Going Meta #3: Controlling the shape of your graph with SHACL

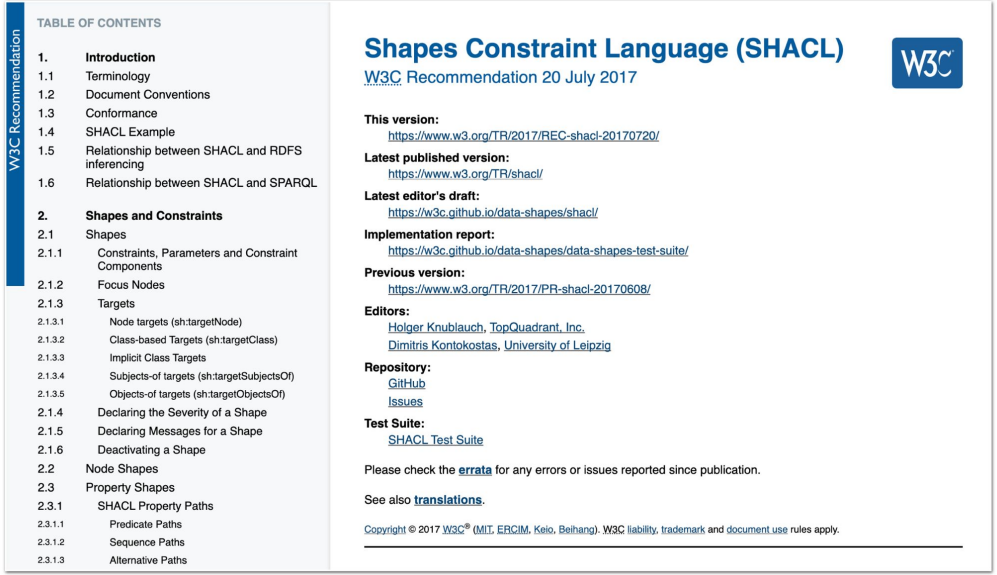
What's the problem?

We want to report on (sometimes enforce) the structure of our graph without having to write a series of custom cypher queries.
Examples:

- *The phone number property in a node of type Customer has to follow this regular expression.*
- *A Product node needs to be connected to at least three suppliers.*
- *A Supplier must have one and only one name.*
- *An order node cannot have properties outside this list.*

What's SHACL?

The SHACL Shapes Constraint Language is a language for validating RDF graphs against a set of conditions.



W3C Recommendation

TABLE OF CONTENTS

Shapes Constraint Language (SHACL)

W3C Recommendation 20 July 2017

This version:
https://www.w3.org/TR/2017/REC-shacl-20170720/

Latest published version:
https://www.w3.org/TR/shacl/

Latest editor's draft:
https://w3c.github.io/data-shapes/shacl/

Implementation report:
https://w3c.github.io/data-shapes/data-shapes-test-suite/

Previous version:
https://www.w3.org/TR/2017/PR-shacl-20170608/

Editors:
Holger Knublauch, TopQuadrant, Inc.
Dimitris Kontokostas, University of Leipzig

Repository:
GitHub
Issues

Test Suite:
SHACL Test Suite

Please check the **errata** for any errors or issues reported since publication.

See also **translations**.

`https://www.w3.org/TR/shacl/`

How do I use SHACL with my Neo4j Graph?

```
@prefix ex: <http://example.neo4j.com/graphvalidation#> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix neo4j: <neo4j://graph.schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

Namespaces are important in RDF

```
ex:ProdShape a sh:NodeShape ;
  sh:targetClass neo4j:Product ;
```

We are defining the shape of nodes labelled as **Product**

```
sh:property [
  sh:path neo4j:productName ;
  sh:pattern "^\\w[\\s\\w\\.]*$" ;
  sh:maxCount 1 ;
  sh:datatype xsd:string ;
];
```

The property **productName** is optional (max card but no min), is of type string and must match this regular expression...

```
sh:property [
  sh:path neo4j:unitPrice ;
  sh:minExclusive 10 ;
  sh:maxInclusive 100 ;
  sh:maxCount 1 ;
];
```

...

```
sh:property [
  sh:path neo4j:supplied_by ;
  sh:class neo4j:Supplier ;
  sh:minCount 2 ;
];
```

The relationship **supplied_by** has a minimum cardinality of 2 and must point to a node of type Supplier

Let’s do it!

What else?

- Enforce your constraints: Use SHACL transactionally
- Keep my graph aligned with an ontology (FIBO, schema.org,...)
  - You'll need to translate OWL -> SHACL
- Try this session at home:
  - https://github.com/jbarrasa/goingmeta
- What's coming up?
  - Node shape granularity
  - Export an RDF serialization of a node shape (for external use)
  - SHACL skeleton generation from your data importer model
  - UI for editing shapes