

Using Ontology Design Patterns To Define SHACL Shapes

Harshvardhan J. Pandit, Declan O’Sullivan, and Dave Lewis

ADAPT Centre, Trinity College Dublin, Dublin, Ireland
{harshvardhan.pandit|declan.osullivan|dave.lewis}@adaptcentre.ie

Abstract. SHACL shapes used for validation are not related to the axioms used in ontologies used to define the instances. One of the issues with using such axioms is their dependence on concepts which may not be used within the data graph, as well as the presence of other concepts and relationships which the axioms do not cover. By contrast, an Ontology Design Pattern (ODP) contains axioms that are closer to how its instances are defined. In this position paper, we discuss the reuse of ODP axioms from modelling contexts to defining SHACL shapes, using the use-case of MicroblogEntry ODP. The aim of this approach is to foster automated generation of SHACL shapes based on contexts, represented by an ODP, defined within data graphs.

Keywords: Ontology Design Pattern, RDF Validation, SHACL

1 Introduction

The Shapes Constraint Language (SHACL)¹ is a language for describing and validating RDF graphs, and is a W3C recommendation since July 2017. The set of constraints used by SHACL for validation are expressed as an RDF graph and are called ‘shapes’ or ‘shape graphs’ and the RDF data being validated is called the ‘data graph’. Shapes offer a description of the data graph in the form of constraints that a valid data graph satisfies. This is based on the closed-world assumption where all the required information is assumed to be present in the correct format or is considered to be invalid. This allows shapes to be used for other purposes such as code generation and data integration¹.

Creation of the validation conditions, called shapes in SHACL, is invariably tied to the data within the graph, and is therefore dependant on the ontologies used to model them. Such ontologies are modelled using axioms which serve to provide constraints over the use of the ontology and can therefore also be used to validate data. However, such axioms are only scoped to the ontology they are defined in, and therefore may not relate to other ontologies used within the RDF graph. Additionally, the RDF graph may only use selective concepts and properties from multiple ontologies. Such selective use may not be verifiable using axioms, which could depend on concepts and properties not used within

¹ <https://www.w3.org/TR/shacl/>

the graph. Therefore, there is no reuse within the activities of modelling ontology axioms, choosing ontologies for use in data graphs, and creating SHACL shapes to validate the data graph.

By contrast, an Ontology Design Pattern (ODP) captures only the concepts and relationships necessary to define a particular context. Such ODPs can combine concepts and relationships from multiple ontologies to express new relationships between them. Since an ODP is smaller and more modular than a comparatively larger ontology, the coverage of its terms used is larger in instances based on it. Its axioms, therefore, are more suitable for validation, and can be related to constraints in SHACL shapes.

In this position paper, we discuss this similarity between the axioms used to model ODPs and the constraints within SHACL shapes. We present our argument in using ODP axioms for generating SHACL shapes for validation of instances based on the pattern. Our aim is to investigate the automation of SHACL shape generation from modular patterns for a given data graph. This paper also serves to encourage the reuse of ODPs outside of modelling ontologies for validating RDF graphs. The approach discussed in this paper is applicable only to those graphs which use a selective set from multiple ontologies.

The rest of the paper is structured as follows: Section 2 discusses the use of ODP axioms to generate SHACL shapes, with an example provided in Section 3. Section 4 concludes this paper.

2 ODP axioms and SHACL shapes

An axiom is defined within description logic as a logical statement relating roles and/or concepts [3]. Axioms in an ontology define constraints over concepts and relationships that must be satisfied by the instances that use the ontology. These axioms cannot be reused as part of an ODP as this can cause issues with missing entities (dependencies) which are not part of the ODP. Instead, the ODP defines its own set of axioms that are limited to only those concepts and relationships that are a part of it. In that sense, these axioms operate in a similar fashion to the constraints within SHACL, which are based on the closed-world assumption.

Existing work comparing OWL axioms and SHACL [4] finds the expressivity of OWL being comparable to the SHACL Core vocabulary, and that syntactic translation between OWL and SHACL is straight-forward in most cases. Automating this process would involve two steps - the first to identify the relevant OWL statements forming a single constraint, and the second to then generate their equivalent SHACL shape constraints. Since both OWL and SHACL are essentially defined using RDF triples, both steps can be performed programmatically using the table of associated concepts mapping OWL and SHACL constraints [4].

3 Example

The MicroBlog ODP [5] is based on real-world use-cases for modelling data related to tweets (Twitter posts). Its core class, *MicroblogEntry*, defines three axioms describing constraints and relationships within the ODP, which are:

1. $\text{MicroblogEntry} \sqsubseteq \forall = 1 \text{hasPayload.Payload}$
2. $\text{MicroblogEntry} \sqsubseteq \forall = 1 \text{hasAuthor.Author}$
3. $\text{MicroblogEntry} \sqsubseteq \forall \leq 1 \text{writtenAt.Location}$

These are defined² using *rdfs:subClassOf* and *owl:Restriction* as:

```
1 ### http://www.example.org/dase/MicroblogEntry#MicroblogEntry
2 :MicroblogEntry rdf:type owl:Class ;
3   rdfs:subClassOf :ReportingEvent ,
4     [ rdf:type owl:Restriction ;
5       owl:onProperty :hasPayload ;
6       owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
7       owl:onClass :Payload
8     ] ,
9     [ rdf:type owl:Restriction ;
10      owl:onProperty :hasAuthor ;
11      owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
12      owl:onClass :Author
13    ] ,
14     [ rdf:type owl:Restriction ;
15      owl:onProperty :writtenAt ;
16      owl:maxQualifiedCardinality "1"^^xsd:nonNegativeInteger ;
17      owl:onClass :Location
18    ] .
```

These axioms can be used to directly generate the corresponding constraints in a SHACL shape using *sh:class* and *sh:qualified(Max/Min)Count* conditions. An example of this is the following SHACL shape:

```
1 :MicroblogEntryShape
2   a sh:NodeShape ;
3   sh:targetClass :MicroblogEntry ;
4
5   sh:property [
6     sh:path :hasPayload ;
7     sh:class :Payload ;
8     sh:MinCount 1;
9     sh:MaxCount 1;
10  ] ,
```

² <https://raw.githubusercontent.com/cogan-shimizu-wsu/MicroblogEntryOWL/master/MicroblogEntry.owl>

```
11     sh:property [
12         sh:path :hasAuthor ;
13         sh:class :Author ;
14         sh:MinCount 1;
15         sh:MaxCount 1;
16     ] .
```

While the SHACL shape defines constraints for two axioms, the third axioms defines an optional triple, which does not provide a constraint on the data, and hence is not part of the shape. It can be represented as a constraint with a maximum of one to specify none or at most one location.

4 Conclusion

Through this position paper, we presented our arguments towards the use of ontology design patterns (ODPs) to generate SHACL shapes. The approach uses axioms defined within ODPs to generate equivalent SHACL shape constraints for data validation over a RDF dataset using those ODPs. The paper provides an example of this, where the RDF triples representing the ODP axioms within an OWL file are used to generate their corresponding SHACL shape. The paper also discusses the possibility of automating the SHACL shape generation process.

The paper emphasises the use of ODPs rather than ontologies as the basis of validation. This is because RDF ontologies may contain multiple ontologies, where the axioms in any ontology may not be sufficient for verification of their instances in the data graph. ODPs can capture such use-cases due to their smaller and modular structure that allows representing axioms over multiple ontologies in different ways to express varying contexts.

This approach encourages the reuse of ODPs beyond the data modelling phase. Relating such ODPs with their corresponding SHACL shapes provides a way to visualise the model of the data as well as to validate it using the same context. The ODPs defined in this manner are modelled more closer to the instances used in actual RDF graphs, and can therefore be used in approaches such as data summarising, visualisation, and exploration. Conversely, ODPs can assist approaches relevant to validation such as visualising SHACL shapes. This can be done by taking SHACL shapes and generating corresponding ODPs to represent their context.

In terms of future work, the approach discussed in this position paper needs to be validated in terms of mappings between (OWL) axioms and SHACL shape constraints. In addition, the paper only considers the SHACL core vocabulary, and needs an investigation of the features provided by SHACL advanced and SHACL-SPARQL. The ability to convert OWL axioms to SPARQL queries using approaches such as OWL2SPARQL³ would allow the generation of SHACL shapes from ODPs by using the SHACL-SPARQL features. This could also potentially assist in dealing with recursive constraints based on existing methods

³ <https://github.com/SmartDataAnalytics/OWL2SPARQL>

[1,2]. Based on these, an implementation of a proof-of-concept model needs to be created to demonstrate the feasibility of the approach. Anti-patterns that increase complexity for generation of SHACL shapes also need to be investigated.

Our intended application is to automate the generation of SHACL shapes from existing patterns/models describing the data. This will allow us to validate a data graph based on specific contexts (represented through ODPs), and reusing the same validation mechanisms to check for existence and correctness of required data in the graph. The reporting feature of SHACL would then be used to produce a documentation based on the outcome of the validation to describe the quality of data.

Note: The paper, “Learning SHACL Constraints for Validation of Relation Assertions in Knowledge Graphs” by André Melo and Heiko Paulheim, presents work of relevance to this paper regarding generating SPARQL queries from OWL axioms and using SHACL-SPARQL to use them as constraints. It was submitted⁴ to ESWC 2018, but was not accepted.

Acknowledgements

This work is supported by the ADAPT Centre for Digital Content Technology which is funded under the SFI Research Centres Programme (Grant 13/RC/2106) and is co-funded under the European Regional Development Fund. We also wish to thank Heiko Paulheim and Sebastian Rudolf for their discussion and guidance in the comparison of axioms and SHACL constraints.

References

1. Corman, J., Reutter, J.L., Savkovic, O.: KRDB18-01.pdf. Technical Report KRDB18-01 (Apr 2018), <https://www.inf.unibz.it/krdb/KRDB%20files/tech-reports/KRDB18-01.pdf>
2. Corman, J., Reutter, J.L., Savković, O.: Towards a Robust Semantics for SHACL: Preliminary Discussion. In: Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management. Cali, Colombia (2018), <http://ceur-ws.org/Vol-2100/paper22.pdf>
3. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. *Web Semantics: Science, Services and Agents on the World Wide Web* 6(4), 309–322 (Nov 2008), <http://www.sciencedirect.com/science/article/pii/S1570826808000413>
4. Knublauch, H.: SHACL and OWL Compared (2017-08-07), <http://spinrdf.org/shacl-and-owl.html>
5. Shimizu, C., Cheatham, M.: An Ontology Design Pattern for Microblog Entries. In: Proceedings of the 8th Workshop on Ontology Design and Patterns (WOP 2017) co-located with the 16th International Semantic Web Conference (ISWC 2017) (2017), <http://ceur-ws.org/Vol-2043/paper-06.pdf>

⁴ https://2018.eswc-conferences.org/paper_151/