

Shapes Constraint Language Masterclass

By Veronika Heimsbakk

Shapes Constraint Language Masterclass



Veronika Heimsbakk

Managing AI Engineer | SME Semantic Technologies

veronika.heimsbakk@capgemini.com

Capgemini 

 vheimsbakk

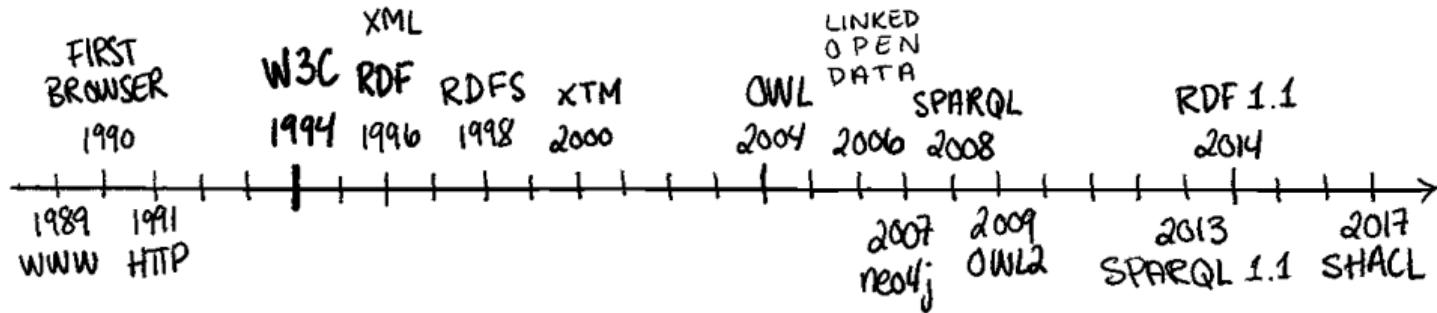
 veleda

 veronikaheim

 veronahe.no

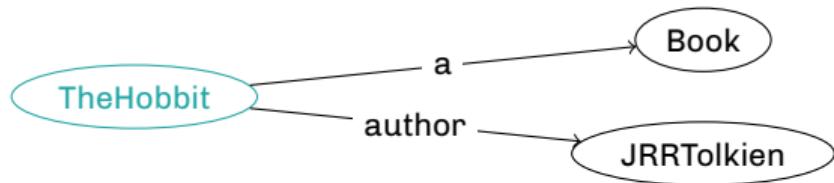
Once upon a time...

TIMELINE OF GRAPH ON THE WEB



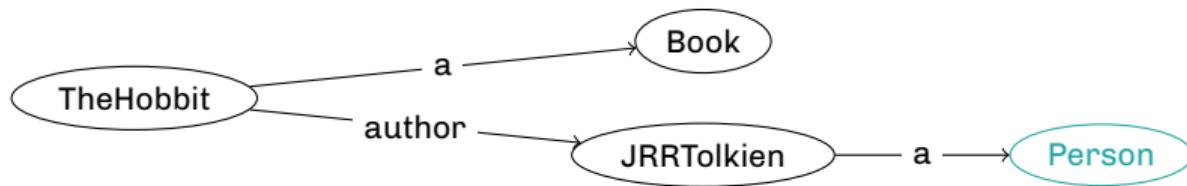
domain & range

domain



```
:author a rdf:Property ;  
rdfs:domain :Book .
```

range



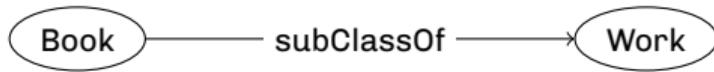
```
:author a rdf:Property ;  
    rdfs:domain :Book ;  
    rdfs:range :Person .
```

TBox & ABox

TBox

TBox (terminological component)

- › Sets of individuals.
- › Classes and concepts.



ABox

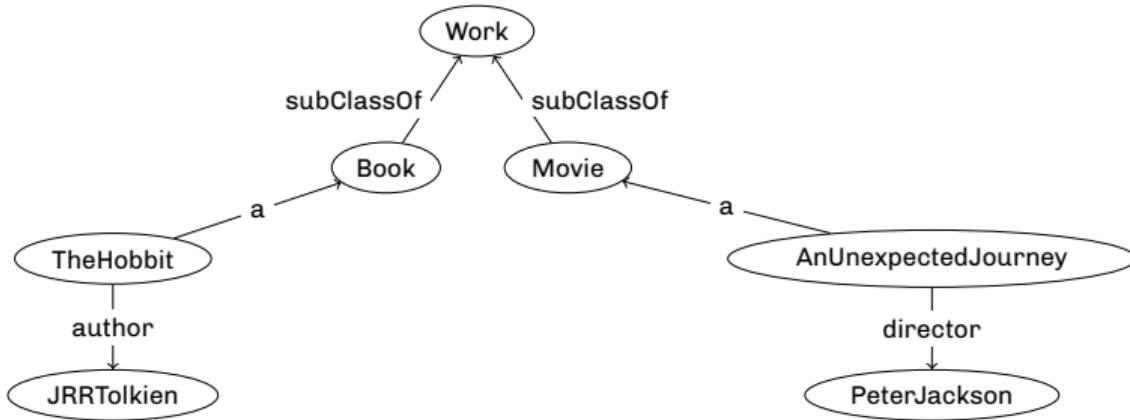
ABox (assertion component)

- › Individuals and instances.



Knowledge graph

TBox + ABox = Knowledge graph ❤



World assumptions

Open world assumption (OWA)

- > Admits incomplete knowledge.
- > Ontologies with Web Ontology Language (OWL).

The assumption that the truth value of a statement may be true irrespective of whether or not it is known to be true.



Example

Statement: In a **hole in the ground** there lived a **hobbit**.

Question: Do **Gandalf** live in a **hole in the ground**?

OWA: Unknown

Closed world assumption (CWA)

- › Shape constraints with Shape Constraint Language (SHACL).

Any statement that is true is known to be true. What is not currently known to be true is false.



Example

Statement: In a **hole in the ground** there lived a **hobbit**.

Question: Do **Gandalf** live in a **hole in the ground**?

CWA: No

Shape Constraint Language

A language for describing and validating RDF graphs

Validation of RDF – a brief history

- › Prior to SHACL; no W3C standard for validating RDF.
- › SPARQL Inferencing Notation (SPIN), IBM Resource Shapes, Shape Expressions (ShEx)
- › W3C recommendation in July 2017.

SPIN

- › Both are backed by SPARQL.
- › SHACL Constraint Components are more flexible than SPIN Templates due to the possibility of combining multiple constraint types into same shape definition.
- › SPIN Templates require new instances and multiple `spin:constraint` triples.

ShEx

- › Both have shapes on nodes and properties.
- › ShEx intends to be a grammar or schema for RDF.
- › SHACL aims to provide a constraint language for RDF.
- › ShEx returns an annotated data graph.
- › SHACL return a validation report as RDF.

SHACL & OWL

Common

- › RDF & URIs
- › Rely on RDF Schema (RDFS)

Difference



Inference Validation

When to use SHACL?

It depends on the use case!

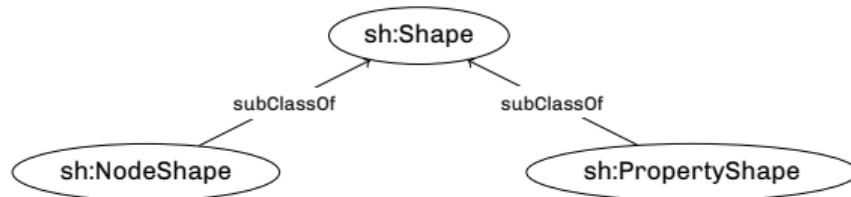
- > Validation, before or after reasoning (or both).
- > Automate certain parts of a data pipeline.
- > Acceptance testing ontologies and/or shapes.
- > Information modelling.

We'll get back to this!

SHACL Shape

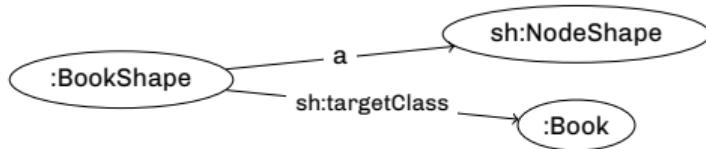
A collection of constraints for given RDF resource.

- › Shapes about focus nodes (**sh:NodeShape**).
- › Shapes about values of a property or path for the focus node (**sh:PropertyShape**).



sh:NodeShape

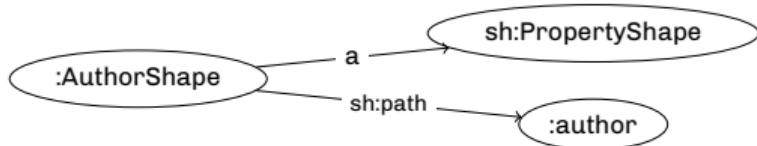
A *node shape* is a shape that is **not** the subject of a triple with *sh:path* as its predicate.



```
:BookShape
a sh:NodeShape ;
sh:targetClass :Book .
```

sh:PropertyShape

A *property shape* is a shape that is the subject of a triple that has *sh:path* as its predicate.



```
:AuthorShape
  a sh:PropertyShape ;
    sh:path :author .
```

BookShape

```
:BookShape
  a sh:NodeShape ;
    sh:targetClass :Book ;
    sh:property :AuthorShape .
```

```
:AuthorShape
  a sh:PropertyShape ;
    sh:path :author .
```

SHACL Core Constraint Components

SHACL Core Constraint Components

Value type

- sh:class Each value node is an instance of a given type.
- sh:datatype Datatype of each value node.
- sh:nodeKind Node kind (IRI, blank node etc.) of each value node.

```
:BookShape
  a sh:NodeShape ;
  sh:targetClass :Book ;
  sh:property [
    sh:path :author ;
    sh:class :Person ;
  ] ;
  sh:property [
    sh:path :published ;
    sh:datatype xsd:date ;
  ] .
```

SHACL Core Constraint Components

Cardinality

sh:minCount	Minimum cardinality as xsd:integer.
sh:maxCount	Maximum cardinality as xsd:integer.

Value range

sh:minExclusive	$x < \$value$
sh:minInclusive	$x \leq \$value$
sh:maxExclusive	$x > \$value$
sh:maxInclusive	$x \geq \$value$

```
:BookShape
  a sh:NodeShape ;
  sh:targetClass :Book ;
  sh:property [
    sh:path :pages ;
    sh:minInclusive 10 ;
    sh:maxExclusive 5000 ;
  ] .
```

SHACL Core Constraint Components

String-based

sh:minLength	Minimum length as xsd:integer.
sh:maxLength	Maximum length as xsd:integer.
sh:pattern	Regular expression.
sh:languageIn	A list of languages as per RFC5646.
sh:uniqueLang	One unique tag per language.

```
:BookShape
  a sh:NodeShape ;
  sh:targetClass :Book ;
  sh:property [
    sh:path :ISBN ;
    sh:pattern "^(?=^(?:\D*\d){10}(?:^(?:\D*\d){3})?$.)([\d-]+$" ;
  ] .
```

SHACL Core Constraint Components

Property pair	Compare two IRIs where,
sh>equals	$x \equiv y$
sh=disjoint	$x \cap y = \emptyset$
sh=lessThan	$x < y$
sh=lessThanOrEquals	$x \leq y$

```
:PersonShape
  a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:property [
    sh:path :birth ;
    sh:lessThanOrEquals :death ;
  ] .
```

SHACL Core Constraint Components

Logical	List of value nodes that,
sh:not	Cannot conform to given shape.
sh:and	Conforms to all provided shapes.
sh:or	Conforms to at least one of the provided shapes.
sh:xone	Conforms to exactly one of the provided shapes.

```
:PersonShape
  a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:or (
    [ sh:path :firstName ; sh:minCount 1 ; ]
    [ sh:path :lastName ; sh:minCount 1 ; ]
  ) .
```

SHACL & OWL

```
:BookShape
  a sh:NodeShape, owl:Class ;
  sh:targetClass :Book ;
  sh:property [
    sh:path :author ;
    sh:or (
      [ sh:class :Author ]
      [ sh:datatype xsd:string ]
    )
  ] .
```

SHACL Core Constraint Components

Shape-based

sh:node

sh:property

Each value node,
Conforms to the given node shape.
Has a given property shape.

Other

sh:closed

sh:ignoredProperties

sh:hasValue

sh:in

Boolean signalling a complete shape.

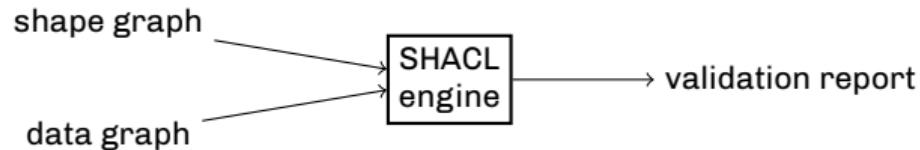
List of properties to ignore.

At least one value node is equal to the given term.

Value node is member of given list.

```
:BookShape
  a sh:NodeShape ;
  sh:targetClass :Book ;
  sh:closed true ;
  sh:ignoredProperties (rdf:type) .
```

SHACL engine



Validation report

Each instance of `sh:ValidationReport` has exactly one value of `sh:conforms`.

`sh:conforms` is true iff the validation did not produce any **validation results**, and false otherwise.

Iff validation conforms false, the report will contain an instance of `sh:ValidationResult`.

```
[  
  a sh:ValidationReport ;  
  sh:conforms true ;  
] .
```

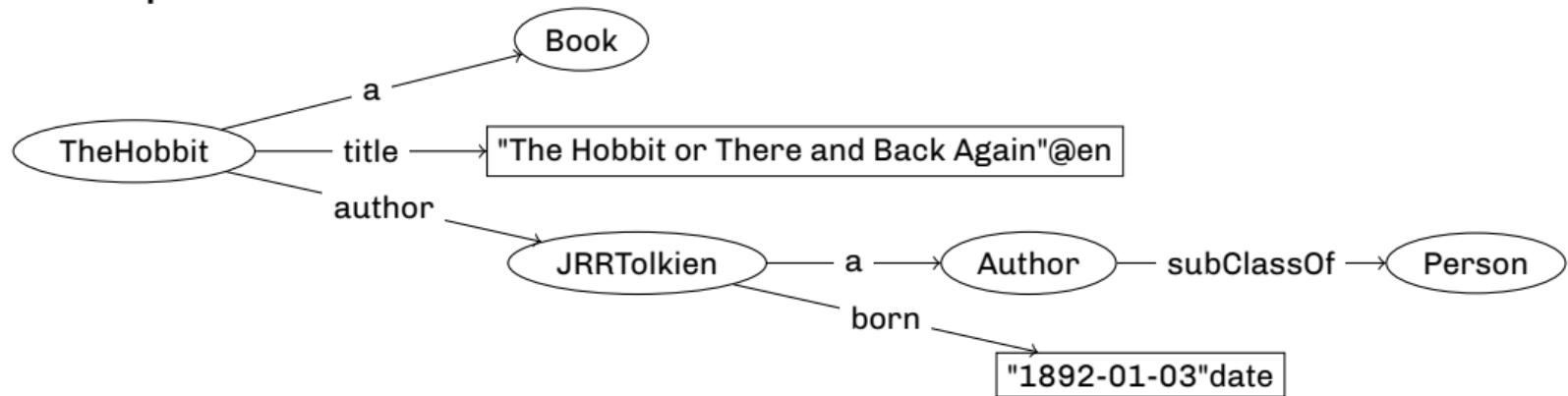
Validation result

All properties described can be specified in a validation result.

sh:focusNode	Node that caused the result.
sh:resultPath	Pointing to value of sh:path
sh:value	Value node that violated constraint.
sh:sourceShape	Shape that given focus node validated against.
sh:sourceConstraintComponent	Constraint component that caused the result.
sh:detail	Parent result containing more details about the violation.
sh:message	Annotation property with textual details.
sh:severity	Default sh:Violation .

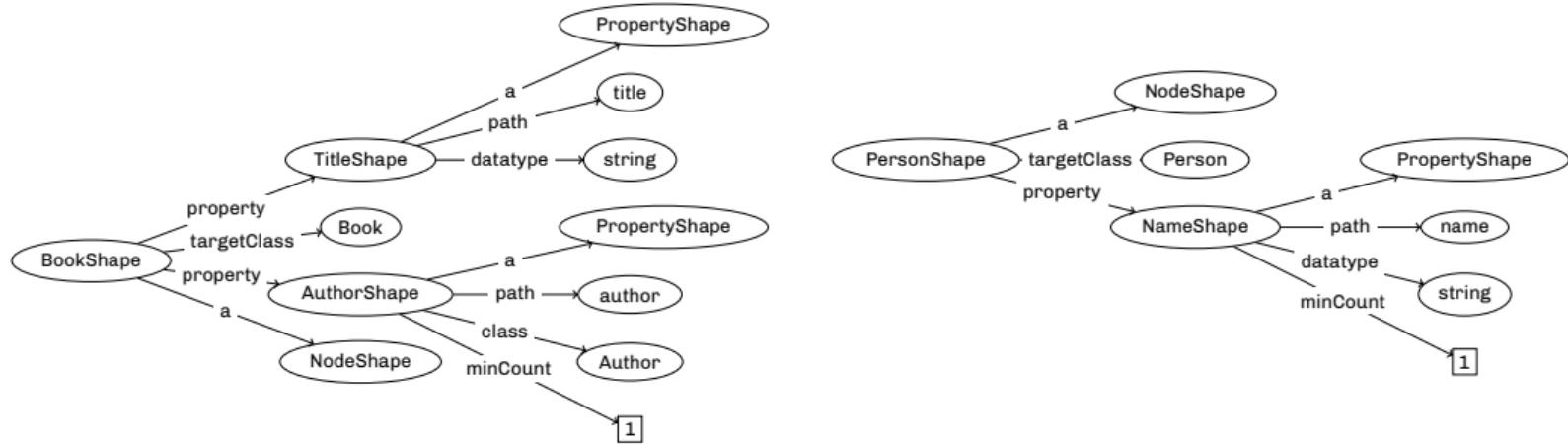
Validation example

Data Graph



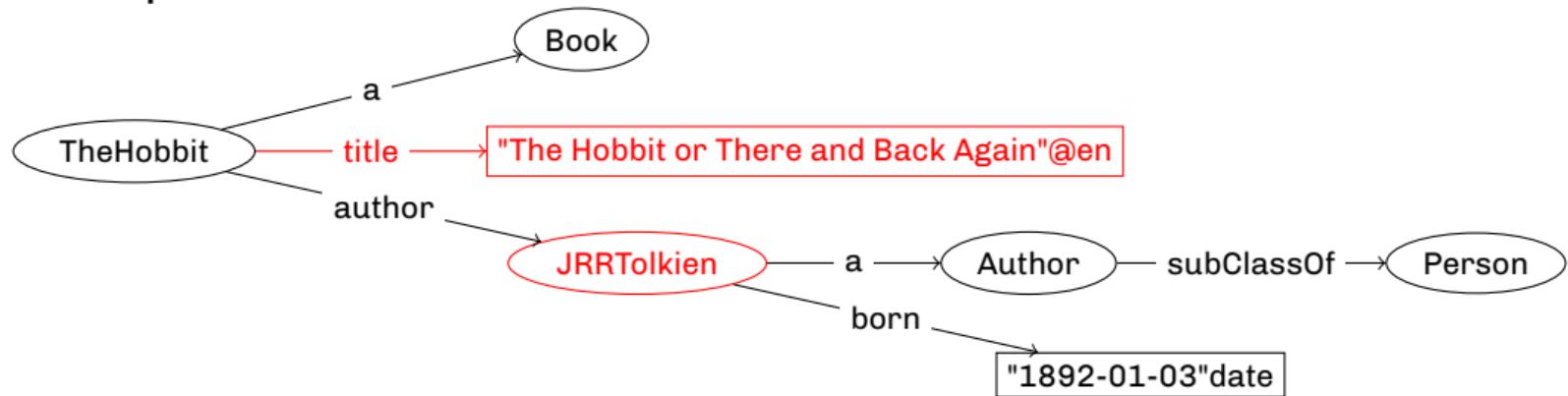
Validation example

Shapes Graph



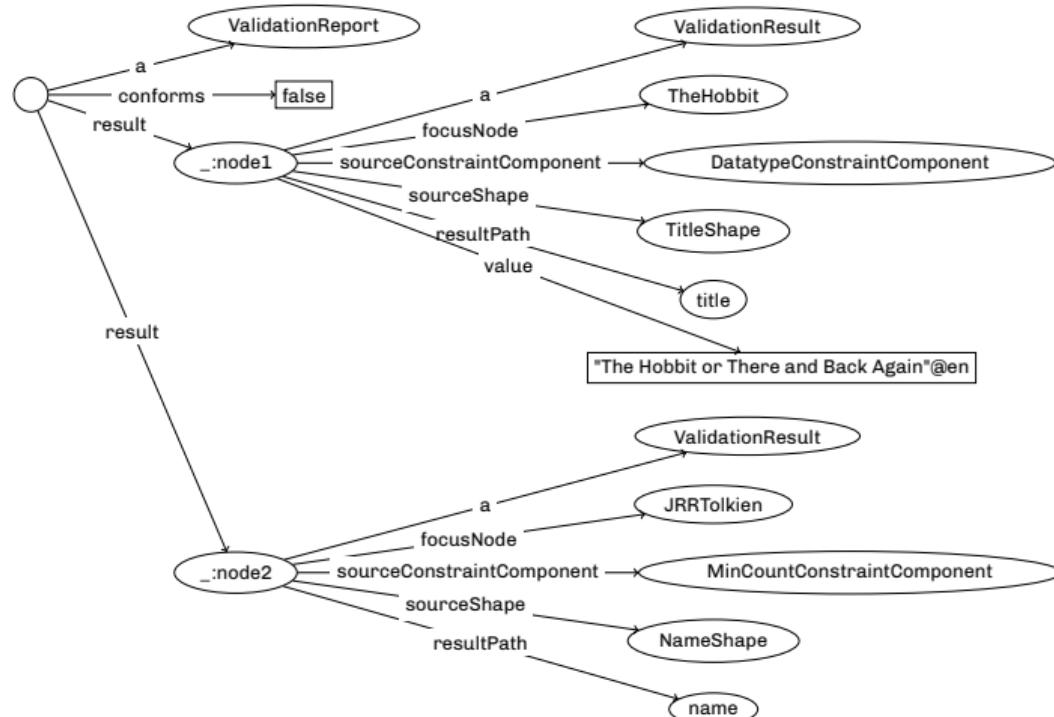
Validation example

Data Graph



Validation example

Validation Result (RDF4J)



Other nice to knows about SHACL

- > Deactivating shapes
- > Non-validating property shape characteristics
 - » sh:name & sh:description
 - » sh:order & sh:group
 - » sh:defaultValue
- > Syntax checking of shapes graph

SHACL Implementations

Framework

ruby-rdf/shacl	https://github.com/ruby-rdf/shacl
dotNetRDF	https://dotnetrdf.org/docs/stable/api/VDS.RDF.Shacl.html
pySHACL	https://github.com/RDFLib/pySHACL
RDF4J	https://rdf4j.org/
Jena	https://jena.apache.org/

Vendors

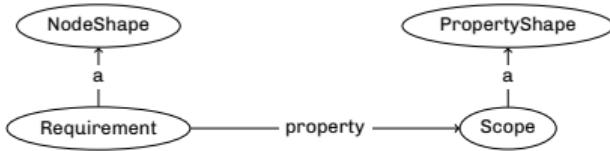
TopQuadrant	https://www.topquadrant.com/
Stardog	https://www.stardog.com/
Cambridge Semantics	https://cambridgeseantics.com/anzograph/
Franz	https://allegrograph.com

Web playground

SHACL Playground	https://shacl.org/playground/
------------------	---

SHACL Stories

Regulatory Requirements



```
:REG201311221404S7P1 a :Requirement, sh:NodeShape ;
  :regulationTitle "A very fine title"@en ;
  :theme "Some very fine theme"@en ;
  rdfs:label "Paragraph title"@en ;
  sh:property :BuiltDate_after19980102 .

:BuiltDate_after19980102 a :Scope, sh:PropertyShape ;
  sh:path :builtDate ;
  sh:minInclusive "1998-01-02" ;
  sh:datatype xsd:date .
```

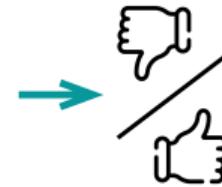
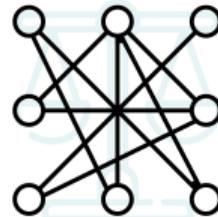
- › Intuitive way of modelling Requirements and Scopes.
- › Verbose vocabulary for describing scope characteristics.
- › Closed World Assumption in the domain of law.

More details on a specific project: http://www.lotico.com/index.php/Data_Shapes_in_Action

Issue Certificates

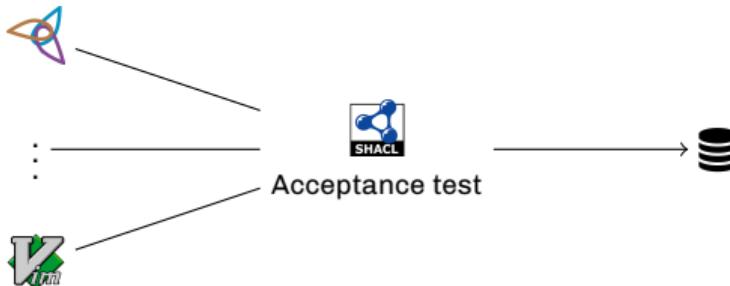


&



```
...  
sh:or (  
  [ sh:and ( # first alternative  
    [ sh:or ( cert:D2A0 cert:D2B0 cert:D3A0  
              cert:D3B0 cert:D4B0 cert:D4F0 ) ]  
    [ sh:path nma:hasSeagoingServiceRequirement ;  
      sh:hasValue nma:SGS_500_1080_D0 ; ]  
  )]  
  
  [ sh:and ( # second alternative  
    [ sh:or ( cert:D2A0 cert:D2B0  
              cert:D3A0 cert:D3B0 ) ]  
    [ sh:path nma:hasSeagoingServiceRequirement ;  
      sh:hasValue nma:SGS_500_720_D0 ; ]  
    [ sh:path nma:hasSeagoingServiceRequirement ;  
      sh:hasValue nma:SGS_500_360_C0 ; ]  
  )]  
)  
;  
...
```

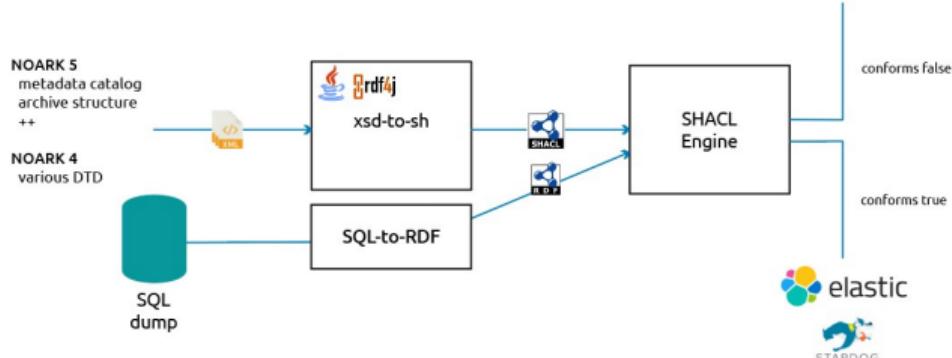
Acceptance Testing



```
:RDFSClassShape a sh:NodeShape ;  
  sh:targetClass rdfs:Class ;  
  sh:property :RDFSLabelShape .  
  
:RDFSLabelShape a sh:PropertyShape ;  
  sh:path rdfs:label ;  
  sh:minCount 1 ; sh:maxCount 1 ;  
  sh:datatype rdf:langString .
```

- › Shapes to validate the **structure** of the TBox.
- › Included in the commit-pipeline, or outside if git is not used.
- › Does not validate the **content** of the graph.

Schema



Journal post snippet

```
<jp/123> a :JournalPost ;
  :numAttachments 1 ;
  :documentMedium :ElectronicArchive ;
  :journalPostNumber "12/123-4"^^xsd:string ;
  ... .
```

SHACL snippet

```
:JP_numAttachments a sh:PropertyShape ;
  sh:datatype xsd:integer ;
  sh:maxCount 1 ;
  sh:path :numAttachments .
```

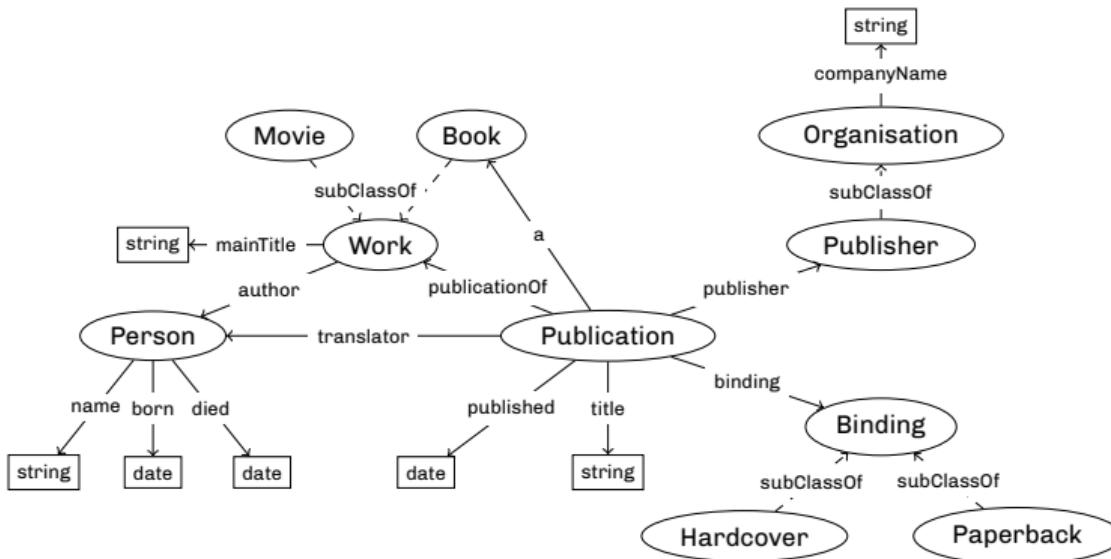


Get to work!



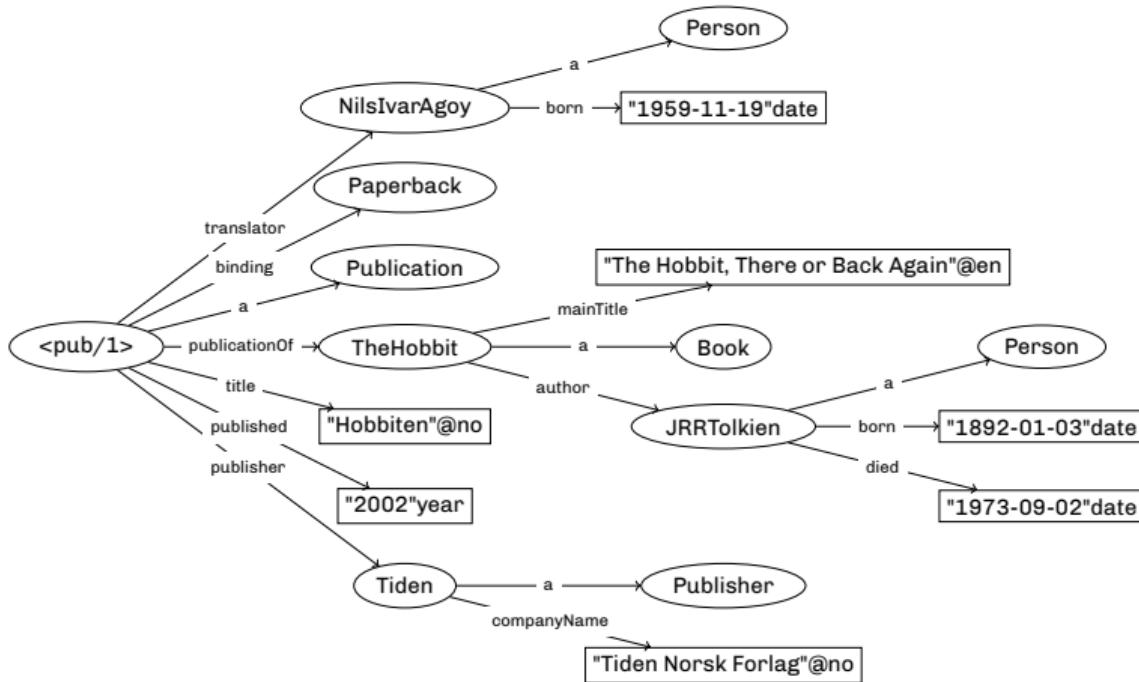
- › Publications of books
- › Persons and affiliations

Example Publication Ontology



<https://tinyurl.com/2mfn4epk>

Example data graph



References & resources

Images

My toy box & bookshelf freepik.com

Around the web

W3C Recommendation	<i>Shape Constraint Language</i>	https://www.w3.org/TR/shacl/
Holger Knublauch	<i>SHACL and OWL Compared</i>	https://spinrdf.org/shacl-and-owl.html
W3C Working Group Note	<i>SHACL Advanced Features</i>	https://w3c.github.io/shacl/shacl-af/
TopQuadrant	<i>DASH Data Shapes</i>	http://datashapes.org/

Book

Jose Emilio labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas, *Validating RDF Data*, 2018.