.e., class of resources corresponding to $\mathcal{B}$-objects); and Q2 since 92 classes (98 %) were successfully labeled.[32] As a by-product, the conducted exercise also indicated that

- The labels can mostly be determined using simple annotation guidelines. For borderline cases, at least the 'dominant' label can be established this way.
- Reliance on lexical information solely (labels and descriptions) can lead to mis-interpretation; interrelated entities should be examined, too.
- There are differences in the distribution of labels among different vocabularies, presumably related to the intended purpose of the vocabularies (which may not however be always obvious from browsing the intro texts of the specifications).

[29] An example of 'dominant' label is CTO for `mo:Instrument`. Its use in the range of properties such as `mo:primary_instrument` clearly indicates the character of $\mathcal{B}$-type, however the description of the class does not strictly preclude its use for $\mathcal{B}$-objects, e.g., individual instruments exhibited in a museum.

[30] Besides, the annotation concerned individuals defined in the ontology, and property ranges. However, the consensus-reaching process was not yet concluded for property ranges in the time of writing the paper, due to the very high number of properties in MO.

[31] Apparently, FOAF exhibited clean alignment to its PURO OBM at the level of classes. However, as regards the range of its properties, the labels indicating types (PrT: properties valued by resources corresponding to $\mathcal{B}$-types) or identifiers (PrId: properties valued by identifiers without ontological relevance) were nearly as numerous as the 'default' PrO label (property valued by resources corresponding to $\mathcal{B}$-objects, as 'PURO equivalent' to object property) and PrV (property valued by resources corresponding to PURO data values, as 'background equivalents' to data properties).

[32] The two classes that 'resisted' labeling were the 'utility' class `foaf:LabelProperty` (which is not part of a domain but in fact metamodels the domain of 'LOD vocabularies' itself) and the class `gr:ProductOrService`, which overroofs subclasses strictly incompatible in terms of the PURO model: `gr:Individual`, `gr:ProductOrServiceModel` and `gr:SomeItems` (i.e. classes of individual products, product types and placeholders for multiple individual products, for the last recall the MISO pattern in Section 3.10).

Table 5
PURO labels of vocabulary classes (consensus of experts)

Vocab.	# Classes	CO	CR	CV	CTO	CTR	CTV	Other
FOAF	13	12					1	
GR	28	4	7	6	4	5	1	1
MO	53	41	7		5			
Total	94	57	14	6	9	5	2	1

Table 6
Expert PURO labels of classes chosen for labeling by students

Choice	CO	CR	CV	CTO	CTR	CTV
1st	7	4	2	5	2	0
1st + 2nd	10	6	4	5	2	0

6.1.2. Second phase: labeling by students

Experimental setup The second phase of Evaluation 1, in turn, involved 13 MSc-level students of a linked data course, aiming to answer the research question Q3 (as a more ambitious variant of Q2). 20 classes from two ontologies, GR and MO, were chosen for the evaluation, 10 from each. For 13 of them the experts considered the PURO label choice as unambiguous; for the remaining 7 they also provided the second most plausible one; see Table 6 for a summary overview. The classes were selected not randomly but with respect to covering a wider range of cases: 1) the coverage of labels other than CO was increased, and 2) multiple classes semantically similar to one another were avoided, especially as regards taxonomic siblings. All 20 selected classes were present in the assignment sheet of each student, but in a different (random) order. The students were to choose one most plausible label per class, and also provide their choice with a *confidence degree* ranging from 3 (highest confidence) to 1 (lowest confidence). The translated instructions as well as the list of classes are in Appendix A.

Results The experiment yielded 260 data points overall (20 classes × 13 students), all with their associated confidence. The correlation of the collected data was then calculated both with the experts' first choice only, as a kind of *strict* correlation, and with the disjunction of the first and second choice (where it was present), as a kind of *relaxed* correlation. Furthermore, the impact of confidence was taken into account.

The main results are summarized in Table 7. The results achieved by students significantly exceed the

Table 7
Aggregate result of the labeling experiment

	Strict	Relaxed
Baseline – random	17%	23%
Baseline – majority	35%	50%
Students	42%	55%
Confidence level 3 (40 labels)	45%	53%
Confidence level 2 (112 labels)	34%	46%
Confidence level 1 (108 labels)	49%	67%

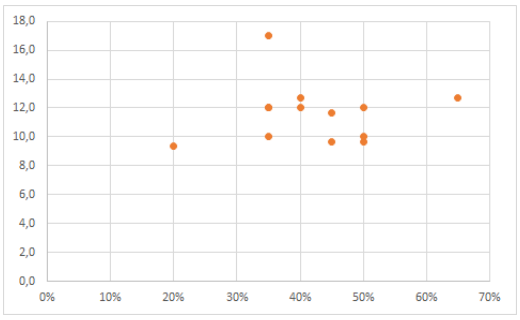

Fig. 8. Labeling confidence (Y-axis) vs. success rate (X-axis) for strict mode

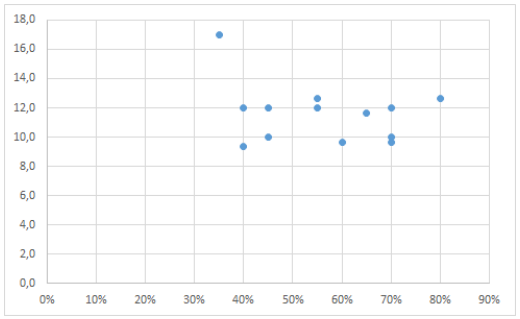

Fig. 9. Labeling confidence (Y-axis) vs. success rate (X-axis) for relaxed mode

baseline consisting in random choice among the six labels (in strict mode it is $1/6$ for all classes; in relaxed mode it is $1/3$ for classes having a 2^{nd} choice and $1/6$ for the rest). The students' results are also better than the majority rule (always choosing CO) as an alternative baseline. However, the difference is not statistically significant – despite the fact that the proportion of CO-labeled classes is lower in the evaluation sample than (most likely) in the real world. To assess the statistical significance of our results, we used binomial and Mann-Whitney U tests. Both tests suggests the difference between student performance and the baseline of the majority rule is significant in case of the 'strict' correlation and not clearly significant in the 'relaxed' case. Using the binomial test is quite a simplification as it takes each classification attempt by each student as an independent observation. However, thanks to it simplicity, it directly reflects the aggregated values from Table 6. One-tailed p-value of the student performance in 'strict' mode (success rate of 42%) with the majority rule performance of 35% taken as expected success rate is approx. 0.0035, way below 0.05 threshold. In 'relaxed' approach, it would be 0.0041. Mann-Whitney U test p-value when taking average accuracy of each student as one set of observations and accuracy always equal to the baseline as another is less then 0.00001 in 'strict' case and approx. 0.089 in 'relaxed' case. The former then being significant and the latter not significant at $p < 0.05$.

The impact of the confidence degree was inconclusive if not surprising. The agreement was slightly higher for labels with confidence 1 ('least confident') than for those with confidence 3 ('most confident'), while for confidence 2 it was even below the majority rule baseline. Also looking at the scatterplots in Fig. 8 and 9, where the overall confidence per student (Y-axis) is roughly aggregated as a weighted count of the confidence degree occurrence,

$$Conf(s) = |l(s, L, 3)| + \frac{2}{3}|l(s, L, 2)| + \frac{1}{3}|l(s, L, 1)|$$

where $l(s, L, Cf)$ is the assignment of label L by student s with confidence Cf, there is no sign of more confident labeling decisions being better in terms of average success rate of the student (X-axis).

We also constructed the confusion matrix of the labeling task. For the strict mode of evaluation, we simply compare the experts' 1^{st} choice with the student's label. In the relaxed mode we exclude the cases when the student's label matched the experts' 2^{nd} choice; we thus only consider cases when either the student's label matched the 1^{st} choice or did not match any choice. For better readability, we do not display the absolute values but the percentage, in Table 8 and Table 9. Counts of correct matches on the main diagonal are left out, as is the row for CTV (as there wasn't any class labeled so by the experts in the testing sample). Percentage values of 25% or more are in bold.

In Table 8 we for example see that students often mistook CR for CO; this is mainly due to class `mo:Recording`, which the experts viewed (according

Table 8
Confusion matrix in Evaluation 1 as percentage of expert's 1^{st} choice, strict mode; values ≥ 25% in bold

		Students					
Experts	#	CO	CR	CV	CTO	CTR	CTV
CO	91	-	5%	8%	24%	1%	0%
CR	52	**25%**	-	**31%**	12%	10%	0%
CV	26	4%	8%	-	12%	8%	**31%**
CTO	65	**35%**	5%	14%	-	5%	0%
CTR	26	15%	23%	4%	**42%**	-	0%
Total	260						

Table 9
Confusion matrix in Evaluation 1 as percentage of expert's 1^{st} choice, relaxed mode (matches to 2^{nd} choice left out); values ≥ 25% in bold

		Students					
Experts	#	CO	CR	CV	CTO	CTR	CTV
CO	87	-	5%	8%	24%	1%	0%
CR	38	**34%**	-	5%	16%	13%	0%
CV	26	4%	8%	-	12%	8%	**31%**
CTO	45	7%	7%	20%	-	7%	0%
CTR	25	16%	20%	4%	**44%**	-	0%
Total	225						

to its context and specification) as a class of reified relationships connecting a sound and a signal, while the students probably judged it as a class of musical items. CR was also often mistaken for CV, but only in the strict mode; this was mainly due to classes `gr:DeliveryChargeSpecification` and `gr:WarrantyPromise`, which however had CV as their 2^{nd} choice. Analogously to the change of CR to CO, there often appeared its meta-level counterpart, CTR to CTO. This most often concerned the class `gr:WarrantyScope`.

6.1.3. Summary of Evaluation 1

We have to conclude that while the first phase of Evaluation 1 yielded positive results wrt. research questions Q1 and Q2 (as we noted previously), its second phase, involving students as 'novice ontological engineers', *did not* positively answer the research question Q3. It rather indicated that plausible PURO category labeling of OWL ontologies, never mind their re-engineering to fully-fledged PURO models, might require thorough ontologistic *experience* and/or *intuition*. The impact of the latter (intuition) might, in turn, be conjectured based on the distribution of the students' success rate, see the X-axis values in Fig. 8 and 9. The rate ranges between 20%–65% in the strict mode and between 35%–80% in the relaxed mode. We may hypothesize that a certain level of correct PURO labeling might be reached via rapid grasping the gist of PURO at the conceptual level (from simple guidelines) thanks to some intuition (or, innate 'ontologistic thinking'?), even without having to first observe a huge number of examples. Reaching a success rate close to 100% would however still require either following rather detailed guidelines or having thorough experience with foundational aspects of ontologies.

6.2. Evaluation 2: Text-grounded development of PURO models transformable to OWL models

The second experiment, also carried out in the context of classroom assignments, aimed to verify to what degree will the 'expert ontologist bottleneck', revealed by the previous experiment, persist if we move from the labeling scenario to the (graphical) PURO authoring scenario. Indication of the PURO term choice is thus no longer an OWL model but a textual description of a situation, as foreseen in Sec. 5.1.

In this direction we have performed two evaluation rounds with users so far. After the first round, we tried to improve the user interface of PURO Modeler, mainly by adding simple syntax checking, and then performed a second evaluation with similar settings, using slightly different tasks, more users and the System Usability Scale [8] questionnaire (SUS). As we found out the tasks given to the users in the first round had been too time-consuming, we prepared simpler tasks for the second round. Results of the first round of this evaluation have already been published [21], we therefore only include their shortened overview. The second round is described here for the first time, in turn.

6.2.1. Experimental setup

First round The experiment[33] involved with 10 MSc-level students of an ontology engineering course.[34] The students had basic knowledge about OWL (understanding of clasess and their hierarchy, instances,

[33] Details about the evaluation not present in the paper [21] are at `http://protegeserver.cz/puroeval`.

[34] Their background could theoretically have better covered ontological engineering proper than that of students from the first, 'labeling-oriented', evaluation, who had rather been trained in linked dataset management. In practice, however, the erudition of students in 'semantics' topics did not much differ between the two groups.

properties and domains/ranges) from the course lectures and had taken tutorials about Protégé and PURO Modeler (each about 90 minute long).[35] We prepared 2 example situations to be modeled by the students: an example from air transportation (A) and a human relationships example (B). Five students were asked to model A with Protégé, i.e., to create an ontology that will cover the described example, and to model B with PURO Modeler, i.e., to create an OBM that could be transformed to OWL ontology covering the example. The remaining five students modeled A in PURO and B with Protégé. Each student had 45 minutes to accomplish both tasks.[36] Each example consisted of (1) an abstract description of entity and relationship types and (2) an example situation from the domain. To make the results comparable, we instructed the students to only create classes, properties and subClassOf and domain/range axioms in OWL, i.e., omit any possible complex classes, restrictions etc. since these are not created by the PURO-to-OWL transformation in OBOWLMorph. We measured the time each student needed to accomplish each task and then we examined the resulting ontologies and OBMs handed in by students. The students were also given a questionnaire focusing on comparison of PURO and direct OWL modeling at the end of the evaluation.

Second round The experiment involved 26 users – now students attending a linked data course (i.e. a different than in the first round, but the same one as in Evaluation 1). The experiment had two phases or turns. Approximately half of the students attended the first one, and the remaining ones the second one, i.e. each student attended one turn. In each turn, the students were divided into two groups in a similar way as in the first round. At the beginning, the students were given a 10 minute tutorial on PURO Modeler and the same for Protégé. They had also some preliminary experience with both tools from previous lessons. We prepared two simple situations to be modeled by the students: (A) representing an offer of a laptop with a few parameters and (B) about a person who read a book.

The situation presented in assignment A was as follows:
General domain description: This is a domain of sellers and their offers of laptops. A seller can offer a particular laptop model for a specific price. A laptop is characterized by having a specific CPU model, hard drive type and an amount of installed RAM.
Example situation: Seller DalZa offers a laptop model Bell Longitude E1234 for 15 250 CZK.[37] *The model has the CPU of type More i7-2503 and a hard drive of type SSD. It has 8 GB of RAM.*

The situation presented in assignment B was as follows:
General domain description: This is a domain of persons reading books. Each person lives in a city and has specific age. A person can know another person and read books of specific genres.
Example situation: John Doe lives in Prague and is 25 years old. He has read Mort, which is a fantasy genre book. He knows Peter Wellknown.

We made the tasks as simple as possible while keeping most of the types of relations and entities in them – trying to learn from the caveat of the first round, where most students did not manage to finish both tasks.

The students in the first group were assigned to model A in OWL using Protégé, and after they finished it, to model B in PURO Modeler. The second group modeled A in PURO Modeler and then B in Protégé. Before starting the second task, each student was given a SUS questionnaire about the tool s/he was just working with, and the same applied after finishing the second task. When modeling in PURO, the students were asked to generate the ontology skeleton from the PURO model, using the default configuration of encoding style and omitting possible reuse of existing ontologies. They handed in both the PURO model and the ontology skeleton.

6.2.2. Technical arrangement

In both rounds, all students performed the tasks at the same time in one room, using either the desktop PCs available there or their own notebooks. One of the authors was present during the experiment to assure each student worked separately, without communication with others. Students sitting next to each other were assigned different groups to minimize influence between them. No signs of online interaction between the students were intercepted; while such an effect cannot be completely ruled out (there was no technical measure applied to avoid such interaction), it is rather unlikely, since 1) the modeling decisions are relatively complex and hard to convey via simple hints, and, in particular, 2) the students were only (mildly) incen-

[35] Based on user and modeling guides, see `http://protegeserver.cz/puroeval`

[36] Due to lack of time, only 2 students finished both tasks. We took even unfinished results into account as they still allowed us to see what errors the students made.

[37] Czech crown (local currency unit).

tivised to take part in the experiment but not to deliver 'good' results by some external measure. We therefore assume that the results achieved indeed reflect the varying skills and intuition (and possibly also degree of enthusiasm for the given task) of *individual* students.

6.2.3. First round results

Correctness We classified the errors students made into three levels. Level 1 errors are against the syntax of the language (PURO or OWL). In PURO these are for example missing labels or wrong orientation of links (e.g., when both links originate in a binary relation node, making the orientation of the relation unclear). Level 2 errors are such that are correct syntactically, but do not make sense in the language: for example economy-class being a subclass of plane. Level 3 errors occur when something is modeled differently than in a reference model created by us. These 'errors' are thus rather just differences from our version of the model. As there are usually many correct ways of modeling the same thing, it is not necessarily an error when something is modeled differently. We call them 'errors' to keep consistence with level 1 and level 2 error naming system, but they are not an important indicator. We also tried to evaluate the overall severity of errors – whether the errors are critical and affect the whole model, or non-critical where at least part of the model is correct. An example of a critical error in the case of PURO is when the students created only the types and did not include the instance-level entities.

There were no level 1 errors in the OWL ontologies as Protégé does not allow to make such errors. PURO Modeler checks for only some such errors and an obvious conclusion is that the application has to check for all syntactic errors as they occur very frequently: 7 out of 10 PURO models contained level 1 errors, and in 4 cases these were critical.

The amount of level 2 errors is quite comparable between the two tools. 6 of 9 OWL ontologies (one student did not hand in the result) and 4 of 6 syntactically mostly correct PURO models[38] contained such errors, which is actually the same percentage. A common critical-level 2 error in case of PURO models was that of modeling only the 'Tbox' part.

There were no critical-level 3 errors, i.e., all models without critical level 1 or 2 errors could be used without major changes. There was 1 PURO model without error, however unfinished due to lack of time. No OWL ontology was without errors. The students had problems modeling n-ary relationships in OWL. For example, no one was able to model the "is angry at someone because of something" relationship.

Model coverage The models that did not contain any critical errors were evaluated in terms of coverage, i.e., whether they covered all relationships and entities described in the real-world situation. The entities or relationships were considered covered even when there were minor errors in the model. Based on that, we found out that only 1 OWL ontology had complete coverage, in contrast with 3 such PURO models. However, due to the low number of participants, the result was not statistically significant according to Fisher's exact test: the p-value was approx. 0.582, which is way above the threshold of 0.05. We would need more than 30 test-users with such a relative difference in success to have a reasonable probability that the difference did not occur by chance.

Time Relevant time measurement is only for the first task (example A), as most students did not finish the second task. The average times were 32 minutes to create an OBM and 26 minutes for an ontology created directly in OWL. The medians were 28 and 27 minutes, respectively. Given the low number of measurements (only 5 per each method), we could only assume that there are probably no extreme differences in time-consumption between the two approaches. We would need more experiments to obtain statistically sound results.

6.2.4. Second round results

After the experiment we analyzed the correctness and coverage of the resulting models; their visualizations, which we used for the analysis, are available online.[39] We tried to also measure the time again, but due to technical problems with PURO Modeler, we do not take time measurements into account.

Correctness and coverage (overall) We analyzed the correctness and coverage of all models created by users in a similar manner as in the first round. Tables 10 and 11 shows the results. Each row represents one user (users in Table 11 are different from those in Table 10). The first six columns describe the models created in PURO, while the remaining five columns describe the models created directly in Protégé. Table 10 refers to students working on the "offer" model in PURO

[38] We did not check syntactically incorrect models for level 2 errors as it is meaningless.

[39] `http://protegeserver.cz/puroexpII/`

and "books" model in Protégé (OWL). Users in the second table had the models switched the other way around. The Tech error column tells whether a technical error occurred. The columns for Error Level 1, 2 and 3 indicate whether the student made the respective kind of error in the model. The Covers All column tells whether all entities and relations were somehow modeled – regardless of correctness. The Usable Ontology column shows whether the resulting ontology skeleton (the OWL fragment created in Protégé or OWL generated from the student's model in OBOWLMorph) can be actually used to describe the situation assigned to students at the beginning, i.e. whether it contains all necessary classes and properties to encode the given information in RDF. The "N/A" values in the table mean that the model was not finished by the user or not handed in at all. In the case of PURO models, all aspects except "Usable ontology" were analyzed directly on the PURO model; i.e. only the usability of the ontology was assessed based on the ontology skeleton generated by OBOWLMorph from the model.

To decide about the coverage and usability of each model, it was first independently rated by two authors of this paper (VS and MD) and one external ontology engineering expert. We then took the major opinion as the final decision about coverage and usability. Disagreement was very rare and subsequent consensus was then reached in all cases. We fully agreed on about 96 % of models both about coverage and usability. More precisely, the inter-rater agreement counted as Fleiss' kappa was approx. 0.93 in both cases.

In total, 24 PURO models and 17 OWL fragments were handed-in by the students. We assume that the reason for this difference were various technical issues or some wrong interpretation of the assignment fulfillment, and thus do not take this fact into account – we only analyzed the models we had received and did not investigate why some might have not been handed in due to the lack of logging and other data. Of the 24 PURO models, 15 covered all facts from the description and 11 resulted in a usable ontology. On the other hand, only 9 OWL fragments out of 17 had complete coverage and 8 (of those 9) were considered as usable ontologies. In other words, the coverage was complete in about 63% of PURO models compared to 53% of OWL fragments. In terms of effect size, using the odds-ratio approach, we could say there is about 48% higher chance of creating a model with complete coverage when using PURO compared to Protégé. However, as we discuss later, an experiment with more participants is needed to confirm that. Ontologies considered as usable were created in 45% cases of work with PURO Modeler and in about 47% of cases of work with Protégé.

To sum up, the evaluation results suggest the users were similarly successful in terms of creating a usable ontology skeleton in both PURO Modeler and Protégé, but in PURO Modeler the students tend to omit facts from the description less often than in Protégé. This is more noticeable in the "offer" example, where 3 of 8 (about 38%) OWL models had complete coverage compared to 10 of 15 (about 67%) PURO models. We assume it is due to the fact that the "offer" example included an n-ary relation (a laptop is offered by a seller for some price) and the students tend to forget some participant of the relation, usually the seller, when modeling directly in OWL. However, none of the above-described results represent statistically significant differences. We calculated Fisher's exact test for all the above described values. The lowest p-value is in case of the number of complete models of the "offer" task and it is approx. 0.22, while it should be lower than 0.05 for the results to be considered significant. While the representativeness of the result somewhat improved compared to the first round, a larger-scale trial is still needed to obtain statistically significant results.

In/correct recognition of PURO categories As a part of the experiment, we also looked at the detailed results through the prism of recognition of adequate PURO terms, thus aligning it to some degree with the 'labeling' approach in Evaluation 1 (esp. the confusion matrix of the students and expert labels), although now the evaluation is tougher due to intricate structure of the graphically authored models. We made the following main observations (related to the presence/absence of Level 3 errors) within the 24 PURO models delivered (15 for the 'offer' task and 9 for the 'book' task):

- In 6 cases the student modeled one or more instantiations, e.g., assignment of a computer to its model "Bell Longitude E1234", as a valuation (although it is not of quantitative nature).
- In 3 cases, analogously, the student modeled one or more facts, e.g., that of a book having "fantasy" as genre,[40] as a valuation.

[40] We preferred modeling this situation as a fact rather than an instantiation in PURO. While 'fantasy book' could also be a type of books, modeling of genres as $\mathcal{B}$-objects allows to reuse the same genre for other types of artifacts such as movies or clothes. In any case, however, valuation is not an adequate choice.

- In 6 cases, in turn, the student modeled one or more valuations as facts, e.g., the assignment of RAM capacity[41] to a computer or of age to a person.
- In 2 cases the student modeled one or more instantiations as facts, e.g., again, the model of the computer was modeled as a $\mathcal{B}$-object linked to the computer by a $\mathcal{B}$-relationship such as "name".
- In 5 cases the students ignored the existence of a $\mathcal{B}$-type such as "Bell Longitude E1234" entirely and only put it into the name of the particular $\mathcal{B}$-object.

In the 'laptop' task the students also generally failed to identify the presence of the MISO pattern (which was not present in the PURO tutorial they had been given). Consequently, they (in 10 cases out of 15) assigned the price to the laptop model itself rather than to its particular offer.

Influence of users' experience To illustrate the users' experience in ontology engineering and its possible influence on the experiment results, we compared the students' results in the experiment with their results in written tests in the linked data course during which the experiment took place. The tests included three practical assignments dealing with lightweight ontology design and RDFS reasoning. The average test score of the 15 students who achieved complete coverage when modeling in PURO was 22%, while those 7 who failed in covering the model in PURO had an average test score of 17%.[42] The 9 students successfully covering the model when using Protégé had an average test score of 33%, while the average test score of those (7) failing in this task was only 6%. We also used the Spearman's rank correlation coefficient. The correlation coefficient between the students' test scores and their success in achieving complete model coverage in PURO (represented by a binary value) was approx. 0.08. The correlation between the test scores and success in coverage when using Protégé was about 0.47. When comparing the students' test scores and their success in creating a usable ontology skeleton, the values are very similar. Although none of these results are statistically significant, they suggest that users with more knowledge about ontology engineering are better at using Protégé (which is quite an expected result), while their success in PURO modeling (at least, as regards the model coverage) does not clearly depend on this knowledge.

SUS Questionnaire The questionnaire consists of 10 questions about how hard it was to learn and use the tool and whether its interface is consistent. The users choose answers from the Likert scale ranging from "strongly disagree" to "strongly agree." An overall score is computed from the answers resulting in a number between 0 and 100. The higher the score, the more usable the tool appeared to the user.

The SUS score computed from the answers about PURO Modeler ranged from 22.5 to 97.5 with an average of 72.8 and median 75. Protégé scores ranged from 27.5 to 77.5 with an average of 52.1 and 55 median. Given the Mann-Whitney U test p-value of 0.00054 being lower than 0.05, we consider the results significant. As some of the students answered the questionnaire both about PURO Modeler and Protégé, the two groups of scores are not completely independent. As the Mann-Whitney U test expects the groups to be independent, we also computed the Wilcoxon Signed-Ranks test, which is designed for correlated samples. For that, we used the answers of 8 students who completed both PURO Modeler and Protégé questionnaire. The resulting W-value from the test is 1.5, confirming the result is significant at $p < 0.05$. To express the difference in SUS scores more exactly, we computed the effect size (of using PURO Modeler instead of Protégé) using the Cohen's d method. The resulting d is approximately 1.28, meaning the average SUS score of PURO Modeler is 28% of pooled standard deviation higher than the score of Protégé.

This suggests PURO Modeler was found much more usable than Protégé for the task of creating an ontology skeleton. The students could have been biased by the fact that PURO Modeler is being developed at our university, however, we tried not to stress that fact too much and encouraged the students to remain objective. Another reason of course might be that while PURO Modeler is designed exactly for the task we evaluated, Protégé offers much more functionality and is thus more complex and harder to learn. That is however exactly one of the reasons why we propose the use of PURO for OWL ontology boostrapping: the users can first focus on the task of creating the skeleton or core model defining the concepts and relations in a simple user interface and proceed to adding other OWL constructs to the ontology in a complex tool like Protégé later.

[41] Modeling RAM as a $\mathcal{B}$-object would be fine. However, its capacity should be a mere value.

[42] The counts do not sum up to the total number of students since some did not complete either the experimental task or the course test.

Table 10
Correctness and coverage of resulting models by the first group of users in the 2nd round of Evaluation 2

	PURO offer						OWL books				
		Error level					Error level				
User	Technical error	1	2	3	Covers all	Usable ontology	1	2	3	Covers all	Usable ontology
1				×	✓	✓				✓	✓
2				×	✓	✓	N/A	N/A	N/A	N/A	N/A
3	×	×	×	×	✓		N/A	N/A	N/A	N/A	N/A
4	×			×	✓		N/A	N/A	N/A	N/A	N/A
5			×	×	✓	✓	N/A	N/A	N/A	N/A	N/A
6			×		✓	✓				✓	✓
7		×	×	×			N/A	N/A	N/A	N/A	N/A
8			×						×	✓	✓
9			×						×	✓	✓
10			×					×	×	✓	
11				×	✓	✓	N/A	N/A	N/A	N/A	N/A
12	×	×	×		✓	✓		×			
13			×	×							
14	×			×	✓	✓			×	✓	✓
15				×	✓	✓			×		

Table 11
Correctness and coverage of resulting models by the second group of users in the 2nd round of Evaluation 2

	PURO books						OWL offer				
		Error level					Error level				
User	Technical error	1	2	3	Covers all	Usable ontology	1	2	3	Covers all	Usable ontology
16		×	×		✓		N/A	N/A	N/A	N/A	N/A
17			×						×		
18			×	×			N/A	N/A	N/A	N/A	N/A
19			×					×	×		
20			×						×		
21			×		✓	✓	N/A	N/A	N/A	N/A	N/A
22		×			✓	✓			×	✓	✓
23			×		✓	✓			×	✓	✓
24				×	✓			×	×		
25		N/A	N/A	N/A	N/A	N/A			×	✓	✓
26		N/A	N/A	N/A	N/A	N/A		×	×		

6.2.5. Summary of Evaluation 2

Let us recall that the research question Q4 was: "Can minimally trained ontology engineers identify the type of PURO entity described in a free text, and compose the respective PURO entities into a complete PURO model?". Based on Evaluation 2, the answer to both parts of the question would be "partially", due to significant number of errors in the choice of PURO terms and occasionally missing entities (even if the completeness of the models seems to be slightly better than for OWL).

The rate of ontology usability, relevant for Q5, "Can the resulting PURO models be transformed into usable skeletons of OWL models, which have quality comparable to models directly authored in OWL based on the same described situation?", seems approximately the same for PURO and OWL, but, again, probably, insufficient. Also here, thus, the answer should be "partially".

7. Discussion

The previous parts of the paper attempted to demonstrate different aspects of PURO: Sections 2 and 3 presented its basic principles, including some degree of formal rigor; Sections 4 and 5 indicated its relevance to a large number of scenarios and use cases; finally, Section 6 tried to find out about its comprehensibility and usability (primarily) from the human point of view.

The principal design decision behind PURO is that of *unifying the two distinctions*, P-U and R-O(-V), in one representational language that is otherwise rather parsimonious, and also that of the primordial role of *alignments with OWL structures*. As a consequence of its parsimony, PURO is both poor in logical expressiveness (compared to OWL with its repertory of TBox axiom types) and shallow in its ability to grasp ontological distinctions of the real world (compared to methods inspired by applied ontology research). Yet, we believe that this parsimony contributes to making PURO a meaningful modeling tool, for three main reasons. First, it may be relatively easy and intuitive to read the graphically expressed models, and even to write them provided the author has the conceptualization well set up in his/her mind. Second, compared to an OWL ABox, PURO is prone to be more parsimonious (unless the OWL model implements shortcuts). It may also yield a better verbalization.[43] Third, the space of alignment/transformation patterns, either to OWL (where PURO models serve as OBMs), or possibly from richer languages such as OntoUML (where PURO models would serve as OFMs), does not explode.

Yet, despite the small number and apparent simplicity of PURO primitives, making plausible choices among them based on verbal description is not always obvious for newcomers, as the experiments indicate. For some people 'ontologistic' thinking is more natural while others hesitate even for very simple choices. Education of PURO *by examples* thus seems to be the most important means of its spreading. Since examples are at the very heart of the authoring style supported by PURO Modeler, this approach is also quite natural. By informal observation, of the scenarios explored within the use cases (Section 5), those involving *explicit*, graphically expressed PURO models seem more end-user accessible than those only involving isolated labels.

Notably, concerning all kind of generalization from the PURO evaluations from Section 6, we are aware that their representativeness is rather limited and they should thus be only considered as preliminary. Although the number of label assignments in Evaluation 1 and of individual modeling decisions in Evaluation 2 are in the order of several hundreds, they are severely interdependent due to having been made by a limited set of people with similar background and for a small set of items/tasks; the number of independent observations for a phenomenon is thus typically just a few. This is a problem hard to overcome in short term, since volunteers willing to undertake a relatively advanced ontology analysis/synthesis exercise are rare, and the space of patterns and challenges to be explored is huge. A possible workaround could perhaps be the use of properly *verbalized* versions of PURO models and OWL ontologies.

8. Related research

Several works have addressed the duality of a surface model (a kind of OFM), typically suffering from inadequacies or intentionally simplified for pragmatic purposes, and a more principled model (a kind of OBM) covering deeper distinctions of the same modeled domain/situation. The dimension according to which the

[43] This is an aspect and use case of PURO we have not yet explored at all, except that the second evaluation from Section 6.2 featured PURO model design based on a verbal description, i.e. the opposite, and manual, process.

second model was 'deeper' however varied, and so did the roles of both kinds of models in the overall ontology engineering (or related) scenarios.

We currently view the related research as divided, although not exclusively, into seven areas. The first area are *KR languages and formalisms* that share some aspects with PURO as if it were a standalone KR language (i.e., not considering the mentioned duality). As a second area we discuss *foundational ontologies*, which, unlike in the OBM approach, tackle the mentioned duality in the same representational space (shared with the OFM), even if OBM languages can also be derived from them. The third area is *multi-level modeling*, which addresses what we call the 'P-U distinction'; of this broad topic, we however confine ourselves to the Multi-Layer Theory (MLT), which tackles the topic in the practical context of ontologies. The fourth area is the foundational treatment of *relationships* (and the 'R-O distinction' as we call it); again, we confine the comparison to one recent work that summarizes many previous ones. The fifth area covers research projects that explicitly distinguish a kind of *OFM and OBM* (although not called this way and not necessarily fulfilling all the conditions from Def. 1 and 2 of this paper) such that an existing OFM (in OWL, or even another language) is augmented with background distinctions corresponding to the OBM, similarly as we address in the use cases 4–9 in Section 5; in simple terms, 'OFM to OBM' approaches. Analogously, the sixth area is aimed at developing an OFM on the basis of a kind of OBM, similarly as we address in the use cases 1–3 in Section 5; in simple terms, 'OBM to OFM' approaches. Finally, as the seventh area, we discuss OFM development from a different kind of model in general, including methods where the initial model does not need to be an OBM; the commonality with PURO is however in the possibility of generating different alternative OFMs.

It should be noted that PURO and its computational 'eco-system' lends itself to both the 'OFM to OBM' and 'OBM to OFM' scenarios (with the specific assumption of the OFM being expressed in OWL), and to a limited degree even to being used as a standalone graphical KR language. This versatility, witnessed by the span of the outlined use cases (and only shared by a fraction of the related research) could be viewed as a particular added value.

We do not devote an explicit subsection to the topic of *modeling patterns* in OWL, to avoid making the section excessively long. Some relevant patterns per se have been already discussed in Sections 3 and 5; use of patterns in particular approaches is also mentioned throughout the current section.

8.1. KR languages and formalisms comparable to PURO

As we stated at the beginning, the motivation for developing PURO was *not* to compete with existing standalone KR languages; the representational power of PURO is intentionally designed with mapping to OWL (and existing OWL users) in mind. Therefore we only make this comparison very brief and do not attempt to claim an added value of PURO as a standalone language with respect to other (often more powerful) languages.

As discussed in the introduction, OWL, even in its version 2 [62,12], lacks a number of expressive features needed to capture PURO models, notably n-ary relationships and higher-order features (such as metaclasses). While OWL 2 introduced *punning*, which allows us to reuse the same IRI both in the context of an *individual* and a *class*, this is not yet satisfactory, since they remain semantically decoupled into two independent entities. This is similar to Motik's contextual higher-order semantics [57]. It ought to be noted that RDF and its ontological extension RDFS, which form the 'substrate' for OWL, naturally support higher-orders, though they lack the expressiveness needed to capture constraints and richer models. OWL's *Full* semantics [56], which keeps these higher-order features, is known to be undecidable [57].

OWL's decidable semantics is derived from *description logics*, particularly from $\mathcal{SROIQ}$ [46]. Classical DLs do not allow for higher-order features; however, recently researchers have been working on their various higher-order extensions. Namely, Motik [57] and De Giacomo et al. [29] proposed a semantics for higher-order DL that maintains decidability. Similarly Pan and Horrocks proposed an alternative semantics for OWL and RDFS in order to take metamodeling into the framework while maintaining decidability. It is remarkable that Pan et al. proposed a variant of RDFS semantics, dubbed RDFS(FA) [45], later extended into OWL(FA) [63], in which types are divided into multiple strata, analogously to what we use in the PURO model when assigning labels $\mathcal{B}$-type-1, $\mathcal{B}$-type-2, etc. (as required to detect some forms of incoherence in the use cases 7–9).

Further, inspired by the representation needs of PURO, Homola et al. [44] proposed $\mathcal{TH}(\mathcal{SROIQ})$ a typed higher-order variant of $\mathcal{SROIQ}$, including

strictly-typed (i.e., homogeneous, in the sense of PURO) higher-order meta classes and meta properties. In a follow-up work Kubincová et al. [51] extended this language to $\mathcal{HIR}(\mathcal{SROIQ})$ which in addition enables to express modeling constraints over the instantiation relation. An interesting nuance of these works is that the languages can be translated back to classical $\mathcal{SROIQ}$, thus on the one hand pointing out that this DL is already more expressive than originally thought, and on the other hand allowing to employ an existing reasoning infrastructure for inference.

Motz et al.[59] proposed a different approach by developing the description logics $\mathcal{ALCQM}$ and $\mathcal{SHIQM}$ featuring a new logical operator $=_{\mathsf{m}}$ that enables to equate individuals and classes. This operator supports a modeling use case in which an entity is treated as an individual as long as it is not known to have instances, and it is equated with a class once needed. Their logics employ a stronger Henkin-style semantics [42] for higher orders. A similar extension of OWL was proposed by Motz [58].

Further higher-order extensions of DLs concentrated on answering meta-queries, namely the works of Lenzerini et al. [52,53], Cima et al. [10,11], Gu [33], and Gu and Zhang [34].

All the languages discussed so far, however, still lack the ability of directly capturing n-ary relationships. Note that in the past there have also been proposals [9,54] for DLs extended with n-ary relationships, especially in context of their applications in databases, and none of them considering higher orders.

When shifting to other kinds of logic-based KR beyond DL, a well-known formalism having both a textual and graphical syntax are Sowa's *conceptual graphs* (CG) [70], with grounding in Common Logic. Both CG and PURO support n-ary relationships and nested assertions (in PURO, as relationships participating in other relationships). From the usage point of view, PURO differs from CG in its explicit coverage of type hierarchy modeling and meta-typing and in its distinction of valuations. The graphical display of PURO in PURO Modeler also exhibits closer affinity to a natural language representation (such as flipping or removing the edge arrows, or removing the role labels when subsumed by the relation label). PURO, on the other hand, lacks some of the CG 'TBox' expressiveness, of which it currently only keeps subclassing and disjointness, and only combines facts via conjunction (i.e. it has no negation or disjunction), similarly to early versions of conceptual (existential) graphs.

Among the more pragmatic KR representations connected to the need of sharing structured data over the web, we can mention Topic Maps[44] (TM) [28]. Similarly to PURO, TM allow to model n-ary relationships and endow their ends with roles. However, the system of roles is more complex in TM: they have their own instance level and type level, while the roles in PURO only occupy a single sort. TM also support type hierarchies, but do not allow for meta-typing, nor for nested assertions. TM's 'internal occurrences' to some degree correspond to PURO valuations, though they are not confined to quantitative values and thus do not guide the modeler towards the identification of 'under-cover' universals in qualitative 'values'.

The *UML* family of conceptual modeling languages, representing an industry standard, features a specific kind of diagrams, *object diagrams*,[45] for designing examples of class diagram instantiation. Class and object diagrams in UML can thus be used in a similar way as PURO models. However, the UML diagrams are very rudimentary: they do not allow for n-ary relationships, meta-typing nor roles.

Overall, the choice of PURO language primitives was formed by the desideratum of smooth transformation to OWL that would minimally alter the graph structure (if not forced by constraints of a particular application) and by the assumption that the TBox structures would better be provided later in OWL. This setting is unique compared to all the mentioned languages.

8.2. Foundational ontologies

Foundational ontologies (FO) are, obviously, closely related to the notion of OBMs as coined in this paper, since both are concerned with expressing deeper ontological distinctions in order to enhance the quality of more pragmatically or less competently developed (but more directly usable) models. An apparent distinction lays in the way of coupling the foundational/background model with the lightweight/foreground model which it aims to support (cf. Condition 3 in Def. 1). The FO concepts serve as root concepts of the OFM concepts, which allows for unified reasoning over the whole stack of models; the cons of this solution is that the combined model including the 'injected' FO becomes more complex, and the FO con-

[44] http://www.topicmaps.org/

[45] https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-object-diagram/

cepts may also be structurally distant from the more specific OFM concepts and thus less on sight. In contrast, the OBM entities reside in a different space than those of the OFM and are connected to them via an exogenous mapping (annotation and/or metamodeling). The OFM thus need not be structurally altered, and the OBM labels can be displayed where needed.

A prominent example of an OBM language and FO couple is that of *UFO* and OntoUML [37]. UFO provides, in several modules (for endurants, perdurants, and other), a careful axiomatization of a huge number of ontological distinctions. OntoUML then applies the same distinctions as stereotypes over elements of UML models.

Although a formal (OWL) ontology expressing the PURO distinctions has been developed for pragmatic purposes (OWL ontology metamodeling, coherence checking, and OWL ontology seed construction), there is no yet a proper, fully documented and philosophically anchored FO specifically developed with respect to PURO. However, the core PURO distinctions are rather basic from the FO viewpoint and could also be grounded on existing FO, since they appear, among other, in DOLCE[46], UFO, or BFO[47]). A more foundational-ontologistic exercise in this respect is left to future work, probably in connection with the design of optional extensions of the core PURO language model.

8.3. Multi-level modeling and MLT

Modeling entities in multiple levels of instantiation is a ubiquitous phenomenon, appearing not only in ontology engineering but also in process modeling, enterprise engineering or linguistics; this is witnessed by the scope of domains discussed at the recent Dagstuhl seminar on this topic [1]. However, we confine ourselves to the *Multi-Level Theory* (MLT),[48] which can serve as the ontological engineering proxy to this area of thinking, and particularly to its mapping to OWL.

Brasileiro et al. [7] are concerned with the inability of OWL vocabularies to express higher-order types and the general lack of consideration of principles related to relations between types in modelling. They point out to the previously developed Multi-Level Types (MLT) theory [14] and explore its potential to be used in connection with linked data vocabularies with the aim to formulate expressive constraints related to relations between types (possibly higher-order) and for automated verification of these constraints. They propose an OWL vocabulary for this purpose that can be combined with other ontologies and a set of associated constraints which are implemented as SPARQL queries which enables their automated verification.

Similarly to our work, Brasileiro et al. also focus on the problem of capturing certain relations between entities which stem from their ontological nature and which either cannot be represented or are modeled only approximately. However they predominantly focus on types and on capturing certain complex relations among them. In this respect their work is more detailed than ours. On the other hand, the focus of PURO is much broader, notably including the R–O distinction as well. Also, the general focus of our work lies in providing the non-expert authors of linked data vocabularies not only with a suitable framework that is easy to grasp and start using but as well with complementary tools and methodology. We assume that in our future works further exploring PURO background coherence ckecking, we should explicitly leverage on MLT and related results of Brasileiro et al.

As an example, consider the diagram from Fig. 4. We could apply the notion of *powertype* (known from prior works, but thoroughly elaborated in MLT), defined as 't is powertype of t' iff every instance of t is a specialization of t' and all specializations of t' are instances of t.'. This means, among other, that if *Release type* is derived (via an obvious lexical pattern) as the powertype of *Release* (a $\mathcal{B}$-type not present in the diagram for simplicity) then *Album* as instance of *Release type* would be inferred to be a $\mathcal{B}$-subtype of *Release*.

8.4. Ontological analysis of relation/ship/s

The notion of relation and relationship in often treated as intuitive in ontological engineering, though there are quite some intricacies, recently summarized by Guarino & Guizzardi [35]. They demonstrate that relationships should be treated as endurants and not as events (perdurants), which is coherent with our approach. Their analysis however goes much deeper than the treatment of relationships in PURO, which merely allows to make relationships participate in other relationships. On the other hand PURO also explicitly distinguishes quantitative valuations. The notion of relation is used differently by Guarino & Guizzardi; we merely use this term to denote types of relationships,

[46] http://www.loa.istc.cnr.it/DOLCE.html

[47] http://www.ifomis.org/bfo

[48] The most recent works term it MLT* and present a textual language, ML2, that fully implements it [25].

while in their approach 'a relation holds while a relationships exists', both in connection to the same state of affairs (presumably, at the level of particulars). They also reject the interpretation of relationships as facts, a major objection being the multitude of facts being expressible based on the same core relationship; e.g., one fact may state that a person studies at a university, another one that she studies there with certain grading or number of credits, etc. In PURO, 'fact' is a category of relationship, which admittedly brings PURO close to the surface, data-centric approach of OWL. Yet, we allow for a variable number of participants and their roles for different relationships instantiating the same relation; we therefore grasp the core relation(ship) at the level of the universal entity, being roughly compatible with Guarino & Guizzardi, though with using a shifted terminology. Further subtle issues entailed by their analysis could possibly be added to PURO in the future as optional refinements to its simple metamodel.

8.5. From OFM to OBM

As we discussed in Section 4 for the particular case of PURO, approaches aiming to proceed from an existing OFM to an OBM can either 1) *annotate* the OFM with the OBML distinctions, 2) *metamodel* the OFM inside the OBM metamodel, or 3) *transform* the OFM to an OBM (which can be viewed as a kind of reengineering). Different ontology management projects incorporated one or more of these techniques.

A prominent example of OBM distinctions being applied to existing concept taxonomies as *annotations* is the OntoClean method [36], designed independently, but later naturally applied to OWL [85]. While OntoClean focuses on annotating classes only and on repairing their taxonomic relationships, PURO primarily examines instantiations and facts, and aims to map OFMs to OBMs not necessarily for repair purposes. OntoClean is concept-oriented (in contrast to PURO's fact-orientation), and requires the engineer to take into account OBML notions such as rigidity that are fundamentally different from what s/he knows from the OFM representation in OWL.

Seyed [68] suggests an OntoClean-based OBML with finer categories of entities by which classes are annotated; however, relationships are not considered, the main focus of the approach being the rigidity metaproperty.

When assigning the OntoClean meta-properties to an OWL ontology, Welty [85] applied an additional OWL-DL ontology that contained the background information in the form of metaknowledge. The ontology represented the basic OntoClean metaproperties and constraints; an instance view of the OWL class hierarchy was then *metamodeled*. A similar kind of metamodeling, although with less OBM flavor (since the meta-knowledge was rather intended to express technical aspects such as provenance or access rights), and with possibly multiple meta-ontologies, was proposed by Tran [82]. Furthermore, Glimm et al. [30] extended such an approach by introducing a technique that enabled class-based metamodeling within one ontology – meta-level constraints on classes and their subsumption relationships are expressed as OWL axioms in the same ontology as the actual content. Inconsistencies in base ontology design then can be localized by comparably cheap instance retrieval operations. The approach we took in the coherence checking use case is similar: we metamodel the foreground ontology and then express constraints that are necessary to check the coherence w.r.t. the PURO model over the metamodeled ontology. We are then able to detect incoherence using a regular ontology reasoner. In this sense our approach is closer to the one of Welty, as we require the preprocessing step in which the deductive closure of the foreground ontology is computed and then the meta-level reasoning is performed independently. In the future we would like to investigate on the possibility to perform both levels of reasoning at the same time, similarly to Glimm et al.

Gangemi [27] uses Descriptions and Situations (DnS), an expressive ontology pattern, to express alternative conceptualizations of a same set of facts. It distinguishes the 'ground' description represented as a *situation* from the 'duper' description represented in DnS as a *description*, which provides an alternative conceptualization or schema. This part of DnS uses (reified) relations, concepts, and the relations between them and also provides a description of the relations between them and ground entities. The typical scenario of this inventory is that of *reengineering* (the duper description from the ground one).

The recent research (by some of the authors of this paper) on *focused categorization power* of ontologies [76] also shares some aspects with OBM reengineering. The computation of focused categorization power aims to approximate the usability of an OWL ontology to assign individuals already known to belong to a named class (called focus class) to finer-grained categories. Such categories can be constructed from patterns occurring in the ontology structure. For example, if the ontology contains a property p whose domain

is a subclass of the focus class FC and its range is a class D having an instance i, the concept expression $\exists p.\{i\}$ may be, with some probability, a plausible subcategory of FC. Since it has been demonstrated that in many cases such an individual i metamodels a class, it corresponds to a 'non-trivially manifested' $\mathcal{B}$-type in PURO terms.

Replacing OWL with UML as the OFM language, we should also recall the previously mentioned OntoUML [37] here. OntoUML adds the background distinctions to UML entities through the mechanism of UML stereotypes. Subsequently, rigidity and other constraints similarly as in the OntoClean method.

8.6. From OBM to OFM

Referring again to Section 4, proceeding from an OBM to an OFM may have the form of *transforming* the OBM to a new OFM or of *aligning* it with existing OFMs.

While the early best practices of ontological engineering [32] recommended creating the first formal draft of an ontology in a more flexible language such as first-order logic, in reality most ontologies of the semantic web era seem to be first formalized in OWL proper, only relying on prior textual glossaries and semi-informal diagrams. The negative impact of such direct coding has however been recognized, and several researchers then attempted to build 'pipelines' allowing to transform richer or more flexible KR languages to OWL. Note however that the unique feature of PURO is, however, its backwards design direction: from OWL proper to a language relaxing its specific obstacles to faithful modeling.

Sunagawa [73] presented a fact-oriented approach that maps the background notion of 'role' onto OWL. Role concepts are exemplified, among other, as those of food or fuel: an object can become and then cease to be food or fuel during its lifetime. Roles are considered as essential for ontology development, however it is difficult to represent them in OWL properly. Several patterns are provided for this purpose.

The most similar to our approach is probably the research carried out in the context of the OntoUML project. Several implementations of OntoUML-to-OWL transformation exist. Probably the most advanced was done by Barcelos et al. [2]. It focuses on capturing all constraints modeled in OntoUML. To achieve that, SWRL[49] rules are produced where OWL constructs are insufficient.

There is a graphical editor for OntoUML, formerly called OLED [5], recently renamed to Menthor.[50] Apart from modeling and consistency checking, it offers transformation of the model into various languages, including the above-mentioned transformation to OWL. The support of checking the model for errors is quite advanced. The tool not only detects syntactic errors, but can also (semi-)automatically detect semantic errors. Two methods are implemented for the latter. The first is detection of anti-patterns. Anti-patterns are defined as "recurrent error-prone modeling decisions." Simply put, they are model structures that are syntactically correct, but usually have different meaning than the one intended by the engineer. The tool has a database of such 'typical errors.' The second method is called 'visual simulation' and consists of generation of instances of the model. The idea is that the user will be able to see errors in the conceptual model by checking its automatically created example instantiations.

PURO Modeler currently does not offer error detection based on anti-patterns. However, PURO models include instance-level entities, or more specifically, the modeling should start from the instances. That might lead to a similar result as the 'visual simulation,' except that the user create the instances manually instead of having them generated automatically.

One of the main purposes of PURO models in the OBM-to-OFM scenario is to allow generation of OFMs in various encoding styles. Menthor also supports various OntoUML-to-OWL transformations, resulting in different OWL styles. However, this option is so far only focused on different ways of modeling temporally changing information in OWL [38] and it is applied on the whole model: the user cannot choose a different transformation method for a specific part of the model. On the other hand, the temporal aspect of modeled entities, i.e., supporting the appropriate way of modeling entities and relationships that can change in time, was not yet taken into account in PURO-to-OWL transformation and is left to be dealt with by the user.

To sum up, we could say that OntoUML and PURO have a slightly different aim in terms of the OBM-to-OFM scenario. While PURO serves for capturing the common background from which various different OFMs can be generated, OntoUML allows creat-

[49] https://www.w3.org/Submission/SWRL/

[50] http://www.menthor.net/ontouml.html

ing a detailed, complex model which can be thoroughly checked before it is transformed to OWL.

We are unaware of prior research addressing the *alignment* from an OBM to an OFM. The *alignment patterns*, although structurally similar, have a different purpose: to increase the power of automatic matchers that attempt to heuristically find an alignment between entities from different ontologies from the same domain. While the heterogeneity of the encodings employed might confuse simple, e.g., string-similarity-based matchers, alignment patterns can bridge between them. The binary character of the typical ontology alignment tasks implies that the alignment patterns are biased towards pairwise interconnection of the encodings. They are normally expressed in terms of template TBox axioms (containing placeholders). The most comprehensive alignment pattern catalog are probably those by Scharffe [67] and [24].

Very similar to alignment patterns are OWL-to-OWL *transformation patterns*. A representative of approaches employing such patterns is the PatOMat project [74], as part of our previous research. Its various use cases, including, in particular, ontology adaptation to a best-practice pattern/ontology imported to it or to the requirements of a reasoner, are summarized in [75]. The representation of transformation patterns in PatOMat is however less declarative than that of alignment patterns. Both the alignment and transformation patterns also suffer from maintenance issues due to the potentially high number of pairwise combinations, which limits their usability for the generic scenario of ontology development.

Aside the mentioned projects that exhibit some kind of true 'OBM vs. OFM' representational duality, we could also recall modeling approaches that include both a complex and *shortcut* way of modeling within the same representation space. We already discussed such approaches as well as the relationship to PURO in Section 3.9: prominent examples are the modeling of the performance–signal relationship in the Music Ontology or the pattern for LD publishing by Krisnadhi et al. [50].

8.7. Indirect OFM creation in general

Proceeding from OBM to OFM in order to create an ontology may be thought of as a special category of what could be called indirect ontology creation methods. There are several proposals of approaches where engineers first create a model in some different language, which does not necessarily need to be an OBM, and generate an ontology (or an ontology skeleton) from it automatically. We can differ between four kinds of non-OWL models for indirect methods known to us: (1) controlled natural language, (2) simplified structured textual notation, (3) graphical non-OWL models and (4) table templates.

Examples of controlled natural language approaches are the tool GINO by Bernstein and Kaufmann [6] or ROO by Dimitrova et al. [15]. A simplified notation for writing structured text transformable to OWL was developed by Kalyanpur et al. [48]. Examples of graphical non-OWL models are conceptual models in OntoUML, concept maps in CmapTools COE (Collaborative Ontology Environment) [41] and ontological background models in PURO. The table template approach, implemented, e.g, in the tool Webulous [47], is based on the idea where ontology engineer prepares table templates with, e.g., columns for class names and predefined relationships to other classes. Domain experts then just fill-in rows of the table, without the need to understand much of OWL, which is then automatically generated from the table. This approach is usually suitable for large ontologies with repetitive structure that can be represented by the table.

One of the advantages of indirect methods is the possibility of adding an additional step: configuring the transformation to OWL or choosing from several alternative transformation results in order to obtain an ontology seed in the desired *encoding style*. On the other hand, there is often a disadvantage: the initial non-OWL model might not contain all the desired definitions due to simplifications and the result of its transformation to OWL might be a mere ontology skeleton that needs to be further edited, finalized, using some of the direct methods.

Most indirect methods target inexperienced users by hiding some of the OWL complexity or showing it in a possibly more easily understandable way. Such motivation is explicitly stated, e.g., in the cases of GINO or the work by Kalyanpur et al. [48]. Our proposal of using PURO-to-OWL transformation to create ontologies to some extent share this motivation. To what extent it actually helps casual users is however yet to be found out.

To make this section more complete, we should mention several other approaches of non-OWL model transformation to OWL.

Bauman [3] implemented an XSLT-based transformation method of conceptual models into XML Schema, while OWL as target is only mentioned as possible future work. The user can choose a sort of en-

coding style, e.g., whether to transform a concept to an XML attribute or to a child-element.

To allow reusing existing ER diagrams, [23] designed their rule-based transformation to OWL ontologies. The framework is however not intended as a general ontology development alternative.

For transformation between different types of models such as UML, XML Schema or OWL, Kensche et al. [49] suggested to employ a generic metamodel (GeRoMe) as an abstraction of particular metamodels. In order to uniformly capture specific properties of models of different types, elements of GeRoMe are decorated with a set of role objects (e.g., a role attribute is mapped to a column in a relational schema and to a data property in OWL DL). Native models can be imported/exported into/from GeRoMe.

In all mentioned OWL generation methods, the input model is created at the level of types. In our approach, in contrast, the input model is created as an example situation at the instance level (however, including also the types of modeled instances). The second main difference between the described methods and our approach is the support of OWL encoding styles. In the existing approaches the transformation from the initial model to OWL is hard-coded, while in our approach the user can choose the encoding style for each entity in the PURO model.

9. Conclusion and future work

PURO is primarily a language allowing to graphically draft ontology skeletons without some of OWL biases, yet in a formally well-grounded language, and to derive multiple OWL ontology versions ('encodings') from a single conceptualization. However, as the collection of use cases indicates, there is ample space of further opportunities to exploit this relatively simple language in the context of semantic web and linked data. While only one of the use cases has so far passed a proper evaluation phase, the mere span of the diverse use cases to which PURO appears well-fitting (in the proof-of-concept phase) could be a strong argument for its further improvement and promotion.

Ongoing work includes, in particular, more detailed user studies related to the process of authoring an ontology skeleton using PURO, including its comparison with alternative methods: not only mainstream editors such as Protégé, but also graph-based authoring environments, which feature similar ergonomics as PURO Modeler but conform to the OWL metamodel. The feedback from the completed experiments (in Section 6.2) allows to tune the new experimental setting towards relevant aspects of the comparison.

The opportunities for future work are abundant. While the current work as well as our previous papers and reports, and also the OBOWLMorph implementation, only feature ad hoc collections of PURO2OWL *transformation/alignment patterns*, we plan to conduct a systematic exploration of the space of such patterns, yielding a catalog to be both published and operationalized (in the form of an RDF knowledge base) for exploitation in OBOWLMorph or other tools.

We are also working on procedures for interactive *reengineering* of PURO models from existing OWL ontologies. While the same alignment patterns could be employed in a reverse fashion, the points of nondeterminism, requiring user interaction, appear different in PURO model reengineering than in their operationalization. An associated interesting opportunity could be the maintenance of the alignments during the evolution of an OWL ontology originally bootstrapped by operationalization from PURO. The subsequent changes could then feedback the original PURO model, thus completing a *round trip*.

In connection with our synchronous research on *focused categorization power* [76], already mentioned in Section 8.5, we would like to explore the impact of PURO annotation on calculation of this measure. For example, OWL individuals explicitly annotated as PURO $\mathcal{B}$-types would be more plausible constructors of categories than other individuals; similarly, categories defined using property chains would be more plausible if the intermediate object were annotated as a (reified) $\mathcal{B}$-relationship, analogously to the heuristic proposed in the CBD use case (Section 5.6).

On top of the presented core PURO language, *finer-grained distinctions* (to be optionally switched on in PURO authoring and management tools) will also be designed as an extra layer. Examples of such distinctions could be the declarations of: $\mathcal{B}$-types expressing (and $\mathcal{B}$-relations establishing) *roles* (fulfilled by agents or objects in a certain contexts); $\mathcal{B}$-types expressing *qualities*; $\mathcal{B}$-relationships and $\mathcal{B}$-valuations referring to the *extension vs. intension* of a $\mathcal{B}$-type (cf. Sec. 3.12); and many others, including those following from MLT [14] and foundational modeling of relationships proposed by Guarino & Guizzardi [35]. These would, among other, allow to more smoothly bridge between PURO and richer KR languages such as OntoUML [4]; PURO could then serve as a kind of proxy between OntoUML and OWL, possibly as a 'middle-out' ap-

proach to modeling starting from what 'ordinary users' perceive as faithful, unconstrained modeling of reality (analogously to middle-out construction of taxonomies starting from most commonly uttered notions). Some other distinctions could, on the other hand, emulate information expressible by TBox axioms in OWL; these could constrain the scope of possible transformations to OWL and also give rise to the respective OWL axioms. Presumably, the PURO model author could either proactively add the distinctions in the graphical environment, or they could be elicited through a dialog with a wizard (in a similar spirit as proposed by Seyed [68] for OntoClean) specifically for the cases where the transformation to OWL would otherwise be ambiguous (and there would not perhaps be enough lexical hints present).

Aside the logical and (possibly) philosophical aspects of modeling in PURO, the *lexico-linguistic aspect* might also be important. We plan to explore the space of *naming conventions* for PURO, striving at a balance between easy readability of the models both in the graphical format and after machine verbalization (which is a future topic of its own) and their efficient transformability to OWL entity IRIs and labels. The problem of 'naming transformation patterns' has already been tackled in our previous research on OWL2OWL transformation [84], from which we can partially draw inspiration.

Acknowledgments

Vojtěch Svátek, Miroslav Vacura and Marek Dudáš have been partially supported by CSF 18-23964S. Martin Homola and Ján Kľuka are supported by Slovak national project VEGA no. 1/0778/18. Further projects supported aspects of the early PURO-related research but ended prior to the writing period of this paper: bilateral project LAAOS supported by MŠMT under no. 7AMB12SK020 and by APVV under no. SK-CZ-0208-11, and projects ICT FP7-257943 (LOD2), IGS VŠE F4/34/2014, F4/90/2015, F4/28/2016, and VEGA 1/1333/12. The authors are also indebted to colleagues who contributed to PURO-related studies or co-developed relevant tools, particularly, to Tomáš Hanzal, Simone Serra and Ondřej Zamazal, as well as to Jiří Ivánek for comments to the formal portions of the paper.

References

[1] João Paulo A. Almeida, Ulrich Frank, and Thomas Kühne. Multi-Level Modelling (Dagstuhl Seminar 17492). *Dagstuhl Reports*, 7(12):18–49, 2018. ISSN 2192-5283. . URL `http://drops.dagstuhl.de/opus/volltexte/2018/8675`.

[2] Pedro Paulo F. Barcelos, Victor Amorim dos Santos, Freddy Brasileiro Silva, Maxwell E. Monteiro, and Anilton Salles Garcia. An automated transformation from OntoUML to OWL and SWRL. In Marcello Peixoto Bax, Mauricio Barcellos Almeida, and Renata Wassermann, editors, *ONTOBRAS*, volume 1041 of *CEUR Workshop Proceedings*, pages 130–141. CEUR-WS.org, 2013.

[3] Bruce Todd Bauman. Prying apart semantics and implementation. In *Balisage: The Markup Conference*, 2009.

[4] Alessander Botti Benevides and Giancarlo Guizzardi. A model-based tool for conceptual modeling and domain ontology engineering in OntoUML. In *Enterprise Information Systems*, pages 528–538. Springer, 2009.

[5] Alessander Botti Benevides and Giancarlo Guizzardi. A model-based tool for conceptual modeling and domain ontology engineering in OntoUML. *Enterprise Information Systems*, pages 528–538, 2009.

[6] Abraham Bernstein and Esther Kaufmann. Gino-a guided input natural language ontology editor. In *International Semantic Web Conference*, volume 2006, pages 144–157. Springer, 2006.

[7] Freddy Brasileiro, João Paulo A. Almeida, Victorio A. Carvalho, and Giancarlo Guizzardi. Expressive multi-level modeling for the semantic web. In Paul Groth, Elena Simperl, Alasdair Gray, Marta Sabou, Markus Krötzsch, Freddy Lecue, Fabian Flöck, and Yolanda Gil, editors, *The Semantic Web – ISWC 2016: 15th International Semantic Web Conference, Kobe, Japan, October 17–21, 2016, Proceedings, Part I*, pages 53–69. Springer International Publishing, Cham, 2016. ISBN 978-3-319-46523-4. . URL `http://dx.doi.org/10.1007/978-3-319-46523-4_4`.

[8] John Brooke et al. SUS-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[9] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Conjunctive query containment in description logics with n-ary relations. In Ronald J. Brachman, Francesco M. Donini, Enrico Franconi, Ian Horrocks, Alon Y. Levy, and Marie-Christine Rousset, editors, *Proceedings of the 1997 International Workshop on Description Logics, Université Paris-Sud, Centre d'Orsay, Laboratoire de Recherche en Informatique LRI, France*, volume 410 of *URA-CNRS*, 1997.

[10] Gianluca Cima, Giuseppe De Giacomo, Maurizio Lenzerini, and Antonella Poggi. On the SPARQL metamodeling semantics entailment regime for OWL 2 QL ontologies. In Rajendra Akerkar, Alfredo Cuzzocrea, Jannong Cao, and Mohand-Said Hacid, editors, *Proceedings of the 7th International Conference on Web Intelligence, Mining and Semantics, WIMS 2017, Amantea, Italy, June 19–22, 2017*, pages 10:1–6. ACM, 2017.

[11] Gianluca Cima, Giuseppe De Giacomo, Maurizio Lenzerini, and Antonella Poggi. Querying OWL 2 QL ontologies under the SPARQL metamodeling semantics entailment regime. In Sergio Flesca, Sergio Greco, Elio Masciari, and Domenico Saccà, editors, *Proceedings of the 25th Italian Symposium on Advanced Database Systems, Squillace Lido (Catanzaro), Italy, June 25–29, 2017.*, volume 2037 of *CEUR Workshop Proceedings*, page 165. CEUR-WS.org, 2017.

[12] Bernardo Cuenca Grau, Ian Horrocks, Boris Motik, Bijan Parsia, Peter Patel-Schneider, and Ulrike Sattler. OWL 2: The next step for OWL. *J. Web Semant.*, 6(4):309–322, 2008.

[13] Richard Cyganiak, David Wood, and Markus Lanthaler, editors. *RDF 1.1 Concepts and Abstract Syntax*. Recommendation. W3C, 25 Feb 2014. URL `http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/`.

[14] Victorio Albani de Carvalho and João Paulo A. Almeida. Toward a well-founded theory for multi-level conceptual modeling. *Software and System Modeling*, 17(1):205–231, 2018. .

[15] Vania Dimitrova, Ronald Denaux, Glen Hart, Catherine Dolbear, Ian Holt, and Anthony G Cohn. Involving domain experts in authoring OWL ontologies. In *International Semantic Web Conference*, pages 1–16. Springer, 2008.

[16] Marek Dudás, Ondrej Zamazal, and Vojtech Svátek. Exploiting ontology matching to support reuse in PURO-started ontology development. In Pavel Shvaiko, Jérôme Euzenat, Ernesto Jiménez-Ruiz, Michelle Cheatham, Oktie Hassanzadeh, and Ryutaro Ichise, editors, *OM@ISWC*, volume 1766 of *CEUR Workshop Proceedings*, pages 243–244. CEUR-WS.org, 2016.

[17] Marek Dudá'š, Tomáš Hanzal, and Vojtěch Svátek. What can the ontology describe? Visualizing local coverage in PURO modeler. In Valentina Ivanova, Tomi Kauppinen, Steffen Lohmann, Suvodeep Mazumdar, Catia Pesquita, and Kai Xu, editors, *VISUAL@EKAW*, volume 1299 of *CEUR Workshop Proceedings*, pages 28–33. CEUR-WS.org, 2014.

[18] Marek Dudáš, Tomáš Hanzal, Vojtěch Svátek, and Ondřej Zamazal. OBOWLMorph: Starting ontology development from PURO background models. In Valentina A. M. Tamma, Mauro Dragoni, Rafael Gonçalves, and Agnieszka Lawrynowicz, editors, *OWLED*, volume 9557 of *Lecture Notes in Computer Science*, pages 14–20. Springer, 2015. ISBN 978-3-319-33244-4.

[19] Marek Dudáš, Tomáš Hanzal, Vojtěch Svátek, and Ondřej Zamazal. OBM2OWL patterns: Spotlight on OWL modeling versatility. In Eva Blomqvist, Pascal Hitzler, Adila Krisnadhi, Tom Narock, and Monika Solanki, editors, *WOP*, volume 1461 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2015.

[20] Marek Dudáš, Vojtěch Svátek, and Jindřich Mynarz. Dataset summary visualization with LODSight. In Fabien Gandon, Christophe Guéret, Serena Villata, John G. Breslin, Catherine Faron-Zucker, and Antoine Zimmermann, editors, *ESWC (Satellite Events)*, volume 9341 of *Lecture Notes in Computer Science*, pages 36–40. Springer, 2015. ISBN 978-3-319-25638-2.

[21] Marek Dudáš, Vojtěch Svátek, Miroslav Vacura, and Ondřej Zamazal. Starting ontology development by visually modeling an example situation - a user study. In Valentina Ivanova, Patrick Lambrix, Steffen Lohmann, and Catia Pesquita, editors, *VOILA@ISWC*, volume 1704 of *CEUR Workshop Proceedings*, pages 114–119. CEUR-WS.org, 2016.

[22] Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

[23] Muhammad Fahad. ER2OWL: Generating OWL ontology from ER diagram. In *Intelligent Information Processing IV*, pages 28–37. Springer, 2008.

[24] Pablo Rubén Fillottrani and C. Maria Keet. Patterns for heterogeneous TBox mappings to bridge different modelling decisions. In Eva Blomqvist, Diana Maynard, Aldo Gangemi, Rinke Hoekstra, Pascal Hitzler, and Olaf Hartig, editors, *ESWC (1)*, volume 10249 of *Lecture Notes in Computer Science*, pages 371–386, 2017. ISBN 978-3-319-58068-5.

[25] Claudenir M. Fonseca, João Paulo A. Almeida, Giancarlo Guizzardi, and Victorio Albani de Carvalho. Multi-level conceptual modeling: From a formal theory to a well-founded language. In *Conceptual Modeling - 37th International Conference, ER 2018, Xi'an, China, October 22-25, 2018, Proceedings*, pages 409–423, 2018. .

[26] Aldo Gangemi. Ontology design patterns for semantic web content. In Yolanda Gil, Enrico Motta, V. Richard Benjamins, and Mark A. Musen, editors, *International Semantic Web Conference*, volume 3729 of *Lecture Notes in Computer Science*, pages 262–276. Springer, 2005. ISBN 3-540-29754-5.

[27] Aldo Gangemi. Super-duper schema: an OWL2+RIF dns pattern. In Vinay Chaudry, editor, *Proceedings of DeepKR Challenge Workshop at KCAP11*. 2011. URL `http://www.ai.sri.com/halo/public/dkrckcap2011/Gangemi.pdf`.

[28] Lars Marius Garshol and Graham Moore. ISO 13250-2: Topic Maps — Data Model. Final draft, ISO/IEC, 16. December 2005. URL `http://www.isotopicmaps.org/sam/sam-model/`.

[29] Giuseppe De Giacomo, Maurizio Lenzerini, and Riccardo Rosati. On higher-order description logics. In *Proceedings of the 2009 International Workshop on Description Logic (DL 2009)*, volume 477 of *CEUR Workshop Proceedings*, Oxford, United Kindgom, 2009.

[30] Birte Glimm, Sebastian Rudolph, and Johanna Völker. *Integrated Metamodeling and Diagnosis in OWL 2*, pages 257–272. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-17746-0. . URL `https://doi.org/10.1007/978-3-642-17746-0_17`.

[31] Birte Glimm, Aidan Hogan, Markus Krötzsch, and Axel Polleres. OWL: Yet to arrive on the web of data? In Christian Bizer, Tom Heath, Tim Berners-Lee, and Michael Hausenblas, editors, *LDOW*, volume 937 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2012.

[32] Asunción Gomez-Perez, Mariano Fernandez-Lopez, and Oscar Corcho, editors. *Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web (2nd edition)*. Springer-Verlag, 2007. ISBN 1846283965.

[33] Zhenzhen Gu. Meta-modeling extension of Horn-SROIQ and query answering. In Maurizio Lenzerini and Rafael Peñaloza, editors, *Proceedings of the 29th International Workshop on Description Logics, Cape Town, South Africa, April 22–25, 2016.*, volume 1577 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2016.

[34] Zhenzhen Gu and Songmao Zhang. The more irresistible Hi(SRIQ) for meta-modeling and meta-query answering. *Frontiers of Computer Science*, 12(5):1029–1031, 2018.

[35] Nicola Guarino and Giancarlo Guizzardi. "We need to discuss the relationship": Revisiting relationships as modeling constructs. In Jelena Zdravkovic, Marite Kirikova, and Paul Johannesson, editors, *CAiSE*, volume 9097 of *Lecture Notes in Computer Science*, pages 279–294. Springer, 2015. ISBN 978-3-319-19068-6.

[36] Nicola Guarino and Christopher A. Welty. An overview of OntoClean. In Steffen Staab and Rudi Studer, editors, *Handbook on Ontologies*, pages 201–220. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. ISBN 978-3-540-92673-3. . URL `http://dx.doi.org/10.1007/978-3-540-92673-3_9`.

[37] Giancarlo Guizzardi. *Ontological Foundations for Structural Conceptual Models*. Number 15 in Telematica Institute Fun-

damental Research Series. Telematica Instituut, Enschede, The Netherlands, 2005. URL http://www.researchgate.net/publication/215697579_Ontological_Foundations_for_Structural_Conceptual_Models.

[38] Giancarlo Guizzardi and Veruska Zamborlini. Using a trope-based foundational ontology for bridging different areas of concern in ontology-driven conceptual modeling. *Science of Computer Programming*, 96:417–443, 2014.

[39] Tomáš Hanzal, Vojtěch Svátek, and Miroslav Vacura. Event categories on the semantic web and their relationship/object distinction. In Roberta Ferrario and Werner Kuhn, editors, *FOIS*, volume 283 of *Frontiers in Artificial Intelligence and Applications*, pages 183–196. IOS Press, 2016. ISBN 978-1-61499-660-6.

[40] Frank W Hartel, Sherri de Coronado, Robert Dionne, Gilberto Fragoso, and Jennifer Golbeck. Modeling a description logic vocabulary for cancer research. *Journal of biomedical informatics*, 38(2):114–129, 2005.

[41] Pat Hayes, Thomas C Eskridge, Mala Mehrotra, Dmitri Bobrovnikoff, Thomas Reichherzer, and Raul Saavedra. COE: Tools for collaborative ontology development and reuse. In *Knowledge Capture Conference (K-CAP)*, 2005.

[42] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.

[43] Martin Hepp. GoodRelations: An ontology for describing products and services offers on the web. In *Knowledge Engineering: Practice and Patterns, 16th International Conference, EKAW 2008, Acitrezza, Italy, September 29–October 2, 2008. Proceedings*, pages 329–346, 2008.

[44] Martin Homola, Ján Kľuka, Vojtěch Svátek, and Miroslav Vacura. Towards typed higher-order description logics. In Thomas Eiter, Birte Glimm, Yevgeny Kazakov, and Markus Krötzsch, editors, *Informal Proceedings of the 26th International Workshop on Description Logics, Ulm, Germany, July 23–26, 2013*, volume 1014 of *CEUR WS*, pages 221–233, 2013.

[45] Ian Horrocks and Jeff Z. Pan. RDFS(FA): Connecting RDF(S) and OWL DL. *IEEE Transactions on Knowledge & Data Engineering*, 19:192–206, 2007. ISSN 1041-4347. .

[46] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible $\mathcal{SROIQ}$. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning, Lake District of the United Kingdom, June 2–5, 2006*, pages 57–67. AAAI Press, 2006.

[47] Simon Jupp, Tony Burdett, Danielle Welter, Sirarat Sarntivijai, Helen Parkinson, and James Malone. Webulous and the Webulous Google add-on –a web service and application for ontology building from templates. *Journal of biomedical semantics*, 7(1):17, 2016.

[48] Aditya Kalyanpur, Nada Hashmi, Jennifer Golbeck, and Bijan Parsia. Lifecycle of a casual web ontology development process. In *WWW Workshop on Application Design, Development and Implementation Issues in the Semantic Web*, 2004.

[49] David Kensche, Christoph Quix, Mohamed Amine Chatti, and Matthias Jarke. Gerome: A generic role based metamodel for model management. In *Journal on data semantics VIII*, pages 82–117. Springer, 2007.

[50] Adila Krisnadhi, Nazifa Karima, Pascal Hitzler, Reihaneh Amini, Víctor Rodríguez-Doncel, and Krzysztof Janowicz. Ontology design patterns for linked data publishing. In Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, and Valentina Presutti, editors, *Ontology Engineering with Ontology Design Patterns*, volume 25 of *Studies on the Semantic Web*, pages 201–232. IOS Press, 2016.

[51] Petra Kubincová, Ján Kľuka, and Martin Homola. Expressive description logic with instantiation metamodelling. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25–29, 2016*, pages 569–572. AAAI Press, 2016.

[52] Maurizio Lenzerini, Lorenzo Lepore, and Antonella Poggi. A higher-order semantics for metaquerying in OWL 2 QL. In Chitta Baral, James P. Delgrande, and Frank Wolter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Fifteenth International Conference, KR 2016, Cape Town, South Africa, April 25–29, 2016.*, pages 577–580. AAAI Press, 2016.

[53] Maurizio Lenzerini, Lorenzo Lepore, and Antonella Poggi. Answering metaqueries over hi (OWL 2 QL) ontologies. In Subbarao Kambhampati, editor, *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016*, pages 1174–1180. AAAI Press, 2016.

[54] Carsten Lutz, Ulrike Sattler, and Stephan Tobies. A suggestion for an n-ary description logic. In Patrick Lambrix, Alexander Borgida, Maurizio Lenzerini, Ralf Möller, and Peter F. Patel-Schneider, editors, *Proceedings of the 1999 International Workshop on Description Logics (DL'99), Linköping, Sweden, July 30–August 1, 1999*, volume 22 of *CEUR Workshop Proceedings*. CEUR-WS.org, 1999.

[55] Viviana Mascardi, Valentina Cordì, and Paolo Rosso. A comparison of upper ontologies. In Matteo Baldoni, Antonio Boccalatte, Flavio De Paoli, Maurizio Martelli, and Viviana Mascardi, editors, *WOA*, pages 55–64. Seneca Edizioni Torino, 2007. ISBN 978-88-6122-061-4.

[56] Michael Schneider, editor. *OWL 2 Web Ontology Language RDF-Based Semantics*. Recommendation. W3C, 2nd edition, 11 Dec 2012. URL https://www.w3.org/TR/owl-rdf-based-semantics/.

[57] Boris Motik. On the properties of metamodeling in OWL. *J. Log. Comput.*, 17(4):617–637, 2007. . URL https://doi.org/10.1093/logcom/exm027.

[58] Regina Motz. OWL extended with meta-modelling. In Francisco Edgar Castillo-Barrera, Reyna Carolina Medina Ramírez, Liliana Ibeth Barbosa-Santillán, Jorge R. Gutierrez Pulido, J. Román Herrera-Morales, and Héctor Gibrán Ceballos-Cancino, editors, *Proceedings of the 4th International Semantic Web and Linked Open Data workshop co-located with 15th Ibero-American Conference on AI (IBERAMIA 2016), San José, Costa Rica, November 23–25, 2016.*, volume 1807 of *CEUR Workshop Proceedings*, pages 55–60. CEUR-WS.org, 2017.

[59] Regina Motz, Edelweis Rohrer, and Paula Severi. Reasoning for *ALCQ* extended with a flexible meta-modelling hierarchy. In Thepchai Supnithi, Takahira Yamaguchi, Jeff Z. Pan, Vilas Wuwongse, and Marut Buranarach, editors, *Semantic Technology – 4th Joint International Conference, JIST 2014, Chiang Mai, Thailand, November 9–11, 2014. Revised Selected Papers*, volume 8943 of *LNCS*, pages 47–62. Springer, 2015.

[60] Natasha Noy, editor. *Representing Classes As Property Values on the Semantic Web*. Working Group Note.

W3C, 5 April 2005. URL http://www.w3.org/TR/swbp-classes-as-values/.

[61] Natasha Noy and Alan Rector, editors. *Defining N-ary Relations on the Semantic Web*. Working Group Note. W3C, 12 April 2006. URL http://www.w3.org/TR/swbp-n-aryRelations/.

[62] OWL Working Group, editor. *OWL 2 Web Ontology Language Document Overview*. Recommendation. W3C, 2nd edition, 11 Dec 2012. URL https://www.w3.org/TR/owl2-overview/.

[63] Jeff Z. Pan and Ian Horrocks. OWL FA: a metamodeling extension of OWL DL. In Les Carr, David De Roure, Arun Iyengar, Carole A. Goble, and Michael Dahlin, editors, *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23–26, 2006*, pages 1065–1066, 2006.

[64] Jeff Z. Pan and Ian Horrocks. RDFS(FA): connecting RDF(S) and OWL DL. *IEEE Trans. Knowl. Data Eng.*, 19(2):192–206, 2007.

[65] Yves Raimond, Samer A. Abdallah, Mark B. Sandler, and Frederick Giasson. The Music Ontology. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR 2007, Vienna, Austria, September 23–27, 2007*, pages 417–422, 2007.

[66] Alan Rector, editor. *Representing Specified Values in OWL: "value partitions" and "value sets"*. Working Group Note. W3C, 17 May 2005. URL http://www.w3.org/TR/2005/NOTE-swbp-specified-values-20050517.

[67] François Scharffe, Ondrej Zamazal, and Dieter Fensel. Ontology alignment design patterns. *Knowl. Inf. Syst.*, 40(1):1–28, 2014.

[68] Patrice Seyed. A method for evaluating ontologies - introducing the BFO-Rigidity Decision Tree Wizard. In *Formal Ontology in Information Systems - Proceedings of the Seventh International Conference, FOIS 2012, Gray, Austra, July 24-27, 2012*, pages 191–204, 2012. . URL https://doi.org/10.3233/978-1-61499-084-0-191.

[69] Nicholas Sioutos, Sherri de Coronado, Margaret W Haber, Frank W Hartel, Wen-Ling Shaiu, and Lawrence W Wright. NCI Thesaurus: a semantic model integrating cancer-related clinical and molecular information. *Journal of biomedical informatics*, 40(1):30–43, 2007.

[70] John F. Sowa. Conceptual graphs. In *Handbook of Knowledge Representation*, pages 213–237. 2008.

[71] Kent A. Spackman, Robert Dionne, Eric Mays, and Jason Weis. Role grouping as an extension to the description logic of Ontylog, motivated by concept modeling in SNOMED. In *AMIA 2002, American Medical Informatics Association Annual Symposium, San Antonio, TX, USA, November 9–13, 2002*. AMIA, 2002.

[72] Patrick Stickler. *CBD – Concise Bounded Description*. Member submission. W3C, 3 Jun 2005. URL http://www.w3.org/Submission/CBD/.

[73] Eiichi Sunagawa, Kouji Kozaki, Yoshinobu Kitamura, and Riichiro Mizoguchi. Role organization model in Hozo. In *Managing Knowledge in a World of Networks, 15th International Conference, EKAW 2006, Podebrady, Czech Republic, October 2-6, 2006, Proceedings*, pages 67–81, 2006. . URL https://doi.org/10.1007/11891451_10.

[74] Ondřej Šváb-Zamazal, Marek Dudáš, and Vojtěch Svátek. User-friendly pattern-based transformation of OWL ontologies. In *Knowledge Engineering and Knowledge Management*, pages 426–429. Springer, 2012.

[75] Vojtěch Svátek, Ondřej Zamazal, and Marek Dudáš. Using ODPs for ontology transformation. In Pascal Hitzler, Aldo Gangemi, Krzysztof Janowicz, Adila Krisnadhi, and Valentina Presutti, editors, *Ontology Engineering with Ontology Design Patterns*, pages 245–266. IOS Press, 2016.

[76] Vojtěch Svátek, Ondřej Zamazal, and Miroslav Vacura. Categorization power of ontologies with respect to focus classes. In *Knowledge Engineering and Knowledge Management - 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings*, pages 636–650, 2016. . URL https://doi.org/10.1007/978-3-319-49004-5_41.

[77] Vojtěch Svátek, Ondřej Zamazal, Simone Serra, and Miroslav Vacura. LOD2 project deliverable 4.4a, ontological annotation of vocabularies and its impact on mapping discovery and verification. Technical report, 2013. URL http://svn.aksw.org/lod2/D4.4a/public.pdf.

[78] Vojtěch Svátek, Martin Homola, Ján Kľuka, and Miroslav Vacura. Mapping structural design patterns in OWL to ontological background models. In V. Richard Benjamins, Mathieu d'Aquin, and Andrew Gordon, editors, *K-CAP*, pages 117–120. ACM, 2013. ISBN 978-1-4503-2102-0.

[79] Vojtěch Svátek, Martin Homola, Ján Kľuka, and Miroslav Vacura. Metamodeling-based coherence checking of OWL vocabulary background models. In Mariano Rodriguez-Muro, Simon Jupp, and Kavitha Srinivas, editors, *OWLED*, volume 1080 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2013.

[80] Vojtěch Svátek, Simone Serra, Miroslav Vacura, Martin Homola, and Ján Kľuka. B-Annot: Supplying background model annotations for ontology coherence testing. In Patrick Lambrix, Guilin Qi, Matthew Horridge, and Bijan Parsia, editors, *WoDOOM*, volume 1162 of *CEUR Workshop Proceedings*, pages 59–66. CEUR-WS.org, 2014.

[81] Vojtěch Svátek, Ján Kľuka, Miroslav Vacura, and Martin Homola. Pattern alternatives for referring to multiple indirectly specified objects. In Eva Blomqvist, Óscar Corcho, Matthew Horridge, David Carral, and Rinke Hoekstra, editors, *WOP@ISWC*, volume 2043 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017.

[82] Duc Thanh Tran, Peter Haase, Bernardo Cuenca Grau, Boris Motik, and Ian Horrocks. Metalevel information in ontology-based applications. In *Proceedings of the 23thrd National Conference on Artificial Intelligence (AAAI 2008)*. AAAI Press, July 2008. URL http://web.comlab.ox.ac.uk/people/Boris.Motik/pubs/dhmgh08-metalevel-information.pdf.

[83] Pierre-Yves Vandenbussche, Ghislain Atemezing, María Poveda-Villalón, and Bernard Vatant. Linked open vocabularies (LOV): A gateway to reusable semantic vocabularies on the web. *Semantic Web*, 8(3):437–452, 2017.

[84] Ondřej Šváb Zamazal, Vojtěch Svátek, and Luigi Iannone. Pattern-based ontology transformation service exploiting OPPL and OWL-API. In Philipp Cimiano and Helena Sofia Pinto, editors, *EKAW*, volume 6317 of *Lecture Notes in Computer Science*, pages 105–119. Springer, 2010. ISBN 978-3-642-16437-8.

[85] Christopher A. Welty. OntOWLClean: Cleaning OWL ontologies with OWL. In *Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, FOIS 2006, Baltimore, Maryland, USA, November 9-11, 2006*,

pages 347–359, 2006. URL http://www.booksonline.iospress.nl/Content/View.aspx?piid=2221.

Appendix A

Instructions from Evaluation 1, phase 2 (approx. translation)

Fill in the "PURO" column with the PURO category label you consider as most appropriate for the given entity. The possible category labels are: CO, CR, CV, CTO, CTR, and CTV, see the *PURO Annotation Guidelines*.

Then in fill the "Confidence" column with the confidence of this PURO category, as an integer value, as follows:

1. I feel almost/completely confident in the label. Information about the entity, available in rdfs:label and rdfs:comment values, in the summary text in the HTML specification of the ontology and in the entity's arrangement into the structure of the ontology (as viewed in Protégé) fits the description of the PURO category in the Annotation Guidelines.
2. I am leaning towards the given label, but with hesitation, since some entities of the PURO category in the Annotation Guidelines do not fit perfectly, or multiple categories look relevant.
3. I am rather guessing by method of exclusion; no category description in the Annotation Guidelines fits what I observe about the entity, or the entity corresponds to multiple categories to the same degree.

Entities from Evaluation 1, phase 2

The entities are listed in Table 12 together with their first and (optional) second choice by ontology engineering experts.

Table 12
Entities from Evaluation 1

	Experts'	
Entity	1st choice	2nd choice
gr:Brand	CTO	CO
gr:BusinessEntity	CO	
gr:BusinessEntityType	CTO	
gr:BusinessFunction	CTR	
gr:DeliveryChargeSpecification	CR	CV
gr:Location	CO	
gr:OpeningHoursSpecification	CV	
gr:PriceSpecification	CV	CR
gr:WarrantyPromise	CR	CV
gr:WarrantyScope	CTR	CR
mo:Festival	CO	
mo:Genre	CTO	CO
mo:Lyrics	CO	
mo:MagneticTape	CO	
mo:Membership	CR	
mo:Record	CTO	CO
mo:Recording	CR	
mo:ReleaseStatus	CTO	
mo:SignalGroup	CO	
mo:Sound	CO	