Building Ontologies for Reuse

Lessons Learned from Unit Ontologies

Sirko SCHINDLER [a1], Jan Martin KEIL [b2]
[a] *Institute of Data Science, German Aerospace Center (DLR), Jena, Germany*
[b] *Heinz Nixdorf Chair for Distributed Information Systems, Friedrich Schiller University Jena, Jena, Germany,*

Abstract. Reusability is a key advantage promised by ontologies. But in practice, the reuse is oftentimes impeded or even prevented by bad ontology designs. In this case study, we report on experiences when trying to utilize existing ontologies for measurement units in a scientific data management system. For this well defined domain, there is a wide range of ontologies and modeling approaches available. However, the models lend themselves differently to reuse. We want to draw ontology engineers' attention to encountered examples of good and bad design decisions to be considered in future developments.

Keywords. ontology reuse, ontology application, ontology engineering, ontology maintenance

1. Introduction

Ontologies represent knowledge in a machine-interpretable way, and as such they are an invaluable component of many knowledge-based applications. During their design, ontology engineers are urged to reuse existing ontologies wherever possible. This reduces the efforts needed to model the domain at hand and increases the interoperability across applications. Further, the frequent reuse of an ontology will uncover errors and thus improve the ontology. Literature distinguishes three kinds of ontology reuse: (a) *Hard reuse* imports complete ontologies [1], (b) *soft reuse* only references entities of another ontology without importing it [1], and (c) *direct application* employs an existing ontology without creating a new one at all [2].

In practice, however, reusing ontologies may fail for various reasons. Kamdar et al. [2] noticed many cases of intended entity reuse that failed due to erroneous Internationalized Resource Identifier (IRI) references. Fernández-López et al. [1] identified five reuse problems: Missing support of a particular natural language, missing documentation, unavailable dependencies, licensing issues, and heterogeneity between needed and provided concepts. Furthermore, general quality issues, such as described in the ontology pitfalls catalog [3], can prevent reuse.

[1] sirko.schindler@dlr.de, `https://orcid.org/0000-0002-0964-4457`
[2] jan-martin.keil@uni-jena.de, `https://orcid.org/0000-0002-7733-0193`

To complement these findings, we report in this case study on experiences when trying to utilize existing ontologies for measurement units in the scientific data management project LakeBase [4]. For this well-defined domain, there is a wide range of ontologies and modeling approaches. However, the models lend themselves differently to reuse. In previous work, we compared and evaluated nine unit ontologies [5,6]. In this paper, we want to draw ontology engineers' attention to encountered examples of good and bad design decisions to be considered in future developments. In Section 2, we provide a catalog of anti-patterns related to the choice of IRIs (2.1), identity and equivalency (2.2), and the design of properties (2.3). In Section 3, we conclude with possible directions to tackle these issues.

2. Issues

In the following, we discuss issues that we encountered during our efforts to reuse an ontology for measurement units. We illustrate them using examples from the following ontologies. However, they also apply to other unit ontologies and beyond. Due to our previous analysis [6], some of the mentioned examples have been fixed in newer ontology versions.

- Semantic Web for Earth and Environmental Terminology (SWEET)[3]
- Measurement Units Ontology (MUO)[4]
- Extensible Observation Ontology (OBOE)[5]
- Quantities, Units, Dimensions and Data Types Ontologies (QUDT)[6]
- Library for Quantity Kinds and Units (QU)[7]
- Ontology of units of Measure (OM)[8]

2.1. Choosing good IRIs

The first class of issues is related to the choice and maintenance of namespaces and IRIs. Kamdar et al. [2] noticed problems while reusing IRIs of entities. Ontology engineers occasionally inserted errors in the IRIs of the entities they reused. They could be supported by a careful choice of IRIs. Based on our experiences with unit ontologies, we give some advices for a robust choice of IRIs. All of these are rooted in the principle of simple, stable, and manageable IRIs [7,8]. In the following, we will highlight instances of what we consider anti-patterns.

[3] `https://web.archive.org/web/20170802032920/https://sweet.jpl.nasa.gov/`
[4] `https://web.archive.org/web/20160323142147/http://idi.fundacionctic.org/muo/`
[5] `http://ecoinformatics.org/oboe/oboe.1.0/oboe-standards.owl`
[6] `http://www.qudt.org/`
[7] `https://www.w3.org/2005/Incubator/ssn/ssnx/qu/`
[8] `http://www.wurvoc.org/vocabularies/om-1.8/`

1a. IRIs should not contain the ontology version. We noticed that the IRIs of entities in several unit ontologies contain the version number of the ontology. The same issue affects some examples given by Kamdar et al. [2]. This will break the reuses of these concepts in case of updates and the disappearance of old versions. For example, in OM the IRI of the measurement unit class changed from `http://www.wurvoc.org/vocabularies/om-1.6/Unit_of_measure`[9] to `http://www.wurvoc.org/vocabularies/om-1.8/Unit_of_measure`. However, a new IRI should only be minted, if the associated definition has changed in a substantial way, to not mix up these two distinct resources. See Section 2.2 for a more detailed discussion.

There is one exception: By using versioned IRIs, the statements made within that particular version can be referenced. This allows the expression of meta-statements (statements about statements), for example, to describe the evolution of a certain model. However, this should not be addressed by including the version in the IRI, but by the *Version IRI* mechanism provided in OWL 2 [9]. Then ontologies that require a specific version of a resource are able to import the particular ontology version.

1b. IRIs should not be too long. We encountered IRIs whose local name included up to 185 characters[10]. We do not advocate a particular maximum length. However, one should keep in mind that, for example, three prefixed IRIs should fit into one line on a screen for uncluttered use in Turtle syntax or SPARQL.

1c. IRIs of large resource collections should not contain natural language. We encountered spelling errors in IRIs. For example, in MUO an "u" in the word "square" is missing for `http://purl.oclc.org/NET/muo/ucum/unit/pressure/pound-per-sqare-inch`. Fixing such errors requires IRIs to be changed. To preserve the integrity of references, dependent ontologies need to be updated as well or equivalence relations between deprecated IRIs and their correctly spelled IRIs need to be maintained. However, some communities use IRIs with a generic alpha-numerical local name to avoid a bias towards a particular language, such as Wikidata [10] and the OBO-foundry [11]. Entity names are represented only by associated literals. Consequently, name changes due to writing errors or the adoption of new naming conventions will not affect the IRIs themselves. This fosters ontology reuse, as the IRIs become more stable and might also ease the ontology maintenance.

Language independence might be a further benefit for ontology reuse, as mentioned by Fernández-López et al. [1]. However, it has also to be considered that numerical identifiers are slightly inconvenient in use and are harder to read without appropriate tool support. Similarly to the decision criteria for the use of hashes between namespaces and local names [8], we recommend to use natural language local names for "rather small and stable sets of resources" and to use generic names for large collections of resources.

---

[9] Meanwhile only available through web-archives: `https://web.archive.org/web/20130110021435/http://www.wurvoc.org/vocabularies/om-1.6/Unit_of_measure`.

[10] An IRI with a local name consisting of 185 characters (243 in total): `http://www.ontology-of-units-of-measure.org/resource/om-2/constantCurrentThatProducesAnAttractiveForceOf2e-7NewtonPerMetreOfLengthBetweenTwoStraightParallelConductorsOfInfiniteLengthAndNegligibleCircularCrossSectionPlacedOneMetreApartInAVacuum`

1d. Prefixes should not refer to multiple namespaces. We encountered cases of prefix re-mapping in different modules of an ontology[11]. For example, in SWEET the prefix `comp` was used in `reprSciComponent.owl` for `reprSciComponent.owl#`, but in `statePhysical.owl` for `matrCompound.owl#`. In theory, some RDF syntaxes like Turtle even allow prefix re-mapping in a single file [12]. However, prefix re-mapping might trigger mix-ups of namespaces during reuse and therefore cause wrong IRI references. Ontology engineers are encouraged to provide a consistent prefix-namespace mapping. While not all namespaces globally can be taken into account, one should at least strive for consistent use within a single ontology across all its modules and imported ontologies. We also refer to namespace lookup services[12] as a source for commonly accepted prefixes.

1e. Namespaces should not be referred by multiple prefixes. Similarly, we encountered cases of multiple prefixes for a single namespace[11]: An example is the namespace `relaSci.owl#` within SWEET. In `propOrdinal.owl` it is referred to by `screla`, but `propEnergyFlux.owl` uses `screla2` instead. This can cause the wrong assumption to deal with different namespaces and therefore cause incorrect use. Ontology engineers should globally determine the prefixes used in a modularized ontology. This will ease the reuse of the ontology as well as maintenance, as the meaning of ontology fragments does not depend on the containing file.

1f. Namespaces should not omit the hash. We encountered prefix mappings that omit the hash used in IRIs[11]: For example, SWEET maps the prefix `screla` to the namespace `relaSci.owl` instead of `relaSci.owl#` in its `propEnergyFlux.owl` module. Although this is permitted in XML-based ontology formats, it will cause problems in Turtle syntax or SPARQL. Here, the hash is reserved for comments. Omitting the hash might require a re-mapping or an additional definition of prefixes during reuse. This increases the risk to generate wrong IRIs.

2.2. Identity vs. Equivalency

OWL's *sameAs* relation is a crucial building block in ontology reuse and alignment. Its definition states that "two URI references actually refer to the same thing: the individuals have the same 'identity'" [13]. Consequently, both IRIs can be exchanged arbitrarily in all other contexts and, hence, all statements are equally applied to both of them. This allows individual knowledge graphs to mint their own IRIs while still connecting to the Linked Data Cloud at large [14].

Similarly, multiple names for a single entity are sometimes represented by separate IRIs. Instead, the reasoning result after a `owl:sameAs` connection can be materialized by attaching multiple labels to a single IRI. However, the consequences of erroneous mappings are the same as in using two separate IRIs and connecting them via *owl:sameAs*.

[11] Common part (`http://sweet.jpl.nasa.gov/2.3/...`) omitted for reader convenience.
[12] for example, `http://prefix.cc`

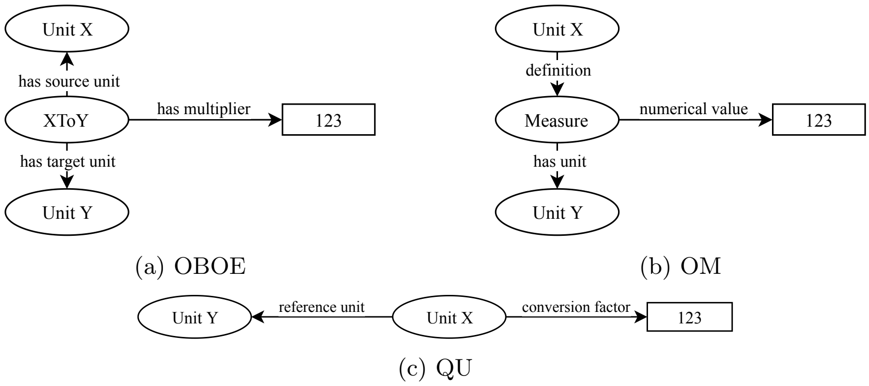

Figure 1. Simplified visualization of conversion models.

2a. Do not confuse equivalency and identity. In practice, the use of `owl:sameAs` differs oftentimes from this strict definition. Halpin et al. [15] mention four variations of weaker definitions used within Linked Data. For example, `owl:sameAs` was used to connect similar resources that "share some but not all properties".

In the context of unit ontologies, pairs of units with a conversion factor of one were sometimes connected using `owl:sameAs`. Although, they might be mathematical equivalent, they do not necessarily share the same semantic identity. The latter entails sharing all other properties. In case of units, this also includes the system of units. While, for example, *liter* and *cubic decimeter* are mathematically exchangeable, *liter* is not part of the SI system of units [16]. Thus, both units do not share the same identity.

2b. Be aware of alleged synonyms. At a first glance, units like *liter per square meter* seem overly redundant and could be expressed by, for example, *decimeter* instead. However, here numerator and denominator refer to particular quantities, for example, the amount of rain and the area it falls upon. Hence, a simplification leads to information loss and is strictly speaking not allowed, as they refer to different quantities. So both units represent different resources and have to have separate IRIs.

2c. Know the exceptions. There is at least one exception to the rule, that two units do not share their identity: *gon* and *grad*. Both labels are defined to denote the same unit [16]. So here, either two labels to the same resource or two resource IRIs connected via `owl:sameAs` are valid approaches to model this unit.

2.3. Properties

Relations between entities of different classes can usually be interpreted unambiguously. However, relations between entities of the same class sometimes leave room for misinterpretation, if the relation's semantics are not handled carefully. This is particularly evident for properties involving other values, such as conversions between units. Here, the modeling requires a relation from one unit to another with additional attributes like conversion factor and offset.

3a. Properties should be modeled resilient against misinterpretation. Within OBOE, conversions are modeled via separate classes whose local names follow the convention `XToY` (cf. Figure 1a). Yet, their interpretation is not consistent throughout the ontology. In the conversion `MicrometerToMeter` the factor f is given as 1 000 000 suggesting a formula like $1\,000\,000\,\mu\mathrm{m} = 1\,\mathrm{m}$ ($f \cdot X = Y$). However, the related conversion `DecimeterToMeter` provides a factor of 0.1 leading to an interpretation of $1\,\mathrm{dm} = 0.1\,\mathrm{m}$ ($X = f \cdot Y$). Although both conversions seem to be correct in isolation, the conversion factors' directions are inverse to one another. This reveals that even ontology authors themselves are susceptible to misinterpretations of their own model.

In contrast, OM models conversions as a measurement of one unit in terms of another (cf. Figure 1b). For example, an *international inch* is defined by a measurement of 0.0254 m. While both approaches are similar in structure, OM's semantics appear more robust against misinterpretation.

3b. Dependent properties should be encapsulated into distinct resources. Conversions in QU are modeled by two properties directly attached to the unit: `referenceUnit` and `conversionFactor` (cf. Figure 1c). This works as long as only one conversion should be defined per unit, but breaks in case of multiple conversion definitions. The dependency between the properties is not represented, and thus the individual conversions can not be retraced. To retain the dependence, the use of a distinct resource is required as done in OBOE (cf. Figure 1a) and OM (cf. Figure 1b).

3. Conclusion

We presented a collection of issues we encountered during the reuse of ontologies for measurement units. We discussed opportunities to avoid these issues by using alternative modeling approaches or avoiding anti-patterns. Many of these issues can be automatically checked during the creation of an ontology. Kamdar et al. already requested better tooling support for reusing other ontologies [2]. We extend that notion and also suggest to improve tool support for ontology engineers to boost the reusability of the ontologies they create.

However, not all issues can be automatically addressed. Especially, to verify the unambiguity of property directions, manual intervention is needed. Therefore, we suggest to add a manual reusability test to ontology creation workflows. Possible tasks include modeling a certain fact using the means provided by the ontology or creating queries for given information needs (for example, by competency questions). Similar to for usability testing in user interface design, these tasks should be performed by humans that were not involved in the development of the ontology under test. Feedback and achieved success or error rates can provide valuable insights into the reusability of the ontology. Regardless of those reusability tests, we hope to draw ontology engineers' attention to reuse problems in practice and thus enhance ontology reusability in the future.

Acknowledgments

Part of this work was funded by DFG in the scope of the LakeBase project within the Scientific Library Services and Information Systems (LIS) program. We thank the three anonymous reviewers for their helpful comments on an earlier draft of this manuscript.

References

[1] M. Fernández-López, M. Poveda-Villalón, M. C. Suárez-Figueroa, and A. Gómez-Pérez. Why are ontologies not reused across the same domain? *Journal of Web Semantics*, 2018. doi:10.1016/j.websem.2018.12.010.

[2] M. R. Kamdar, T. Tudorache, and M. A. Musen. A systematic analysis of term reuse and term overlap across biomedical ontologies. *Semantic Web*, 8(6):853–871, 2017. doi:10.3233/SW-160238.

[3] M. Poveda Villalón, A. Gómez Pérez, and M. C. Suárez Figueroa. OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. *International Journal on Semantic Web and Information Systems*, 10(2):7–34, 2014. doi:10.4018/ijswis.2014040102.

[4] J. M. Keil. LakeBase Semantic Service. In *ICEI 2018: 10th International Conference on Ecological Informatics*, 2018. doi:10.22032/dbt.37852.

[5] M. D. Steinberg, S. Schindler, and J. M. Keil. Use Cases and Suitability Metrics for Unit Ontologies. In *OWL: Experiences and Directions – Reasoner Evaluation. OWLED/ORE 2016*, pages 40–54, 2016. doi:10.1007/978-3-319-54627-8_4.

[6] J. M. Keil and S. Schindler. Comparison and evaluation of ontologies for units of measurement. *Semantic Web*, 10(1):33–51, 2019. doi:10.3233/SW-180310.

[7] T. Berners-Lee. Cool URIs don't change, 1998. URL: https://www.w3.org/Provider/Style/URI.

[8] Cool URIs for the Semantic Web, 2008. W3C Interest Group Note. URL: https://www.w3.org/TR/cooluris/.

[9] OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax (Second Edition), 2012. W3C Recommendation. URL: https://www.w3.org/TR/owl2-syntax/.

[10] D. Vrandečić and M. Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 2014. doi:10.1145/2629489.

[11] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. J. Goldberg, K. Eilbeck, A. Ireland, C. J. Mungall, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, R. H. Scheuermann, N. Shah, P. L. Whetzel, and S. Lewis. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology*, 25(11):1251–1255, 2007. doi:10.1038/nbt1346.

[12] RDF 1.1 Turtle: Terse RDF Triple Language, 2014. W3C Recommendation. URL: https://www.w3.org/TR/turtle/.

[13] Owl web ontology language reference, 2004. W3C Recommendation. URL: https://www.w3.org/TR/owl-ref/.

[14] W. Beek, J. Raad, J. Wielemaker, and F. van Harmelen. sameas.cc: The closure of 500m owl: sameas statements. In *The Semantic Web - 15th International Conference, ESWC 2018*, pages 65–80, 2018. doi:10.1007/978-3-319-93417-4_5.

[15] H. Halpin, P. J. Hayes, J. P. McCusker, D. L. McGuinness, and H. S. Thompson. When owl:sameAs isn't the same: An analysis of identity in linked data. In *The Semantic Web - ISWC 2010 - 9th International Semantic Web Conference, ISWC 2010*, pages 305–320, 2010. doi:10.1007/978-3-642-17746-0_20.

[16] International Bureau of Weights and Measures (BIPM). *The International System of Units (SI)*, 8 edition, 2014.