

RDFShape: An RDF playground based on Shapes

Jose Emilio Labra-Gayo¹, Daniel Fernández-Álvarez¹, and Herminio García-González¹

University of Oviedo
Spain

labra@uniovi.es danifdezalvarez@gmail.com herminiogg@gmail.com

Abstract. The quality of RDF based solutions requires the possibility to describe and validate RDF graphs. In the recent years, two technologies have been proposed for RDF validation: ShEx and SHACL. RDFShape is a web service that can be used as an RDF playground to validate RDF with those technologies, but can also be used as a general RDF playground allowing to parse, convert, visualize and query RDF data. The validation engine supports both ShEx and SHACL, allows the conversion between different schema formats, and is able visualize schemas using UML-like class diagrams. The system offers the possibility to play with different RDF technologies in a unified setting and has already been used to teach RDF technologies in several courses and tutorials. It has also been employed in industrial RDF workflows that needed a web service for RDF data validation and management.

Keywords: RDF, ShEx, SHACL, Validation, SPARQL, Visualization, UML

1 Introduction

The adoption of any technology depends on the available tools. Although RDF is not new, there is a lack of integrated tools that can be used for the different aspects of its management, like RDF parsing, conversion, visualization, validation, querying and inferencing. RDFShape was initially created as an online demo to play with the recently created Shape Expressions language in 2014[1]. Once the W3C Data Shapes Working Group started to develop SHACL [2], RDFShape added also a SHACL validator as well as other features like SPARQL querying, RDF visualization, ShEx/SHACL format conversions, endpoint based validation and querying, UML visualizations of Schemas, etc. We consider that RDFShape offers an useful tool for RDF practitioners, both for educational purposes as well as for more industrial applications.

2 RDFShape features

The RDFShape landing page contains several menu options, the most important of which are:

- **RDF Data.** This pane offers information about RDF data. It has an option to parse and show information about RDF. For educational purposes, we added the possibility of visualizing small RDF graphs converting the graphs to SVG using Graphviz. This url: <http://rdfshape.weso.es/links/i1> shows the result of visualizing a simple RDF graph inspired by the one presented in the RDF 1.1 primer (<https://www.w3.org/TR/rdf11-primer/>). RDFShape also allows to convert between different RDF formats. It handles the common RDF formats supported by the underlying RDF library: Turtle, N-Triples, RDF/XML, JSON-LD, etc. As an example, the result of converting the previous RDF file to JSON-LD is shown in <http://rdfshape.weso.es/links/i2>. The service has a *permalink* option which allows to generate permanent URLs that can be copy-pasted and shared.
- **Schema.** This option can be used to obtain information about ShEx and SHACL schema shapes. It allows to convert between the different schema formats available and to visualize them. Both ShEx and SHACL support the traditional RDF formats Turtle, N-Triples, JSON-LD, etc., while ShEx also supports a compact syntax. Link <http://rdfshape.weso.es/links/i3> shows a ShEx schema converted from ShEx compact syntax to its turtle serialization. It is also possible to visualize ShEx shapes as UML-like class diagrams. The system transforms the schema shapes to UML classes using PlantUML. Those UML class diagrams are later converted to SVG for visualization. Fig. 2 shows an example of a ShEx Schema and its visualization (<http://rdfshape.weso.es/links/i4>)

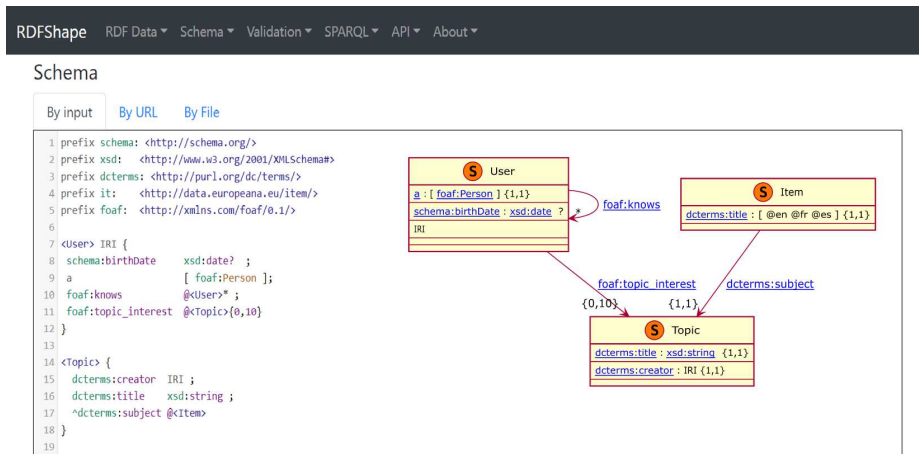


Fig. 1. Example of ShEx schema and its UML-like visualization

- **Validation.** RDFShape supports both ShEx and SHACL. As an example, <http://rdfshape.weso.es/links/i5> shows the result of validating an RDF graph using ShEx, while <http://rdfshape.weso.es/links/i6> shows

the result of SHACL validation report. In the case of SHACL, it shows information about the errors and evidences of the different results. This information is useful to distinguish between nodes that are not signalled as problematic because they have been validated to conform to a shape and nodes that are not signalled just because the validator does not reach them. It is also possible to validate against an SPARQL endpoint. For example, the result of validating a simple Schema against the Wikidata SPARQL endpoint can be seen at <http://rdfshape.weso.es/links/i7>. It is also possible to enable inference over the RDF graphs before querying or validating them. As an example, the following link shows the results of validating a ShEx schema over an RDF graph after applying RDFS inference on it: <http://rdfshape.weso.es/links/i8>.

- **SPARQL.** RDFShape can also be used to run SPARQL queries over RDF data. Although there are several online SPARQL tools like YASGUI ¹, we consider this feature to be very handy to query the same RDF graphs that are being validated, specially in the case of SPARQL-based constraints in SHACL. Link <http://rdfshape.weso.es/links/i9> shows the result of a SPARQL query over the previous RDF graph.
- **API.** RDFShape provides a REST API through which it is possible to invoke the previous actions in a programmatic way.

3 RDFShape components

RDFShape is available as open source software using the MIT licence at a github repository ². It has been designed with the goal of being flexible enough to work with different RDF libraries like Apache Jena or RDF4j. The main components are depicted in figure 2.

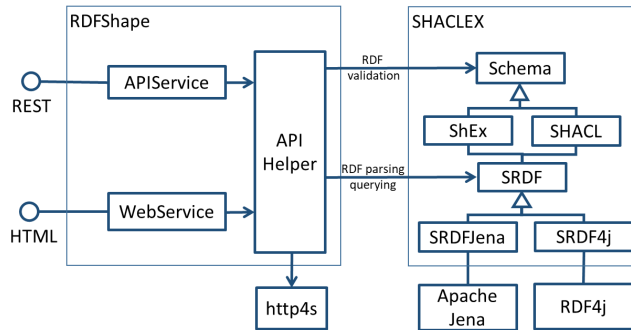


Fig. 2. System architecture

¹ <http://yasgui.org/>

² <https://github.com/labra/rdfshape>

- *APIService* and *WebService* provide the external interfaces which offer both a REST API and an HTML frontend. These two services call different methods from that *APIHelper* module which is based on *http4s*³, a simple http framework using a purely functional programming approach that offers support for streaming. For validation, it relies on the *ShaclEx* library, while for RDF parsing and querying, it relies on *SRDF*.
- The *ShaclEx* library has two modules that contain the ShEx and SHACL implementation. Both implementations use a purely functional approach based on monad transformers provided by the Scala Cats library⁴.
- In order to work with different underlying implementations, all RDF processing is done through a *SRDF* (*Simple RDF*), a generic RDF interface which defined common methods to manipulate RDF data. It has 3 sub-interfaces: *RDFReader* for querying RDF without modifying it, *RDFBuilder* to update and modify RDF graphs, and *RDFReasoner* to add inference capabilities to RDF graphs.
- *SRDF4Jena* is an implementation of *SRDF* for Apache Jena⁵ while *SRDF4J* is an implementation of *SRDF* for RDF4J⁶.

RDFShape has already been used in several semantic web courses and tutorials⁷, and to test the examples from the *Validating RDF data* book [3]. It has also been employed in more industrial settings to validate the ShEx schemas from wikidata (<http://rdfshape.weso.es/links/i10>) and HL7 FHIR (<http://rdfshape.weso.es/links/i11>). As future work, we are planning to incorporate the possibility to validate RDF datasets with named graphs and to adapt the RDF library for RDF Stream processing workflows.

Acknowledgements: This work is partially funded by the Spanish Ministry of Economy and Competitiveness (Society challenges: TIN2017-88877-R)

References

1. Prud'hommeaux, E., Labra Gayo, J.E., Solbrig, H.: Shape expressions: an RDF validation and transformation language. In: Proceedings of the 10th International Conference on Semantic Systems, SEMANTICS 2014, ACM (2014) 32–40
2. Knublauch, H., Kontokostas, D.: Shapes Constraint Language (SHACL). W3C Proposed Recommendation (June 2017)
3. Labra Gayo, J.E., Prud'hommeaux, E., Boneva, I., Kontokostas, D.: Validating RDF Data. Volume 7 of Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool Publishers LLC (sep 2017)

³ <https://http4s.org/>

⁴ <https://typelevel.org/cats/>

⁵ <https://jena.apache.org/>

⁶ <http://rdf4j.org/>

⁷ <http://www.validatingrdf.com/tutorial/iswc2018/>