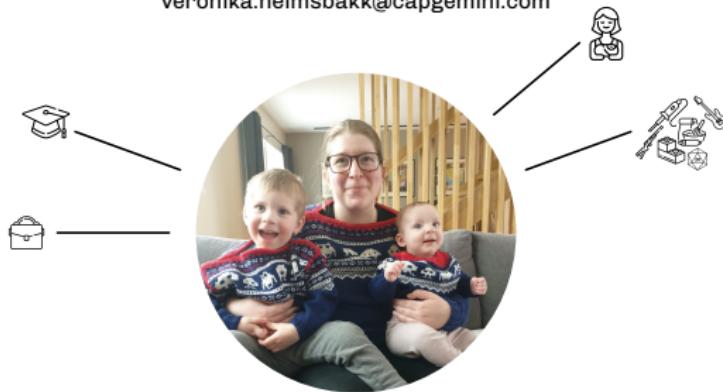


Validating Semantic Knowledge Graphs using SHACL

Veronika Heimsbakk



Senior consultant | Data science & AI | Insights & Data
veronika.heimsbakk@capgemini.com



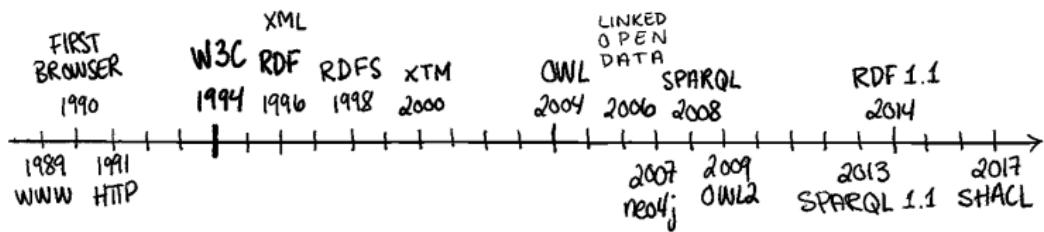
[in vheimsbakk](#) [veleda](#) [veronikaheim](#) [veronahe.no](#)

Agenda

- > Lecture
- > Hands-on exercises
- > Summary

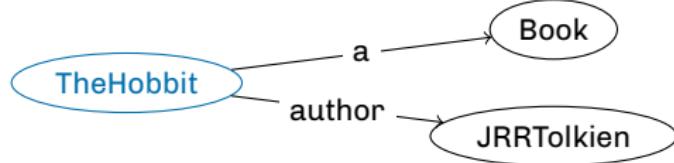
Once upon a time...

TIMELINE OF GRAPH ON THE WEB



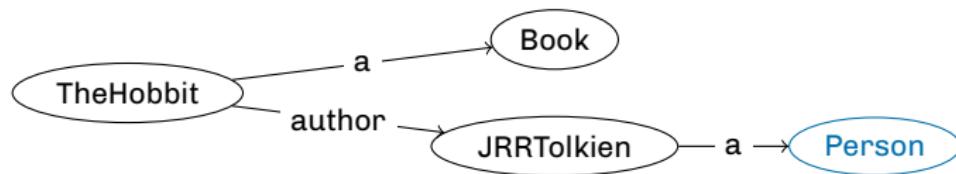
domain & range

domain



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

range

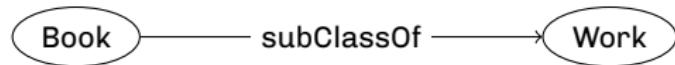


```
:author a rdf:Property ;  
    rdfs:domain :Book ;  
    rdfs:range :Person .
```

TBox & ABox

TBox (terminological component)

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



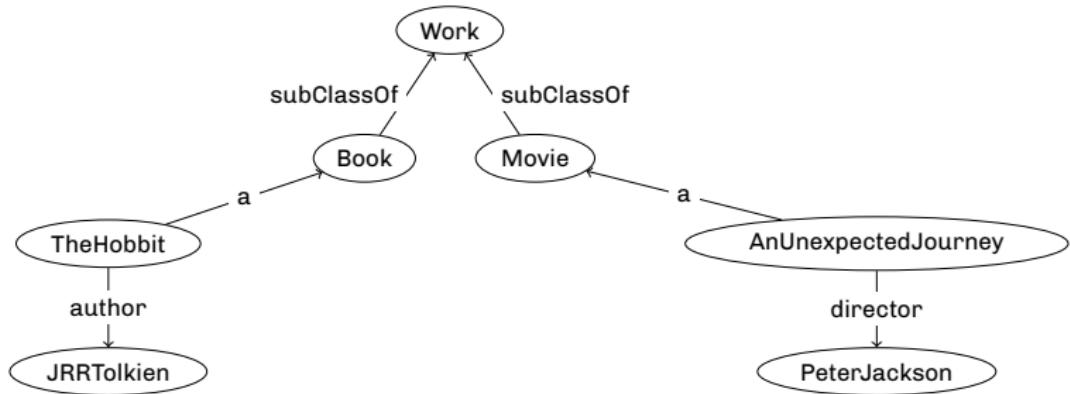
ABox (assertion component)

- > Individuals and instances.



Knowledge graph

TBox + ABox = Knowledge graph ❤



World assumptions

- > 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

- › 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

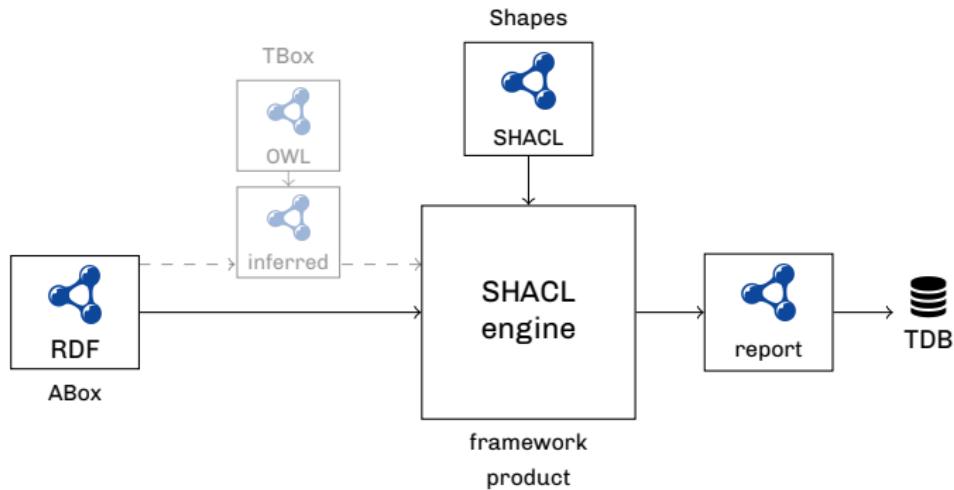
Shape Constraint Language

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.

Workflow



Comparing SHACL and OWL

Common

- › RDF & URIs
- › Infer new triples
- › Rely on RDF Schema (RDFS)

Difference

OWL

- Designed for inference
- Open world assumption
- Limited features
- Logical contradictions

SHACL

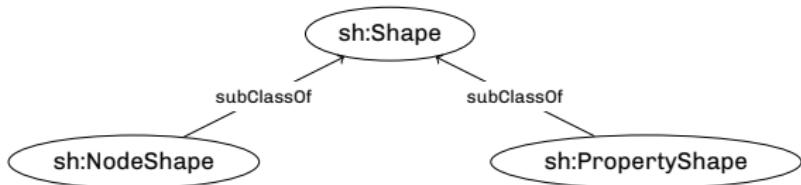
- Designed for validation
- Closed world assumption
- Add your own constraints
- SHACL resources are distinct by default
- Conforms to given schema

When to use SHACL?

| | |
|-------------------------------|-------------------|
| Concept modelling (TBox) | OWL or SHACL |
| Instance data (ABox) | SHACL constraints |
| Need for inference? | OWL (or SHACL) |
| Knowledge graph (TBox + ABox) | OWL, then SHACL |

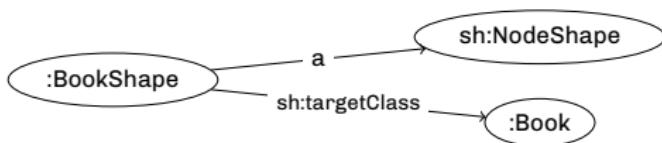
A collection of constraints for given RDF resource.

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



sh:NodeShape

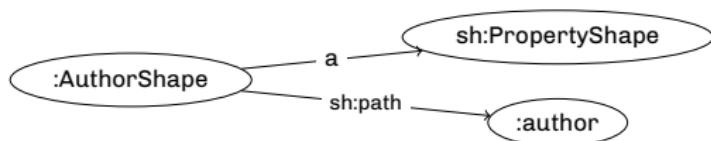
A *node shape* is a shape that is **not** the subject of a triple with *sh:path* as its predicate.



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

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

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 ;
  ] .
```

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:maxInclusive 10 ;
  ] .
```

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

sh>equals

Compare two IRIs where,

$$x \equiv y$$

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 ; ]
  ) .
```

```
:BookShape
  a sh:NodeShape, owl:Class ;
  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 signalising a complete shape.

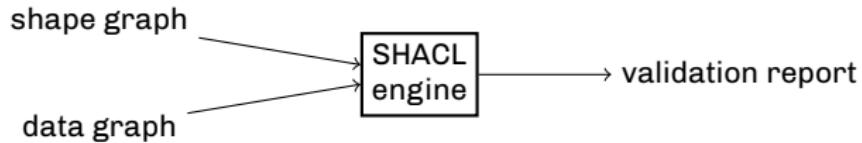
List of properties to ignore.

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 . |

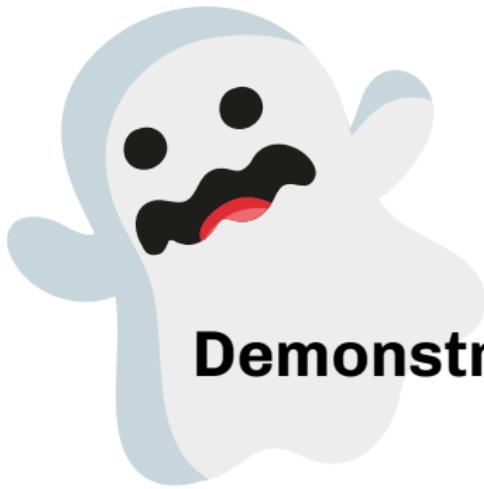
- > Deactivating shapes
- > Non-validating property shape characteristics
 - » sh:name & sh:description
 - » sh:order & sh:group
 - » sh:defaultValue
- > Syntax checking of shapes graph

Not covered in this masterclass

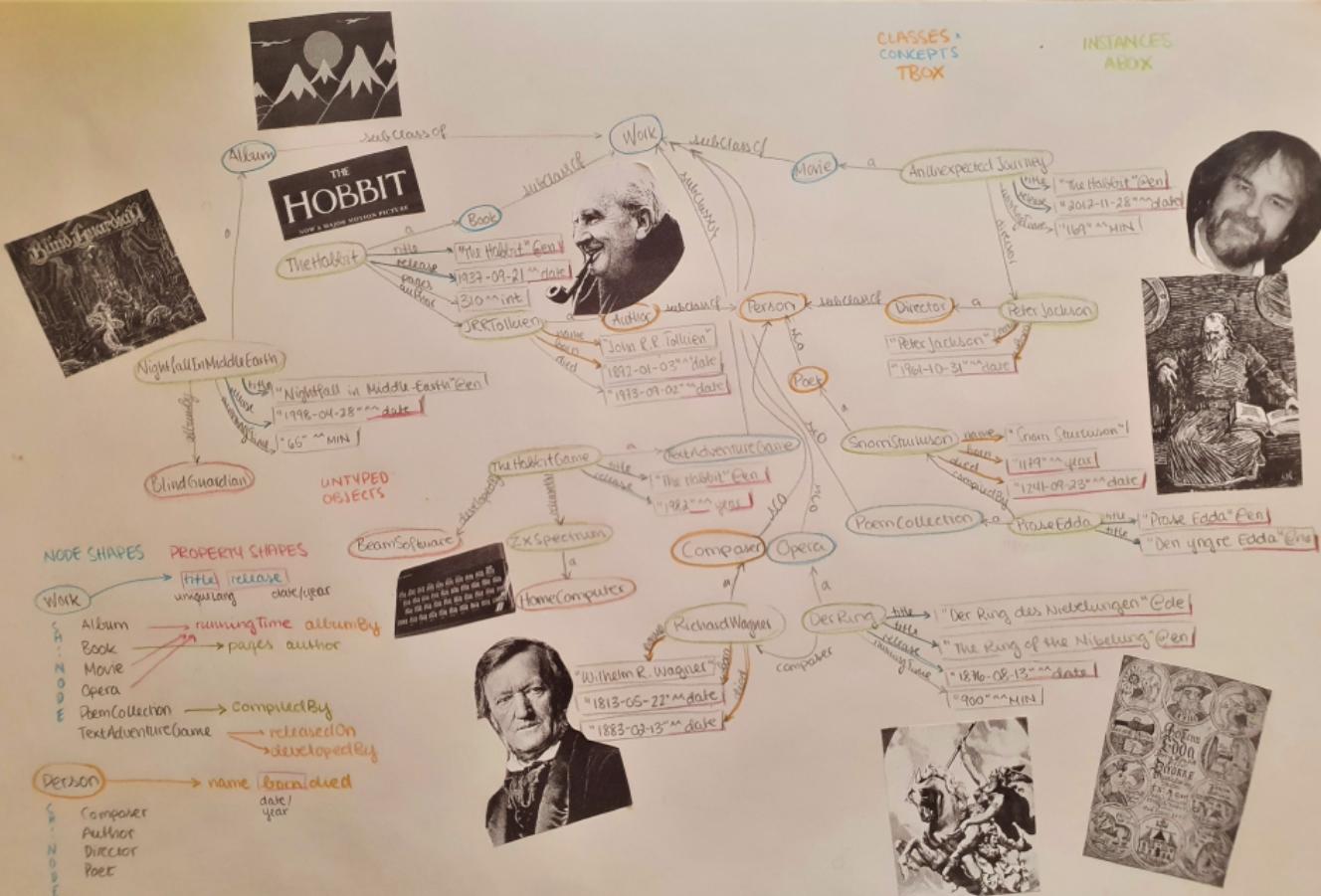
Variants of property paths.

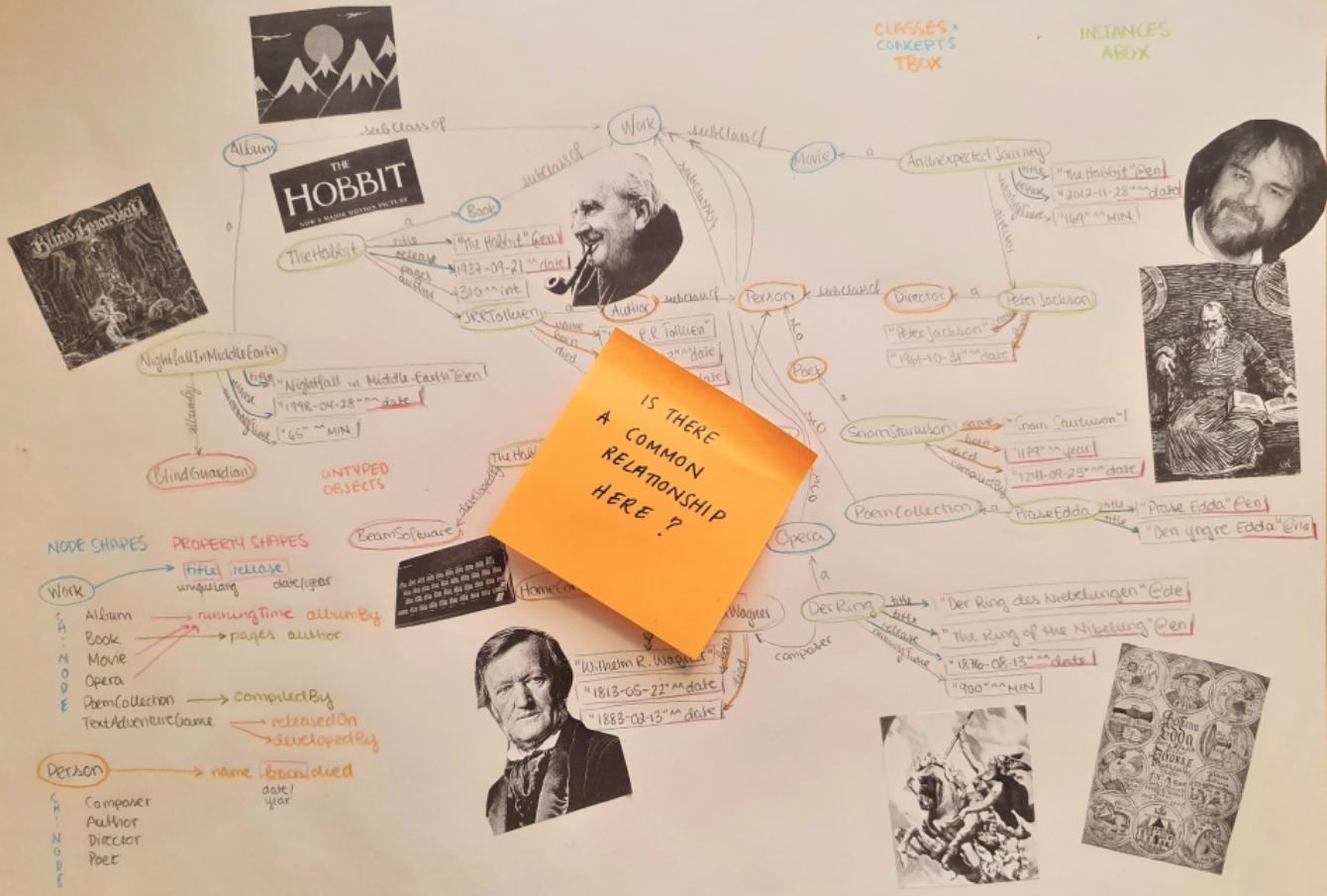
- › sh:inversePath
- › sequence path
- › sh:alternativePath
- › sh:zeroOrMorePath

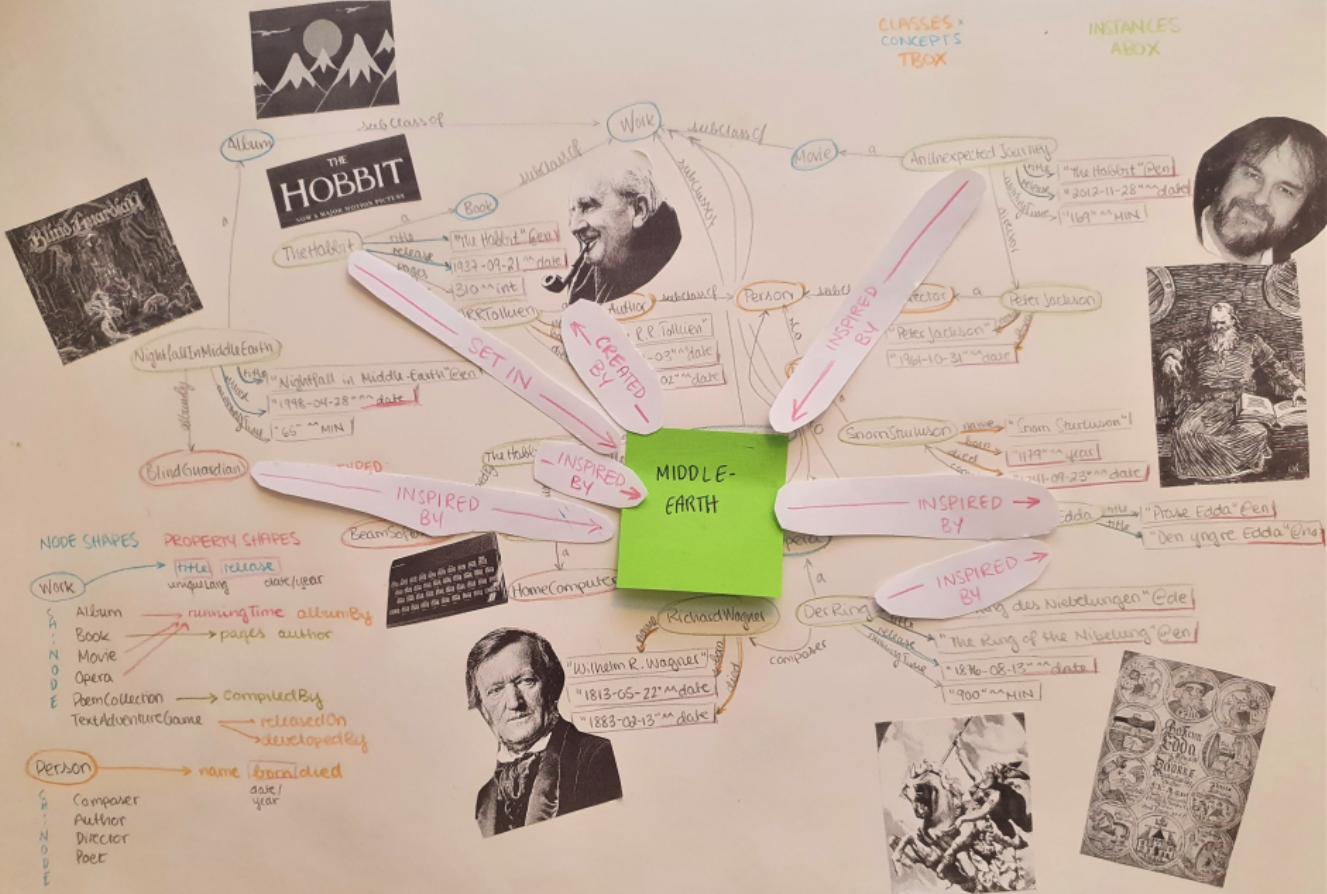
- › SHACL-SPARQL
- › DASH Data Shapes Vocabulary
- › SHACL Advanced Features (SHACL-AF)



Demonstration & hands-on







References & resources

Images

- | | |
|----------|----------------|
| p. 2 | Private photo. |
| p. 11&13 | My toy box. |
| p. 32 | 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/ |