What Can the Ontology Describe? Visualizing Local Coverage in PURO Modeler

Marek Dudáš, Tomáš Hanzal, and Vojtěch Svátek

University of Economics, Prague,
marek.dudas|xhant00|svatek@vse.cz

Abstract. Ontologies and vocabularies written in OWL are a crucial part of the semantic web. OWL however allows to model the same part of reality using different combinations of constructs, constituting 'modeling styles'. Comparing how different ontologies from a similar domain cover a specific part of reality might be more difficult when each ontology uses a different style. PURO, a language for ontological background models, can serve as mediator, as it allows to create models that are directly mappable to OWL but overcome its unnatural modeling limits dictated by description logic. By highlighting the parts of the PURO model covered by particular ontologies, the ontology coverage comparison can be visualized. We demonstrate this approach using a simple graphical tool.

1 Introduction

OWL [8], the language of ontologies on the semantic web, allows to model the same real-world state of affairs using different combinations of language constructs, a kind of modeling styles. For example, in a lightweight ontology, data properties might be preferred, while in a more complex ontology, object properties might be used for describing the same relationships, allowing to state more facts about the property value. Current ontology visualization tools (as surveyed in [4,6]) either display the OWL constructs directly or only offer very lightweight generalization, as in [5]. When the user wants to compare, in a visual manner, the *local coverage*[1] of different ontologies, i.e. their capability to express a certain cluster of relationships, s/he has to first translate the OWL language constructs into his/her mental model to abstract from the modeling differences.

A possible solution in such a situation is to express the above mentioned cluster of relationships in a modeling language that allows to abstract from the modeling style differences and thus make the mental model explicit. We believe that *PURO ontological background models* (OBMs) [9] might fit this use case. We are developing PURO Modeler, a Javascript-based tool for graphical design of OBMs and ontology local coverage comparison.[2] In the paper, we demonstrate its usage on two examples.

[1] We assume that typically only a few entities and relationships are considered from the analyzed models (e.g., a book, its author and its subject). To differ from the coverage of the whole domain (e.g., bibliography), we use the term 'local coverage'.

[2] The development version is available at http://protegeserver.cz/puromodeler.

Related Research We are unaware of any prior research on visual comparison of ontology local coverage. Creating and visualizing mappings [3] between the compared ontologies might be useful but it only shows what the ontologies have in common. There is research on ontology comparison (e.g., [7]), but that is rather focused on automatic computation of ontology similarity. In our approach the actual analysis is left to the user; we however propose a method and means for supportive visualization. The PURO language, designed to be easily mappable to OWL, is used as interlingua for this purpose. Other modeling languages, e.g., entity-relationship diagrams [2], might be considered as alternatives to PURO. Of those, *OntoUML* [1], a version of UML for conceptual modeling, is probably the closest match; an existing tool, OLED, even allows to transform OntoUML models into OWL fragments. However, the purpose of OntoUML is conceptual modeling with easy validation and it is not designed for OWL ontology development or usage.

2 PURO Ontological Background Models

The long-term vision of the PURO OBM language is to model the real world, as much as possible, 'as it is', while remaining 'very close' to OWL. It is only intended to serve as an aid in ontological engineering, i.e. models created in it are not supposed to serve (in large scale) as schema for data or input to reasoners. The liberation from the constraints dictated by description logic allowed to drop some constraints of OWL (DL, or even Full) that are often perceived as unnatural; most notably, meta-concepts (such as classes of classes, or classes as 'property values') and arbitrary n-ary relationships can be expressed in PURO.

PURO OBMs are based on two main distinctions: between particulars and universals and between relationships and objects (hence the PURO acronym: Particular-Universal, Relationship-Object). In simple terms: universals, i.e. types, can have instances, while particulars cannot (P-U distinction); objects and relationships are differed by their identification: objects are entities with their own identity that can be 'talked about' independently, while talking about relationships always 'brings in' their participating entities (R-O distinction). There are six basic terms: B-object (particular object), B-type (type of objects/types), B-relationship (particular relationship), B-relation (type of relationship), B-valuation (particular assertion of quantitative value) and B-attribute (type of valuation).

An OBM consists of named *instances* of these terms and relationships between them.[3] To say, it captures a concrete situation that serves as basis for either analyzing existing ontologies or generating new ones.

3 Using PURO Modeler for Local Coverage Comparison

PURO Modeler is basically a graph editor. The user can create an OBM where the terms are represented by nodes and the relationships between them by edges.

[3] The description of the language is simplified, more details are in [10].

As the OBMs are models of specific real-world situations, such an example cluster of relationships has to be chosen. After constructing the model,[4] or loading an already existing one (possibly created by another user and shared), the user can select each node and annotate it with the ontologies that this particular instance of PURO term is covered by (i.e., the instance can be described with that ontology). The parts of the graph locally covered by each vocabulary are then highlighted so that the coverage could be easily compared in one diagram.

3.1 Example 1: A Book and its Issue

Let us say we need to annotate data about books and their various issues (i.e. their manifestations). We can find several ontologies that might be used for it. **BIBFRAME**[5] contains class Text as a subclass of *Work*, class Print as a subclass of *Instance* and the *hasInstance* property for stating that an instance of *Print* is a manifestation of some *Text*. Figure 1b shows an example of possible usage of BIBFRAME to describe the book *The Semantic Web for the Working Ontologist* and its printed paperback manifestation. **FRBR** ontology[6] (and its extension **Fabio**[7]) distinguishes between four different 'levels of abstraction' represented by classes *Work*, *Expression*, *Manifestation* and *Item*, each with subclasses representing more specific types. There is a set of properties that can be used to link the instances of the classes, as shown in Figure 1c. **Schema.org** does not distinguish between the work and its manifestation through class membership, but contains properties *exampleOfWork* and *workExample* that serves that purpose. It contains the class *Book* and allows to specify the format of its instances by linking them to instances of *BookFormatType* with property *bookFormat* (Figure 1a).

For the purposes of the local coverage comparison, we chose the book *The Semantic Web for the Working Ontologist* (as an instance of the B-type Text, which is a subtype of Work) and its 2009 paperback issue (an instance of the B-type Paperback, a subtype of Book). An OBM of these two entities accompanied by a B-object representing an exemplar of the paperback issue (as a tangible item) and the relationships between them is in Figure 2 (exported as a screenshot from PURO Modeler). The color-coded highlighted parts of the model shows what of it can be expressed in each vocabulary.

We can see that, e.g., the physical exemplar of the issue can only be described with FRBR (which 'implements' the whole OBM), BIBFRAME allows to describe an issue of a book, but without specifying its format, and Schema.org does not allow to say that an instance of work is a *Text*.

[4] The methodology for the OBM modeling is yet to be formalized.
[5] `http://bibframe.org/vocab`
[6] `http://purl.org/vocab/frbr/core#`
[7] `http://purl.org/spar/fabio/`

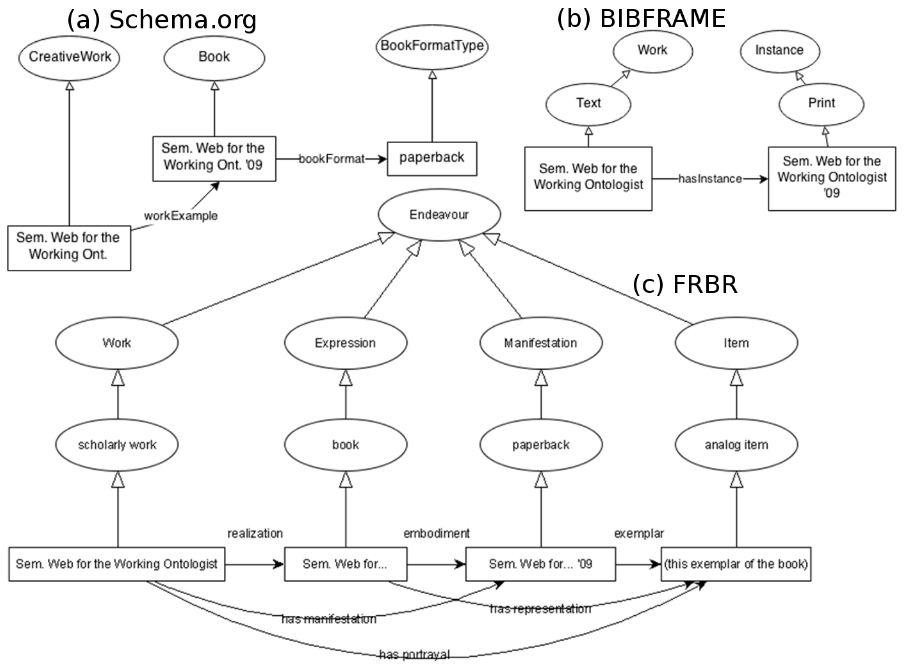

Fig. 1. Diagrams showing three different OWL modeling styles of the relationship between a book and its issue.

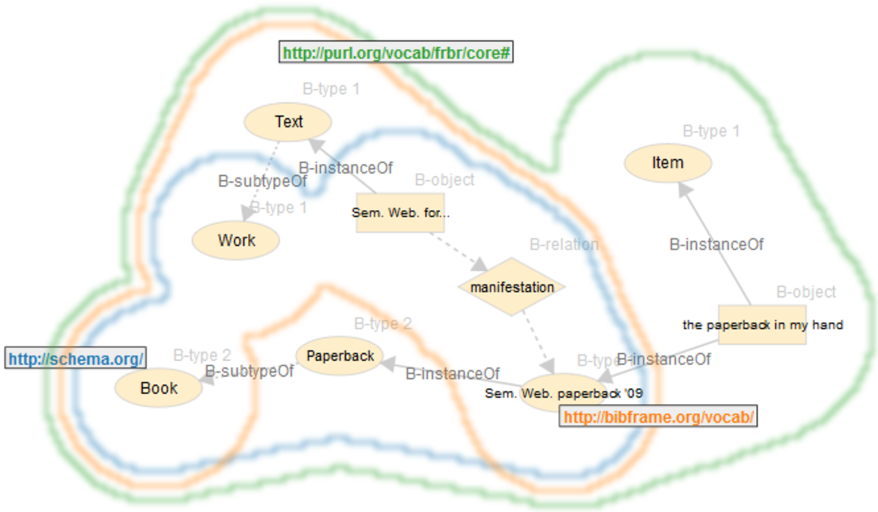

Fig. 2. The OBM of the relationship between a book and its issue. The amount of its coverage in various OWL representations (shown in Figure 1) is highlighted.

3.2 Example 2: A Dish and a Recipe

In the second example, we compare the local coverage of the relationships between ingredients, a recipe and a dish produced by it. We analyzed[8] three ontologies: Schema.org, Food Ontology[9] and Linked Recipes.[10] Their local coverage is highlighted in Figure 3 (for simplicity, we made up a 'boiled egg recipe' with only one ingredient and we did not include all possible B-attributes of the B-objects, like 'protein content' of the dish).

We can see that Schema.org allows to describe the recipe in terms of its origin, ingredients and time needed; the carbohydrates, fat and energy content; and size of the produced food. It does not allow to model the recipe instructions: there is only one datatype property for entering a free text description. Linked Recipes does not allow to describe the characteristics of the recipe product, but (unlike Schema.org) allows to type it as 'Food'. It can be used to annotate instructions including their order, and ingredients including their quantity. Food Ontology can only be used for the (quite detailed) description of the ingredients and the resulting dish. It does not contain classes/properties relevant for cooking instructions. None of the three ontologies covers the whole situation as modeled by the OBM, however, we can see that a combination of two of them might be sufficient, e.g., using Linked Recipes for recipes in combination with Food Ontology for more detailed ingredient and dish description.

4 Conclusions and Future Work

We argued that the task of ontology comparison in terms of local coverage is difficult if the ontologies use different modeling styles. We proposed that using a more general modeling language for visualizing the local coverage of several ontologies in one place, abstracting from the OWL modeling differences, might make the comparison easier, and that the PURO OBM language might be suitable for such a use case. As a first step towards the evaluation of the approach we implemented PURO Modeler: a Javascript application for PURO OBM models design and visualization of the amount of their local coverage in various ontologies. We demonstrated the usage of PURO Modeler for local coverage comparison on two examples, each consisting of an analysis of three ontologies.

The future work might include, besides the full evaluation, building a portal where the developers could publish visualizations of their ontology/vocabulary local coverage of typical situations. Such a website could then serve as an addition or enhancement of the existing Linked Open Vocabularies[11] portal. Intense ongoing research also addresses the obvious bottleneck of the approach: the design of plausible PURO OBMs. It is going to be supported by interactive guidelines and NLP-based model verbalization widgets.

This research is supported by the VŠE IGA project no. F4/34/2014.

[8] See `http://tomhanzal.github.io/owl-modeling-styles/#part2` for full analysis.
[9] `http://data.lirmm.fr/ontologies/food/`
[10] `http://linkedrecipes.org/schema/`
[11] `http://lov.okfn.org/dataset/lov/`

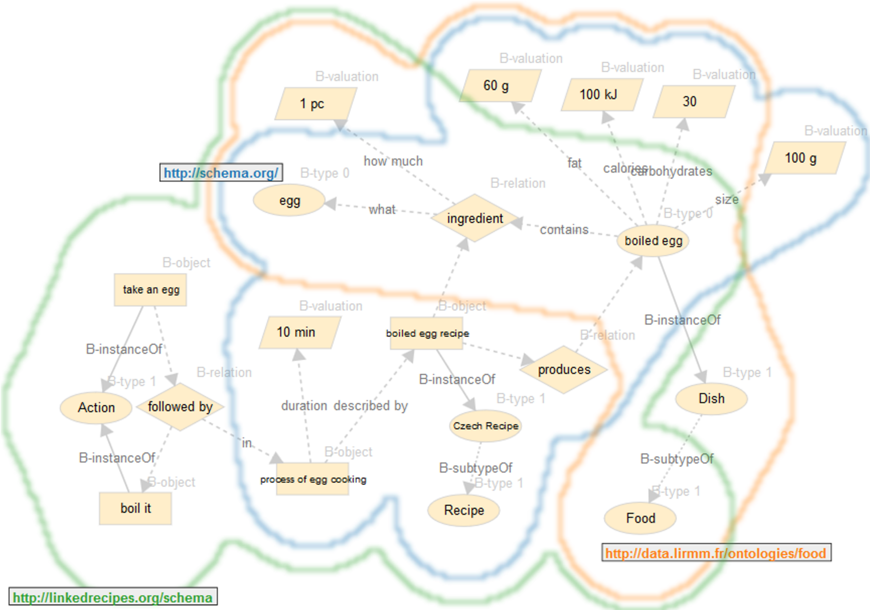

Fig. 3. The OBM of a recipe, ingredients and dish. The amount of its coverage in various OWL representations is highlighted.

References

1. Albuquerque, A., Guizzardi, G.: An ontological foundation for conceptual modeling datatypes based on semantic reference spaces. In Research Challenges in Information Science (RCIS), 2013 IEEE Seventh International Conference, 2013.
2. Chen, P.: The entity-relationship model – toward a unified view of data. In: *ACM Transactions on Database Systems* (TODS), 1976, 1.1: 9-36.
3. Choi, N., Song, I. Y., Han, H.: A survey on ontology mapping. ACM Sigmod Record, 35(3), 34-41. 2006.
4. Dudáš, M., Zamazal, O., Svátek, V.: Roadmapping and Navigating in the Ontology Visualization Landscape. Accepted to EKAW 2014.
5. Hayes, P., et al.: Collaborative knowledge capture in ontologies. In: Proceedings of the 3rd international conference on Knowledge capture – K-CAP 2005, ACM, 2005.
6. Katifori, A., et al.: Ontology Visualization Methods – A Survey. *ACM Comput. Surv.* 39, 10143. 2007.
7. Maedche, A., Staab, S.: Measuring similarity between ontologies. In: Knowledge engineering and knowledge management: Ontologies and the semantic web. Springer Berlin Heidelberg, 2002. p. 251-263.
8. *OWL 2 Web Ontology Language Document Overview.* W3C Recommendation, 2009.
9. Svátek, V., et al.: Mapping Structural Design Patterns in OWL to Ontological Background Models. In: *K-CAP 2013*, ACM, 2013.
10. Svátek, V., et al.: Metamodeling-Based Coherence Checking of OWL Vocabulary Background Models. In: *OWLED 2013*, online `http://ceur-ws.org/Vol-1080/owled2013_6.pdf`.