Towards Reusability in the Semantic Web: Decoupling Naming, Validation, and Reasoning

✉ Johannes Lipp[1,2], ✉ Lars Gleim[2], and Stefan Decker[1,2]

[1] Fraunhofer Institute for Applied Information Technology, Sankt Augustin, Germany
{johannes.lipp,stefan.decker}@fit.fraunhofer.de
[2] Chair of Databases and Information Systems, RWTH Aachen University, Germany
{lipp,gleim,decker}@dbis.rwth-aachen.de

Abstract. RDFS and OWL ontologies simultaneously define naming, hierarchy, syntactical data structure, and axioms. This strong coupling complicates the reusability of both ontological concepts and annotated data, due to logical pitfalls in RDFS and OWL semantics. The differences between OWL axioms and integrity constraints used for validation are often not clear to users and lead to confusing and unintended semantics in practice. To avoid these pitfalls, we revisit Tom Gruber's basic ontology definition and reimagine a more decoupled ontology design pattern, consisting of independent layers for *naming*, *validation*, and *reasoning*. We argue that such decoupling improves reusability because it clarifies the usage of the three layers during ontology creation and reuse. A naming layer built on synonym sets enables reusing named concepts in different contexts, detached from constraints or OWL axioms defined elsewhere. On top of that, we suggest a two-step approach of constraint checking and reasoning: Validate a term's integrity via constraints first, and only include it for reasoning if that validation succeeds. Our proposal is one step towards a clearer in-practice usage of naming, validation, and reasoning - and additionally supports this with a revised semantic layer model.

Keywords: Semantic Web Stack · Ontology Design Pattern · Naming · Validation · Reasoning · Integrity Constraints · Interoperability

1 Motivation

The Semantic Web and ontologies are well-defined and have been continuously improved over the last decades. A comprehensive collection of standards, recommendations, and ongoing drafts is available for multiple application areas. Following the open-world assumption, any new approach can add new concepts, and thereby eventually increase the complexity of ontologies. Nowadays, many common challenges do not reside in theoretical research anymore, but rather in practice when actually using the defined concepts. Ruben Verborgh in [8] discusses a widespread misjudgment, which states that after solving 80% of a research problem, the remaining 20% are practicalities and appear trivial to solve. These neglected "trivialities" however become serious problems, once they are lifted from controlled environments to the Semantic Web in practice.

One example is discussed in [2], where the authors state that *owl:sameAs* is often used incorrectly in practice. People are often not aware that it intentionally means that two individuals are *exactly* the same (e.g., including all properties), and thus create confusion. This does not only apply to that property but also for other types of equivalences between classes and properties. Note that the authors suggest four other properties to use instead. Additionally, alternatives with less conflict potential such as http://purl.org/dc/dcam/domainIncludes have been introduced, however, their practical adoption remains to be improved. Motik et al. document another issue in [6] as they have observed that OWL axioms are sometimes understood as integrity constraints. Although they were never meant to be interpreted like this, these observations, in reality, show that we need to improve clarity indeed. The Semantic Web community developed Shape Expressions (ShEx) [7] and the Shapes Constraint Language (SHACL) [4], which both ease integrity checking via constraints - but the confusion between validation and axioms remains in practice.

Gruber in 1993 defines an ontology as *"an explicit specification of a conceptualization"*, and subsequently a conceptualization as *"an abstract, simplified view of the world we wish to represent for some purpose"* [1]. Simplification w.r.t. some "purpose" is typically an essential part of limiting the scope of such a specification and for its alignment with a given set of requirements. It does, however, at the same time limit reusability of ontologies. Taking an ontology, which was designed for a certain purpose, and reusing it in a different context often leads to inaccuracies or contradictions, including the ones mentioned above. While the combination of upper-level and domain ontologies is a great approach, we argue that the current theoretical definitions are insufficient in practice, especially when mixing multiple ontologies or reusing individual concepts from ontologies with a similar domain.

```
b:Person a owl:Class .

b:hasAge a owl:DataTypeProperty ;
    rdfs:domain b:Person ;
    rdfs:range b:zeroToHundredAndTwelve .

b:hasName a owl:DataTypeProperty ;
    rdfs:domain b:Person ;
    rdfs:range xsd:string .

b:Person rdf:subClassOf [ rdf:type owl:Restriction ;
  owl:minCardinality  "1"^^xsd:nonNegativeInteger ;
  owl:onProperty   b:hasName ] .

b:zeroToHundredAndTwelve [...]
```

Listing 1.1. Naming and axioms defined altogether in a remote ontology. Definitions of prefixes and `b:zeroToHundredAndTwelve` are neglected to save space.

```
a:Bob b:hasAge 29 ; a b:Person .
a:Rex b:hasAge 4 .
```

Listing 1.2. Example instance in local ontology. Prefix definitions are neglected to save space.

The real-world example explained in the following demonstrates a misuse of axioms vs. constraints, which we observe in practice regularly. Listing 1.1 shows an ontology excerpt that models a person with properties for their age and name, respectively. Lines 11 through 13 additionally define an axiom, so that the minimum cardinality of a person's name is one. Furthermore, Listing 1.2 shows a user's local knowledge, which refers to that ontology. It adds an age to both Bob and Rex, while Bob is also explicitly marked as person. Common pitfalls due to human errors or misunderstandings in practice would now cover:

- Assuming that we must define a name via `b:hasName` for Bob right here, as the OWL axiom in Listing 1.1 tells so.
- Assuming Rex is not a person, because it is not explicitly stated so. A reasoner however would infer this, because of the domain of the used property `b:hasAge`.

Both mistakes occur in reality - not because of logical errors in OWL or RDFS, but because of human errors when using these. The actual problem is that, as stated in [6], in-practice usage often means constrains instead of axioms. Therefore, the user in this example should rather create SHACL shapes as shown in Listing 1.3 to properly express their intuitive intention.

```
c:PersonShape a sh:NodeShape ;
    sh:targetClass b:Person;
    sh:property [ sh:path b:hasAge ;
        sh:minInclusive 0 ;
        sh:maxInclusive 112 ] ;
    sh:property [ sh:path b:hasName ;
        sh:minCount 1 ;
        sh:maxCount 1 ] .
```

Listing 1.3. SHACL shape used for validation of `b:Person` instances. Prefix definitions are neglected to save space.

Our research tackles the above-mentioned issues by splitting the three concepts *naming*, *validation*, and *reasoning*, which are currently jointly defined in ontologies, into separate layers to improve their usage in practice. The way how we use ontologies depends on their purpose, which usually includes knowledge linking, sharing, validation, or reasoning. In this paper, we propose a revised Semantic Web layer model that separates the layers *naming*, *validation*, and *reasoning*, and thus supports proper usage in practice. Section 3 subsequently discusses strengths, weaknesses, opportunities and threats of our approach in a SWOT analysis, before we conclude our work.

2 Global Naming, Shared Validation, and Individual Reasoning

This section first defines the concepts naming, validation, and reasoning, and analyzes their individual level of reusability. We then present in Figure 1 a revised Semantic Web layer model that decouples these three to improve reusability in practice.

Naming. In this paper, we use a naming layer as a foundation. Elements in this layer can be compared to so-called synonym sets (synsets) in WordNet [5]. This layer uses a controlled vocabulary, which itself adds labels and textual definitions to persistent identifiers (PIDs), and groups terms into synsets. Please note that the synset grouping is sufficient for this layer, and we do not require any hierarchy or order of terms. Synsets usually contain multiple terms from the underlying controlled vocabulary, and a term can be part of multiple synsets.

Validation. Often used for integrity checking, the validation layer provides constraints that can be used to validate data. As defined in the above-mentioned SHACL and ShEx, this can include cardinalities, domain and range of properties, string and logical constraints, and other conditions.

Reasoning. The reasoning layer follows existing reasoning such as OWL and provides axioms as well as reasoners to apply these to present terms that are interpreted as facts. In further research towards implementation, we need to clarify that this layer must not be confused with the validation layer anymore.

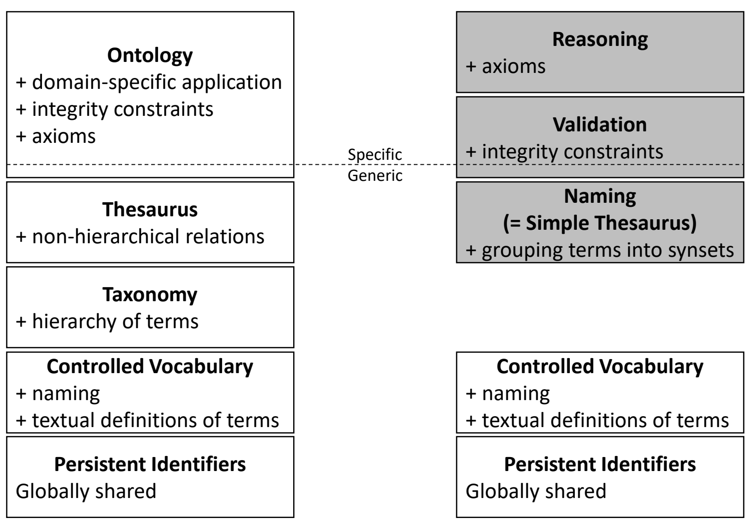

Fig. 1. We propose a revised Semantic Web layer model that does not merge naming, validation, and reasoning into ontologies, but makes them available independently. The current model is shown on the left, while our revised version is shown on the right, with particular updates marked in gray. Note that the lower layers are in general meant to be generic, while the upper layer is more specific. As we split the three layers, we support are more clear differentiation during reuse.

The overall structure of our revised Semantic Web layer model is depicted in Figure 1. It shows the common layer stack on the left, which bases on PIDs that are named and defined in controlled vocabularies, which again are structured hierarchically in taxonomies. A thesaurus adds additional non-hierarchical relations, but still mainly focuses on a generic usage, as indicated by the dotted line. An ontology uses terms from thesauri for a particular domain, and adds axioms (e.g., OWL) and constraints (e.g., SHACL or ShEx). The main issue we explained in Section 1 is that we cross the line between generic and specific usage, which is a crucial issue in combination with the misunderstanding in practice described in that section.

Therefore, we introduce the revised layer model on the right side of Figure 1, which consists of the three main layers *naming*, *validation*, and *reasoning*. While the naming layer reuses terms from a controlled vocabulary, both a hierarchy and further non-hierarchical terms are not required - but multiple terms can be grouped into synsets to ease reusing these. This layer is intended to be generic and we suggest a central repository to organize and provide these synsets, such as Wikidata [9] or similar knowledge bases. A community-driven approach could improve evolution.

The validation layer builds on this naming layer, and adds constraints to validate terms from the naming layer as well as data annotated with these. Note that this layer is on the edge between generic and specific usage, as the validation of a given synset instance (e.g., an email address) might not be exactly the same for all applications, but they might share some consensus at least. On a technical implementation, we suggest for this layer also a repository, similar to how software packages are managed in repositories for usage with package managers. This would map each synset to one or more proposed sets of validation constraints, that can be reused as-is or adopted for specific applications. Note that this layer might intersect with the usage of *application profiles*.

The third layer, namely reasoning, serves as that one in OWL ontologies. Its contents are described above, and we suggest the following implementation in order to clarify the difference to the validation layer. Whenever we instantiate synsets for the naming layer, we should enforce the definition and application of constraints from the validation layer, before we allow any reasoning. That is, we define synsets with an open-world assumption on the naming layer, validate these in our in-practice application with closed-world constraints from the validation layer, and *then* allow open-world reasoning based on axioms from that respective layer. Since this process is bound to a specific domain, we do not suggest generally sharing all axioms globally, but precisely choosing which ones to reuse in an actual domain-specific use-case.

Our proposed approach and revised layer model ease correct in-practice usage of naming, validation, and reasoning.

3 Discussion and Outlook

To dissect strengths, weaknesses, opportunities, and threats of the proposed approach, we analyze it using the SWOT methodology [3], as depicted in Table 1.

The strengths include term (synset) reusability from elsewhere without the burden of accidentally importing validation constraints or OWL axioms that one would not like to apply for their specific domain, the support for existing validation constraints formulated via SHACL or ShEx, and the easier correct reasoning due

Strengths	**Weaknesses**
– Reusing terms without burden – Supports existing validation constraints – Eases correct reasoning	– Backward compatibility should be given but needs to be clarified – Missing tool support for a combination of first validation, then reasoning – Still need to define partial constraint reuse
Opportunities	**Threats**
– Quick drafts without the need to fully understand all relevant ontologies – Simple, dedicated tool support – Community-driven validation constraint development	– Low acceptance in the community, which mostly focuses on well-known standards

Table 1. SWOT analysis for our proposed approach.

to decoupling. The weaknesses include a currently unclear backward compatibility, especially for the selective definition and reuse of open-world axioms. Also, tool support for our proposed combination of "first validation, then reasoning" need is not yet available, plus we need to extend on a proper definition of how to select validation constraints in reality. The opportunities, in case our approach gets broadly adopted, include the possibility of creating quick drafts by only reusing particular layers, without the need to fully understand all three layers defined elsewhere. This would enable simple tool support as well and would foster community-driven constraint development for all available synsets. Finally, we identify a possible low acceptance in the Semantic Web community as a threat, because it is focused on established standards and the challenges that are mostly solved already - in theory, but not in practice.

In this paper, we identified the strong coupling of naming, validation, and reasoning in ontologies as crucial pitfalls when sharing and reusing these. We explained in the motivation that the concepts, which are very well defined in theory, are often not applied correctly as in practice. Our revised Semantic Web layer model decouples these three and suggests to share synsets globally in a community-based repository. Validation constraints could also be developed in a collaborative repository, but there might be multiple alternatives per synset that can be adopted to specific applications. In the reasoning layer, we only apply axioms to answer questions or derive new knowledge, after we validated data against validation constraints.

Since there is a strong discrepancy between good theoretical definitions and actual use in practice, we believe that our approach is an important step towards easier reuse in the Semantic Web. Future research includes further specifying actual implementation suggestions and tackling the weaknesses and threats mentioned in the SWOT analysis above.

Acknowledgements. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612.

References

1. Gruber, T.R., et al.: A Translation Approach to Portable Ontology Specifications. Knowledge acquisition **5**(2), 199–220 (1993)
2. Halpin, H., Hayes, P.J., McCusker, J.P., McGuinness, D.L., Thompson, H.S.: When owl:sameAs Isn't the Same: An Analysis of Identity in Linked Data. In: International Semantic Web Conference. pp. 305–320. Springer (2010)
3. Humphrey, A.: SWOT Analysis for Management Consulting. SRI alumni Newsletter **1**, 7–8 (2005)
4. Knublauch, H., Kontokostas, D.: Shapes Constraint Language (SHACL). W3C Candidate Recommendation **11**(8) (2017)
5. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM **38**(11), 39–41 (1995)
6. Motik, B., Horrocks, I., Sattler, U.: Adding Integrity Constraints to OWL. In: OWLED. vol. 258 (2007)
7. Prud'hommeaux, E., Labra Gayo, J.E., Solbrig, H.: Shape Expressions: An RDF Validation and Transformation Language. In: SEMANTICS. pp. 32–40 (2014)
8. Verborgh, R., Vander Sande, M.: The Semantic Web Identity Crisis: In Search of the Trivialities That Never Were. Semantic Web Journal **11**(1), 19–27 (Jan 2020). https://doi.org/10.3233/SW-190372, https://ruben.verborgh.org/articles/the-semantic-web-identity-crisis/
9. Vrandečić, D., Krötzsch, M.: Wikidata: A Free Collaborative Knowledgebase. Communications of the ACM **57**(10), 78–85 (2014)