SHACL for the Practitioner

Veronika Heimsbakk

SHACL for the Practitioner

Veronika Heimsbakk

Knowledge Graph Specialist | Data Treehouse

sh@veronahe.no

veleda
vheimsbakk
veronahe.no

It's finally here!

- **2022**: SHACL Masterclass, KGC
- **2022–2024**: Pub scribbling moments
- **Spring 2024**: SHACL Stories
- **Nov 2024**: Writing week #1
- **Dec 2025**: Proofreading #1
- **Mar 2025**: Writing week #2
- **Apr–Jun 2025**: Proofreading #2
- **Jul–Sep 2025**: Editing
- **Sep–Oct 2025**: Finals, layout & print
- **Nov 2025: Release!**

SHACL Masterclasses: Knowledge Connexions (2020), Knowledge Graph Conference (2021, 2022, 2023), Connected Data London (2024), Rail Data Forum (2025)
SHACL Guest Lectures: University of Oslo (2017, 2024, 2025)
`https://github.com/veleda/shacl-masterclass`

Dedicated to

Roger Antonsen (1976–2024), friend, mentor and former colleague

—for the love of logic, and art of communication

Thank you!

Proofreaders

Ghislain Atemezing (European Railway Agency)
Filip Hansen (Norwegian Customs)
Thomas Francart (Sparna)
Jose Emilio Labra Gayo (University of Oviedo)
Holger Knublauch (TopQuadrant)
Trine Kruse (National Archives of Norway)
Birgitte Løwe Johnsen (Alv)
Terje Sylvarnes (Norwegian Digitalisation Agency)

Thank you!

Guest authors

Marina Aguado (European Railway Agency)
Ghislain Atemezing (European Railway Agency)
Ashley Caselli (Knowledge Pixles)
Benjamin Degenhart (FörderFunke)
Maarten Duhoux (European Railway Agency)
Thomas Francart (Sparna)
Holger Knublauch (TopQuadrant)
Dragos Patru (European Railway Agency)
Polymnia Vasilopoulou (European Railway Agency)

Thank you!

Editing

Larry Swanson (metaphacts | Knowledge Graph Insights)

Foreword

Holger Knublauch (TopQuadrant | Co-inventor of SHACL)

Acknowledgements

Bjørn Egil Helgerud, the trophy wife

Håvard Ottestad, friend, former colleague, and SHACL nerd

Robert Engels, mentor and former colleague

NMA project managers; Torbein, Magnus Jonas, Wouter

Christian Mahesh Hansen

Part I—Back to Basics

The Wonder of Logic

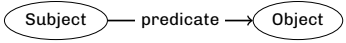
Establishing terms in knowledge representation

- Semantics—the *meaning* of things.
- Knowledge Graph—a graph data structure with accessible semantics (ontology).
- Ontology—formal names and definitions of entities, properties and relations.
- Class, concept, individual; visual thinking through set theory.

The Building Blocks of RDF

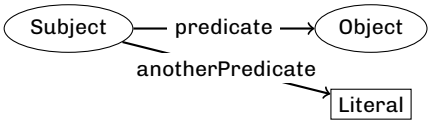
A lightweight introduction to the Resource Description Framework using Turtle

- The triple (a *fact/statement*) with its resources

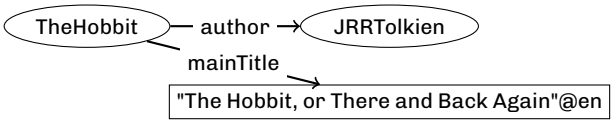


- Global, unique identifiers—Uniform Resource Identifiers (URI)
- RDF Turtle
  - Resource types
  - Blank nodes
  - Lists
  - Property semantics
  - Classes
- World assumptions

Visual notation

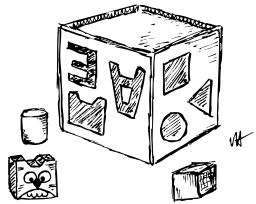


Visual notation



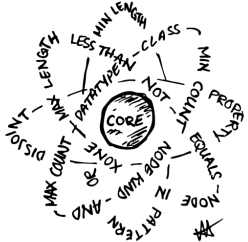
Part II—The Core of SHACL

A Closed World of Shapes



- What the *** is inference and what is validation?
- Difference and similarities between the **Web Ontology Language** and the **Shapes Constraint Language**.
- Examples on how the two work together in a data pipeline.

Shapes of SHACL

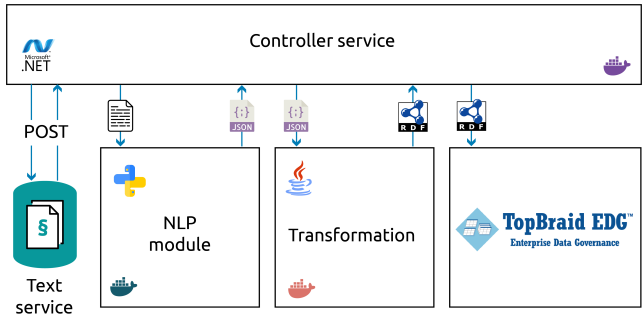


- A systematic setup of every core constraint of SHACL.
- RDF Turtle example for (almost) every single core constraint.
  - Including examples with non-conforming and conforming data for the particular constraint.
- Covering targets (except `SPARQLTarget`) and property paths.
- SHACL-SPARQL constraints, with examples on both node and property level.

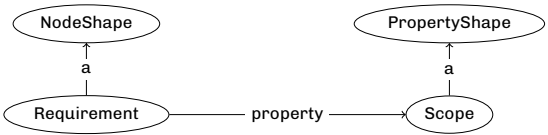
`sh:and`	Conforms to all provided shapes.
`sh:class`	Each value node is an instance of a given type.
`sh:closed`	Boolean signalizing a complete shape.
`sh:datatype`	Datatype of each value node.
`sh:disjoint`	Two properties that don't share a thing, $x \cap y = \emptyset$.
`sh:equals`	Two properties hold exactly the same value, $x \equiv y$.
`sh:hasValue`	At least one value node is equal to the given term.
`sh:ignoredProperties`	List of properties to ignore.
`sh:in`	Value node is member of given list.
`sh:languageIn`	A list of languages
`sh:lessThan`	The value of property *x* must be less than property *y*, $x < y$.
`sh:lessThanOrEquals`	The value of property *x* must be less than or equals to property *y*, $x <= y$.
`sh:maxCount`	Maximum cardinality as `xsd:integer`.
`sh:maxExclusive`	$x >$ value
`sh:maxInclusive`	$x >=$ value
`sh:maxLength`	Maximum length as `xsd:integer`.
`sh:minCount`	Minimum cardinality as `xsd:integer`.
`sh:minExclusive`	$x <$ value
`sh:minInclusive`	$x <=$ value
`sh:minLength`	Minimum length as `xsd:integer`.
`sh:node`	Conforms to a given node shape.
`sh:nodeKind`	Node kind (IRI, blank node etc.) of each value node.
`sh:not`	Cannot conform to given shape.
`sh:or`	Conforms to at least one of the provided shapes.
`sh:pattern`	Regular expression.
`sh:property`	Conforms to a given property shape.
`sh:uniqueLang`	One unique tag per language.
`sh:xone`	Conforms to exactly one of the provided shapes.

Part III—SHACL Stories

The Norwegian Maritime Authority



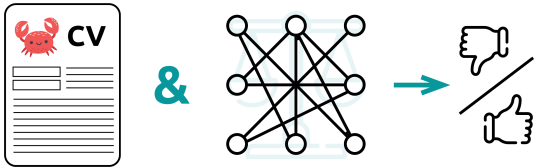
Regulatory Requirements



- The **Requirement** is the core at *every* activity by the Norwegian Maritime Authority.
- SHACL as verbose vocabulary for describing machine-readable requirements.
- CWA for the domain of law.

More on reasons why in *Using the Shapes Constraint Language for modelling regulatory requirements* by Veronika Heimsbakk and Kristian Torkelsen, `https://arxiv.org/abs/2309.02723`

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_DO ; ]
  )]

  [ sh:and ( # second alternative
    [ sh:or ( cert:D2A0 cert:D2B0
              cert:D3A0 cert:D3B0 ) ]
    [ sh:path nma:hasSeagoingServiceRequirement ;
      sh:hasValue nma:SGS_500_720_DO ; ]
    [ sh:path nma:hasSeagoingServiceRequirement ;
      sh:hasValue nma:SGS_500_360_CO ; ]
  )]
) ;
...
```

European Open Data Portal

The European Parliament Open Data Portal (EPODP), launched in 2023, is a knowledge-graph–driven platform that integrates data from around twenty business applications into a semantically interoperable system based on widely used ontologies.

- SHACL is central to the portal, supporting data validation, public documentation generation, and exploratory uses like query and mapping-rule generation.
- The portal maintains multiple SHACL application profiles and dataset-specific definitions rather than a single model, specifying both the overall graph structure and the constraints of individual datasets.
- SHACL specifications are authored in spreadsheets and converted automatically, enabling a systematic, scalable approach to documentation, validation, and data extraction.

`https://data.europarl.europa.eu`

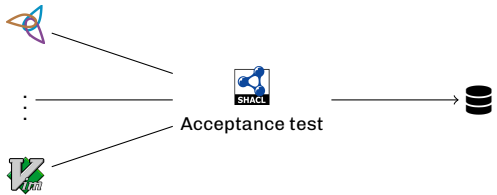
The European Union Agency for Railways (ERA)

ERA uses SHACL to formally encode the business rules embedded in EU railway legislation, enabling consistent interpretation, validation, and interoperability across complex, multi-actor railway data systems.

- ERA applies SHACL to encode legal and business rules in a machine-interpretable way, reducing ambiguity and improving interoperability.
- A data-centric strategy drives the use of ontologies, shared vocabularies, and a FAIR repository of SHACL rules across railway systems.
- Competency questions and ontology axioms guide the creation of SHACL constraints used to validate infrastructure and vehicle data.
- SHACL validation ensures legal compliance and data quality across critical registers, supporting scenarios like route compatibility checks and cross-system interoperability.

https://www.era.europa.eu/domains/registers/era-knowlege-graph_en

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 ontologies.
- Included in the commit-pipeline, or outside if git is not used.
- Does not validate the *content* of the graph.

Nakala

The Nakala, a repository of French research data in the Social Sciences and Humanities, project demonstrates how the Sparnatural knowledge graph explorer uses SHACL shapes to automatically generate a user-friendly query interface for RDF datasets, even when no pre-existing SHACL shapes exist.

- Sparnatural leverages SHACL to drive query UIs, automatically interpreting RDF graphs to determine classes, properties, datatypes, and cardinalities.
- Nakala's SHACL shapes were generated automatically, supporting FAIR principles and avoiding manual shape creation.
- Dataset statistics and filtering were applied to simplify the UI, choosing appropriate widgets for literals and IRIs based on usage frequency.
- The combination of SHACL profiles, reports, filters, and Sparnatural enables rapid deployment of functional query interfaces with minimal manual effort.

https://www.nakala.fr/sparnatural/

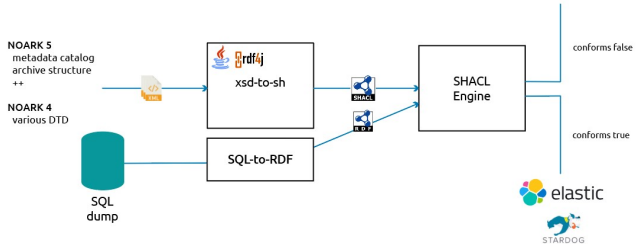
Urban Subsurface Data

The *ImpulseSubsurface* project uses SHACL to improve the quality, completeness, and regulatory compliance of urban subsurface datasets, integrating data from multiple sources with varying schemas and quality.

- SHACL is used to validate and correct heterogeneous urban subsurface data, ensuring accuracy and completeness.
- Data completion and enrichment leverage SHACL inference rules to fill missing information and enhance 3D representation.
- Compliance checking is formalized through SHACL shapes and SHACL-JS rules to verify adherence to regulations.
- The SHACL-X framework allows domain-specific computation to be integrated without altering core SHACL shapes, increasing flexibility and maintainability.

https://github.com/SHACL-X

SHACL as a Schema for Existing Standards



Journal post snippet

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

SHACL snippet

```
[
  sh:datatype xsd:integer ;
  sh:maxCount 1 ;
  sh:predicate :numAttachements ;
]
```

Fun fact! The SHACL Engine implemented at eInnsyn led to the SHACL Engine for rdf4j.

FörderFunke

FörderFunke uses SHACL and SPARQL to proactively match users to state benefits, closing the information gap for citizens in Germany by encoding eligibility criteria as requirement profiles.

- Eligibility criteria are encoded as SHACL shapes (requirement profiles) and used to dynamically guide user questions and eligibility determination.
- User profiles are enriched and validated via SHACL for plausibility, logical consistency, and domain-specific rules.
- SHACL shapes are leveraged for UI generation and orchestrating the quiz flow based on conditional logic.
- The system is fully client-side, open-source, privacy-preserving, and designed for integration with state infrastructure and international standards.

🌐 `https://foerderfunke.org/`

TopBraid EDG

TopBraid EDG provides enterprise-grade tooling for SHACL, enabling users to create, edit, and manage SHACL ontologies and instance data through a flexible graphical interface.

- Interactive panels in TopBraid EDG allow SHACL ontology editing, property management, and form-driven instance editing with real-time validation.
- SHACL shapes drive data entry forms and prevent invalid data while supporting meta-shapes for structural validation.
- Advanced features include inferencing with `sh:values` rules, GraphQL schema generation, and dynamic property computation using SPARQL or JavaScript.
- Enterprise users can integrate scripts, automate workflows, and leverage DASH-based extensions for custom UI widgets and collaboration.

🌐 `https://www.topquadrant.com/topbraid-edg/`

Appendix

SHACL & OWL Example

A G-type star is a star with a kelvin temperature between 5300 and 6000.

- Natural language
- DL ALC
- Manchester
- OWL in RDF Turtle
- SHACL in RDF Turtle

SHACL Implementations

Frameworks

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

Vendors

Most vendors support SHACL, but TopQuadrant, Hanami and Metaphacts are SHACL-first

Web playgrounds

SHACL Play!	`https://shacl-play.sparna.fr/play/`
SHACL Playground	`https://shacl.org/playground/`
Zazuko SHACL Playground	`https://shacl-playground.zazuko.com/`

...and more: `https://github.com/w3c-cg/awesome-semantic-shapes`

How can I get a copy?

shacl.veronahe.no