OWL Data Properties Ontologically

Vojtěch Svátek[1,*], Kateřina Haniková[1] and Ondřej Zamazal[1]

[1]*Prague University of Economics and Business, Czechia*

Abstract

Data properties are a frequently used construct in OWL ontologies. As the name suggests, they are however biased towards the world of 'data' rather than that of ontologically grounded entities. On the other hand, we hypothesize that many data properties bear inherent meaning that can be captured through simple ontological categories (such as objects, relationships or events). We formulated intuitive categories of data property 'background' categories, only indirectly connected to the 'surface' categories (i.e., to data property range options) and performed an annotation experiment of a sample of data properties from different ontologies according to this system of categories. The results indicate that the ontological nature of most data properties can be characterized with reasonable human effort, opening the question of possible automated analysis in the future.

Keywords

Ontology, OWL, data property, ontology reengineering

1. Introduction

The abundance of data properties is an aspect that often distinguishes semantic web ontologies (foremost, OWL) from philosophically grounded reference ontologies as well as from theories expressed primarily in formal logic. The values of data properties are literals, which belong (either explicitly, for typed literals, or implicitly, for simple literals) to data types. While custom data types can be defined in OWL, the range of a vast majority of data properties in ontologies is one of pre-defined data types, corresponding to what is commonly used in programming languages: `xsd:string`, `xsd:integer`, `xsd:float`, `xsd:boolean`, `xsd:dateTime`, or the like.

Ontology-based knowledge graphs are however not of the same nature as procedural program code. Their statements are meant to be declarative facts aiming to describe situations occurring in the world. We hypothesize that this applies not only to object property assertions but also to data property assertions, although a bit less obviously and perhaps with exceptions. To say, the values of data properties, as well as these properties themselves, may refer to "background" ontological entities that can be (were this the modeler's choice) equally expressed by different means, through classes and individuals interconnected by object properties. For example, if a string-range data property refers to a country code (e.g., `someprefix:countryOfBirth`), it actually expresses the relationship of an entity (here, a person) to a country. Similarly, a boolean property may serve for defining a distinction that could equally be modelled as a class (say, a boolean data property `someprefix:retired` may be replaced with a class `someprefix:RetiredPerson`). Note that the background semantics does not unambiguously follow from the data type. In particular, a string may also refer to other things than 'real objects'. Similarly, an integer may express, for example, a mere truncated quantitative value (e.g., the `someprefix:maxSpeed` of a car) or the count of links to objects of some kind (e.g., `someprefix:numberOfDoors` of a car).

Successful capturing of the ontological background of data properties may serve various practical purposes. Feedback on implicit 'background' structures may lead to suggestions for the ontology designers/maintainers to make these structures *explicit* via a reengineering process, and thus increase

EKAW 2024: EKAW 2024 Workshops, Tutorials, Posters and Demos, 24th International Conference on Knowledge Engineering and Knowledge Management (EKAW 2024), November 26-28, 2024, Amsterdam, The Netherlands.

*Corresponding author.

svatek@vse.cz (V. Svátek); katerina.hanikova@vse.cz (K. Haniková); ondrej.zamazal@vse.cz (O. Zamazal)

https://nb.vse.cz/~svatek/ (V. Svátek); https://nb.vse.cz/~svabo/ (O. Zamazal)

0000-0002-2256-2982 (V. Svátek); 0009-0009-7162-5925 (K. Haniková); 0000-0002-7442-9016 (O. Zamazal)

the expressiveness and/or readability of the knowledge graphs for particular purposes. It may also be supportive in *ontology reuse* and *ontology alignment*.

2. Related Research

We are unaware of any study focusing on the problem of this 'background' meaning of data properties in OWL. Empirical studies on OWL ontology collections [1] merely focused on the distribution of data types. A rare attempt to systematically involve data properties into ontology design patterns was the work by Fillottrani and Keet [2], which considered data properties as an alternative modeling solution to object properties. This work was not, however, accompanied by any empirical study, and was confined to a narrow part of the data property ontological background.

Our study follows up with earlier works on analyzing the ontological background of OWL ontologies. Different systems of foundational distinctions were applied for this purpose: OntoClean with its rigidity, identity and unity [3], PURO with particular/universal and relationship/object/valuation distinctions [4], or a subset of the UFO foundational ontology distinctions called GUFO [5]. The usually targeted OWL construct (annotated by foundational 'meta-properties') were classes. The Protégé plugin *B-Annot* [6] allowed to annotate OWL ontologies (both generically and with respect to their use in the schema of a particular dataset) by both OntoClean and PURO distinctions. For the latter, it also dealt with OWL properties (whether object or data ones); however, the set of meta-property options was refined to a few basic ones (properties valued by what is inherently an object, a class, a relationship, or a valuation) which could not sufficiently characterize the property's ontological background.

3. Initial Ontological Categorization of Data Properties

In our current research, we decided to essentially reuse the PURO distinctions; they are centered around a similar, simple meta-model system as that of OWL (PURO objects, types, relationships and valuations resemble OWL individuals, classes, object properties and data properties), thus instructively manifesting the divergence between the surface and ontological view. However, compared to the earlier works [4, 6], we specifically zoom on a particular OWL construct, the data properties, and provide an empirical study for it, with a relatively fine-grained and structured system of categories (potentially interpretable as meta-properties).

The initial version of categories considered did not rely upon any empirical study, but merely upon the accumulated experience of the authors from ontology engineering projects. In this initial version, 10 categories were suggested (name, identifier, object, type, message, quality, quantitative value, count, date/time, existentially quantified relationship). The categories were (non-uniquely) grouped based on clusters of built-in XSD or RDF data types,[1] which were, in turn: string (also including, e.g., `rdfs:Literal` or `xsd:language`), integer (and its variation, e.g., non-negative integer), non-integer numerical (float, double, decimal), date/time (again, all variations of temporal types) and boolean.

- For a string property:
  - **Name**: the property provides a human-readable *name* for the given instance of the domain[2] class.
  - **Id**: the property provides an *identifier* for the given instance of the domain class, for example, a DOI for a publication, or an ORCID for a researcher.
  - **Obj**: the string value of the property represents some particular *object* to which the for the given instance of the domain class is connected.
  - **Tp**: the string value of the property refers to a *type* that is, through the data property, implicitly assigned to the given instance of the domain class (e.g., somePref:Car-1 somePref:carType

[1] https://www.w3.org/TR/xmlschema-2/#built-in-datatypes

[2] By 'domain class' we understand the class that is assigned to the property through the `rdfs:domain` predicate.

"SUV"). It is a kind of anti-pattern, as the type would always be better expressed by a class or object property.
  - **Qual**: the string value of the property expresses some *quality* of the given instance of the domain class (e.g., colour "red"), which is neither its type nor another related object, in the usual sense.
  - **Msg**: the string value of the property is a human-readable message that either does not reference any domain entities or the 'background' of this message is too vague or too complex to be modeled as a structure.
- For an integer property:
  - **Quant**: the integer value of the property is in fact a (rounded) value of *quantitative* property of the given instance of the domain class, which is natively non-integer, such as size, maximum speed, length etc.
  - **Count**: the property is a *counting* property, expressing the fact that there is a finite number of objects connected to the given instance of the domain class via the same kind of relationship (e.g., the number of free workplaces available in the given company).
- For a non-integer numerical property (only one option anticipated):
  - **Quant**: the property value is an (inherently) *quantitative*, continuous-valued property of the given instance of the domain class (e.g., its weight, length, speed, or the like).
- For a date/time property (only one option anticipated):
  - **DTm**: the property value represents the *date* and or *time* of the given instance of the domain class – meaning that this instance is an event or process ('perdurant', in an ontological sense).
- For a boolean property:
  - **Tp**: the boolean values of the property indicate whether the given instance of the domain class is an instance of a specific *type* (e.g., somePref:Dataset-1 somePref:isPublic "false").
  - **Ex**: an *existentially* quantified relationship where the boolean values indicate whether the given instance of the domain class does or does not participate in a particular relationship to another object, whatever the identity/type of this object is (e.g., somePref:Car-1 somePref:hasInsuranceContract "true", where the contract as such is not assumed to be modeled as an explicit object).

Additionally, an option **Chain**+*<Categ>* was anticipated, such that the value of the property would correspond to category *<Categ>*, however, there would be an (anonymous) intermediate object between the instance of the domain class and the entity referred to by the value.

4. Empirical Study and Refined Categorization

We randomly selected 30% of ontologies from the Archivo repository (currently containing 1811 ontologies)[3] and got 543 ontologies. We ran a simple SPARQL query on this sample to detect all data properties that are in these ontologies and only selected those having both a label (`rdfs:label`) and a comment (`rdfs:comment`). From these data properties, we selected at most five for each ontology. This way we got a dataset containing 202 data properties (with their domains and ranges) which were from 64 ontologies (i.e. those containing at least one data property). Finally, to make thorough manual analysis feasible in the limited time, we only picked the first 100 data properties, which corresponded to 30 ontologies.

The annotation campaign was held during August and September 2024 and there were three annotators involved: the paper authors, of which two are ontology engineering experts and one is a PhD student researching in this field, with moderate expertise. The task of all annotators was to select the most adequate ontological category based on simple guidelines;[4] the properties were the same for all.

[3] https://archivo.dbpedia.org/

[4] https://github.com/Onto-DESIDE-VSE/TransformationPatterns/tree/main/experiments/data%20properties%20EKAW24

Table 1
Annotators agreement before workshop

Data type	All agreed	Two agreed	Disagreement	Total
String	25	18	2	45
Integer value	9	9	1	19
Non-integer numerical value	11	1	0	12
Date/time	10	1	0	11
Boolean	5	0	1	6
IRI (Any URI type)	1	6	0	7
Total	61	35	4	100

The annotators only examined the labels and comments of the entities that were an explicit part of the pattern, i.e. they did not examine the ontology as a whole.

The annotation of one property took approximately 1.5 minutes on average, ranging from several minutes for tricky cases to tens of seconds for apparently repeated phenomena (however, given the confinement to max. 5 cases from a single ontology, intra-ontology regularities did not have much impact on such repetitiveness).

After the first round, all three annotators agreed in 61 cases, in another 35 cases two annotators agreed, and there were only 4 total disagreements. The breakdown per data type (property range) groups is in Table 1. Note that the table contains one additional row, for the `xsd:anyURI` type, which had not been considered in the initial category design and could not clearly fit into any of the existing clusters.

This was followed by a workshop, where all three annotators gathered around and discussed the disagreements and the proposed new categories. Through the workshop, agreement was reached in 99 cases out of 100. Of the 38 cases of posterior agreement:

- For 27, all annotators eventually converged to the verdict of one (or two) of them.
- For 5, it was determined that the guidelines were ambiguous in the sense of allowing to give two different labels to what was the same situation in reality (thus upon the curating the guidelines, the clash would not exist any longer). This concerned the alternative categorizations of the value being a *related object* (Obj) vs. the *Id* or *Name* of this related object: the object was always assumed to be manifested via an Id[5] or a name, but some annotators put stress on the fact that this is a related object and others on the nature of its manifestation (Id or Name).
- For 6, the annotators agreed that based on the available information, it was not possible to tell whether the values were to be interpreted as objects or their types. For example, in an ontology about used cars, there was a class *"Modification and maintenance"* and a datatype property called *"part removed"* with its range *rdfs:Literal*. The question is whether the part removed while repairing a car is understood as being described at the level of this concrete, physical part ('left headlight of car 1234'), or of the type of this part ('a headlight'). To determine this, we would have to delve into a concrete knowledge graph, as was previously the case with B-Annot [6].

Some of the discussions led to creation of new categories that had not been covered by the guidelines. For example, for integer values, we had anticipated two categories: continuous quantity (merely truncated to an integer) and count (as an inherent integer). We however discovered two other, also fairly intuitive, meanings of integer values. First, we discovered a property expressing the rank of the domain object (e.g. wrt. ranking a movie). We decided to have a new category called *Rank* for such cases, to distinguish this, *ordinal*, case from both continuous quantities and counts. The rank positions the domain object with respect to other objects and can change even without any alteration of the given object (merely due to alterations of other objects within the ranking). Second, we also discovered a property that expressed the *level* of an object (which was a gaming character, in this case). While the

[5]This corresponds to most of the cases of disagreement for the IRI datatype in Table 1.

level is also ordinal, unlike the rank it depends on the characteristics of the domain objects and is not influenced by the level of other objects. We conceive this category as a special subkind of Type, namely, *"Type – ordinal"*.

Finally, a substantial revision of the overall system of categories was triggered by the discovery of data properties that, in ontological terms, merely consisted in an 'arm' of a relationship (or of a quantitative value assignment). An example is a property *"from"* with domain class *"Ownership info"*. It seems that the class is actually a reification of the ownership relationship, whose semantics is "An agent owns a vehicle within a certain time period under a certain registration plate." Under the PURO lenses, stipulating that relationships should be kept unreified as long as possible (thanks to the support of n-ary relationships, of whatever arity and whatever kind of participants, in PURO) *"from"* corresponds not to a complete relationship (never mind a chain thereof) but to a subpart of it.

The refined system of 12 ontological categories and 2 'special structural cases', resulting from the annotation campaign and workshop, and planned to be used in the next round of experiments, is then as follows (new/redefined categories in bold; existing ones same as in the initial system from Section 3):

- For a string property: Name, Id, Obj, Tp, Qual,[6] Msg
- For an integer property: Quant, Count, **Tp** (added here for the case of ordered types), **Rank** (new category)
- For a quantitative or date/time property: Quant, DTm
- For a boolean property: Tp, Ex, **Match** (new category)
- For any property, special structural cases:
  - Chain+<*Categ*>: the value of the property corresponds to category <*Categ*>, however, there is an (anonymous) intermediate object between the instance of the domain class and the entity referred to by the value.
  - **Part**+<*Categ*>: the value of the property corresponds to category <*Categ*>, however, the instance of the domain class can be understood as a reified relationship.

5. Conclusions and Future Work

We are presenting a presumably pioneering study focused on the ontological grounding of data properties in OWL ontologies. The presented work is a part of a larger initiative aiming at providing a framework for (semi-)automatically transforming the structure of ontologies, and, accordingly, the knowledge graphs based on them, so that the shape of those graphs would fit a particular application requirements.

As with all such small-scale manual studies, in which new concepts emerge on the fly, a natural next step will be to pick up another random collection of data properties (ideally, from different ontologies) and see how well the current version of the category system is applicable and whether it tends to converge or new ontological structures keep emerging.

In longer term, we are aware that analyzing the data properties manually is a relatively demanding exercise. As soon as we gather a decent pool of labeled training examples (for an already consolidated system of categories), we plan to take advantage of *deep-learning classifiers*, which could be fine-tuned for the given task through the provided examples.[7]

Acknowledgments

This work has been supported by the EU's Horizon Europe grant no. 101058682 (Onto-DESIDE).

[6] Not encountered in the initial sample.

[7] A similar approach has already been proven useful for providing UML models with UFO-based stereotypes [7].

References

[1] N. Matentzoglu, S. Bail, B. Parsia, A snapshot of the OWL web, in: H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. F. Noy, C. Welty, K. Janowicz (Eds.), The Semantic Web - ISWC 2013 - 12th International Semantic Web Conference, Sydney, NSW, Australia, October 21-25, 2013, Proceedings, Part I, volume 8218 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 331–346. URL: https://doi.org/10.1007/978-3-642-41335-3_21. doi:10.1007/978-3-642-41335-3_21.

[2] P. R. Fillottrani, C. M. Keet, Patterns for heterogeneous tbox mappings to bridge different modelling decisions, in: E. Blomqvist, D. Maynard, A. Gangemi, R. Hoekstra, P. Hitzler, O. Hartig (Eds.), The Semantic Web - 14th International Conference, ESWC 2017, Portorož, Slovenia, May 28 - June 1, 2017, Proceedings, Part I, volume 10249 of *Lecture Notes in Computer Science*, 2017, pp. 371–386. URL: https://doi.org/10.1007/978-3-319-58068-5_23. doi:10.1007/978-3-319-58068-5_23.

[3] C. A. Welty, OntOWLClean: Cleaning OWL ontologies with OWL, in: Formal Ontology in Information Systems, Proceedings of the Fourth International Conference, 2006, pp. 347–359.

[4] V. Svátek, M. Homola, J. Kľuka, M. Vacura, Metamodeling-based coherence checking of OWL vocabulary background models., in: M. Rodriguez-Muro, S. Jupp, K. Srinivas (Eds.), OWLED, volume 1080 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2013.

[5] P. P. F. Barcelos, T. P. Sales, E. Romanenko, J. P. A. Almeida, G. Engelberg, D. Klein, G. Guizzardi, Inferring ontological categories of OWL classes using foundational rules, in: N. Aussenac-Gilles, T. Hahmann, A. Galton, M. M. Hedblom (Eds.), Formal Ontology in Information Systems - Proceedings of the 13th International Conference (FOIS 2023), Sherbrooke, Quebec, Canada, July 17-20, 2013 and Virtual Event, September 18-20, 2013, volume 377 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2023, pp. 109–124. URL: https://doi.org/10.3233/FAIA231122. doi:10.3233/FAIA231122.

[6] V. Svátek, S. Serra, M. Vacura, M. Homola, J. Kľuka, B-Annot: Supplying background model annotations for ontology coherence testing., in: P. Lambrix, G. Qi, M. Horridge, B. Parsia (Eds.), WoDOOM, volume 1162 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2014, pp. 59–66.

[7] S. J. Ali, G. Guizzardi, D. Bork, Enabling representation learning in ontology-driven conceptual modeling using graph neural networks, in: M. Indulska, I. Reinhartz-Berger, C. Cetina, O. Pastor (Eds.), Advanced Information Systems Engineering - 35th International Conference, CAiSE 2023, Zaragoza, Spain, June 12-16, 2023, Proceedings, volume 13901 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 278–294. URL: https://doi.org/10.1007/978-3-031-34560-9_17. doi:10.1007/978-3-031-34560-9_17.