Validating Bioschemas markup

SWAT4LS'2022 tutorials

Alban Gaignard
CNRS, ELIXIR-FR
Institut du Thorax, Nantes, France

Bioschemas profiles

37 profiles

Name	Group	Use Cases	Cross Walk	Task & Issues	Examples	Live Deploys
ChemicalSubstance (v0.4-RELEASE) 07 April 2020	Chemicals					
ComputationalTool (v1.0-RELEASE) 11 October 2021	Tools					
ComputationalWorkflow (v1.0-RELEASE) 09 March 2021	Workflow					
DataCatalog (v0.3-RELEASE-2019_07_01) 01 July 2019	Data Repositories					
Dataset (v0.3-RELEASE-2019_06_14) 14 June 2019	Datasets					
FormalParameter (v1.0-RELEASE) 09 March 2021	Workflow					
Gene (v1.0-RELEASE) 07 April 2021	Genes					
MolecularEntity (v0.5-RELEASE) 07 April 2020	Chemicals					
Protein (v0.11-RELEASE) 07 April 2020	Proteins					
Sample (v0.2-RELEASE-2018_11_10) 10 November 2018	Samples					
Taxon (v0.6-RELEASE) 07 April 2020	Biodiversity					

- different use of schema.org classes and properties
- Communities agree on minimal/ recommended/ optional annotation

Bioschemas profiles

Profiles ≠ Classes (types)

Bioschemas **profiles** specify

- which RDF triples are expected to describe specific entities
- which ontology classes or properties should be used (mostly from Schema.org)
- different marginalities / priorities (minimal, recommended, optional)
- different cardinalities (one or many) for predicates

Example

ComputationalTool Profile

Version: 1.0-RELEASE (11 October 2021)

Bioschemas specification for describing a SoftwareApplication in the Life Sciences

If you spot any errors or omissions with this type, please file an issue in our GitHub.

Description | Contributors | Links

Schema.org hierarchy

This Profile fits into the schema.org hierarchy as follows:

Thing > CreativeWork > SoftwareApplication

```
ex:myTool    rdf:type    schema:SoftwareApplication .
```

Example

[...]

Property	Expected Type	Description	CD	Controlled Vocabulary	Example
Marginality: Minimum.					
@context	URL	Used to provide the context (namespaces) for the JSON-LD file. Not needed in other serialisations.	ONE		
@type	Text	Schema.org/Bioschemas class for the resource declared using JSON-LD syntax. For other serialisations please use the appropriate mechanism. While it is permissible to provide multiple types, it is preferred to use a single type.	MANY	Schema.org, Bioschemas	
@id	IRI	Used to distinguish the resource being described in JSON-LD. For other serialisations use the appropriate approach.	ONE		
dct:conformsTo	IRI	Used to state the Bioschemas profile that the markup relates to. The versioned URL of the profile must be used. Note that we use a CURIE in the table here but the full URL for Dublin Core terms must be used in the markup (http://purl.org/dc/terms/conformsTo), see example.	ONE	Bioschemas profile versioned URL	
description	Text	**Schema:** A description of the item. **Bioschemas:** A short description of the tool.	ONE		

[...]

```
ex:myTool   rdf:type   schema:SoftwareApplication, prov:SoftwareAgent ;
            schema:description "This tool does … " ;
            schema:license <https://spdx.org/licenses/MIT.html> ;
            schema:codeRepository <http://github.com/...> .
```

Manually checking conformance

```
ex:myTool    rdf:type    schema:SoftwareApplication, prov:SoftwareAgent ;
             schema:description "This tool does … " ;
             schema:license <https://spdx.org/licenses/MIT.html> ;
             schema:codeRepository <http://github.com/...> .
```

Major issues

This markup is missing `dct:conformsTo` properties as well as `schema:name` and `schema:url` ...

Minor issues

This markup should also contains `schema:author`, `schema:citation`, etc.

Not realistic from a human point of view → automation needed !

Supporting automated Bioschemas validation with SHACL

SHACL

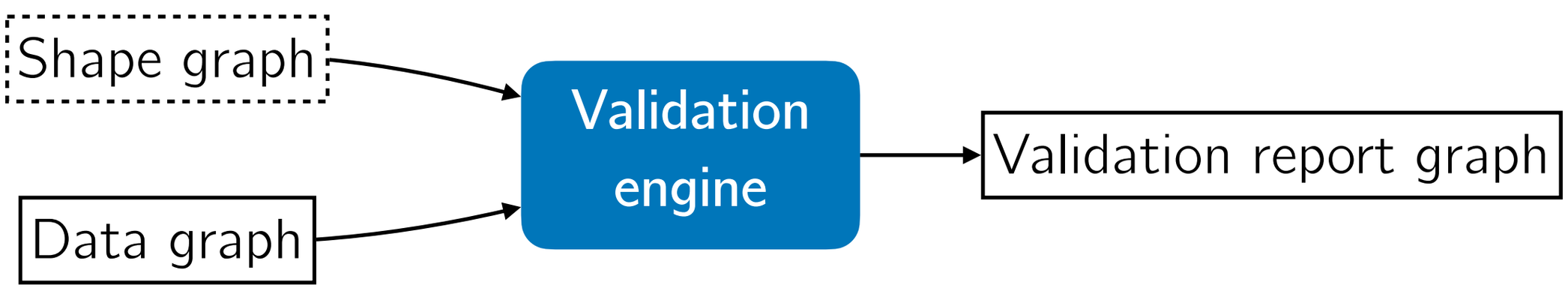
Stands for "SHApes Constraint Language".

W3C recommendation (July 2017) aimed at validating RDF graphs.

Similar to the ShEx (Shape Expressions) initiative.

Shape = pattern / constraints for an RDF graph

SHACL shapes are written with RDF triples



Shape graphs

```
ns:shape_1 rdf:type sh:NodeShape ;
    sh:targetClass  sc:SoftwareApplication ;
    sh:property [
        sh:path sc:description ;
        sh:minCount 1 ;
        sh:severity sh:Violation
    ] .
```

- SHACL provides a controlled vocabulary to describe the topology/ structure of RDF graphs.
- constraints on specific graph nodes
- constraints on specific graph edges

Target nodes / classes

A node shape can be bound to

- a specific class instance (sh:targetNode)
- all instances of a given class (sh:targetClass)
- all nodes subject of a given predicate (sh:targetSubjectsOf)
- all nodes object of a given predicate (sh:targetObjectsOf)

Validation report

```
Validation Report
Conforms: False
Results (2):
Constraint Violation in MinCountConstraintComponent
(http://www.w3.org/ns/shacl#MinCountConstraintComponent):
    Severity: sh:Violation
    Source Shape: [ sh:minCount Literal("1",
datatype=xsd:integer) ; sh:path sc:name ; sh:severity
sh:Violation ]
    Focus Node: ex:myTool
    Result Path: sc:name
    Message: Less than 1 values on ex:myTool->sc:name
Validation Result in MinCountConstraintComponent (http://
www.w3.org/ns/shacl#MinCountConstraintComponent):
    Severity: sh:Warning
    Source Shape: [ sh:minCount Literal("1",
datatype=xsd:integer) ; sh:path sc:citation ; sh:severity
sh:Warning ]
    Focus Node: ex:myTool
    Result Path: sc:citation
    Message: Less than 1 values on ex:myTool->sc:citation
```

Depending on the evaluation engine, you can get a textual report:

- Yes/No answer for the global validation
- One message per error
- Source shape leading to error
- Focus node leading to error

The report is generated from the validation report graph.

Validation report

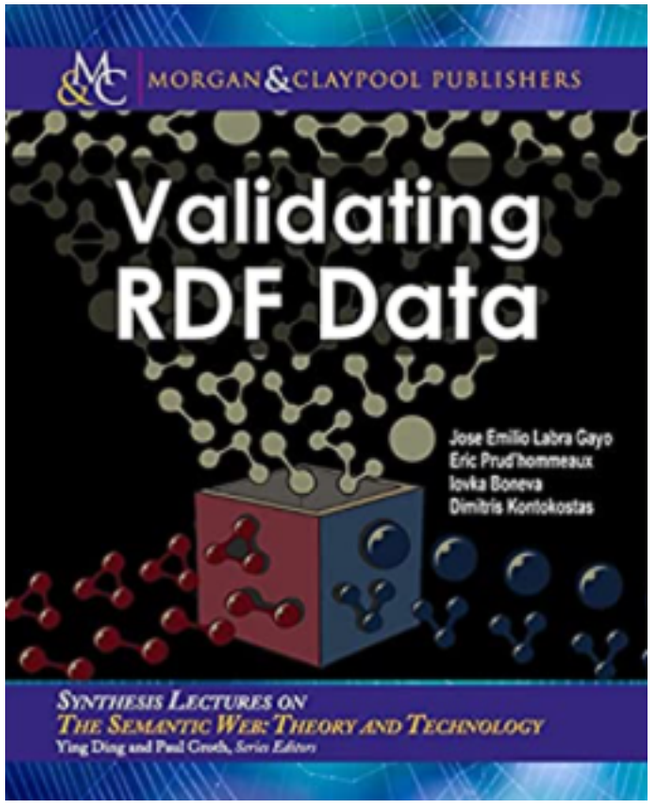
```
@prefix sc: <http://schema.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

[] a sh:ValidationReport ;
    sh:conforms false ;
    sh:result [ a sh:ValidationResult ;
            sh:focusNode <http://
bioschemas.validation.tutorial/myTool> ;
            sh:resultMessage "Less than 1 values on
ex:myTool->sc:citation" ;
            sh:resultPath sc:citation ;
            sh:resultSeverity sh:Warning ;
            sh:sourceConstraintComponent
sh:MinCountConstraintComponent ;
            sh:sourceShape [ sh:minCount 1 ;
                    sh:path sc:citation ;
                    sh:severity sh:Warning ] ],
        [ a sh:ValidationResult ;
            sh:focusNode <http://
bioschemas.validation.tutorial/myTool> ;
            sh:resultMessage "Less than 1 values on
ex:myTool->sc:name" ;
            sh:resultPath sc:name ;
            sh:resultSeverity sh:Violation ;
            sh:sourceConstraintComponent
sh:MinCountConstraintComponent ;
            sh:sourceShape [ sh:minCount 1 ;
                    sh:path sc:name ;
                    sh:severity sh:Violation ] ] .
```

- SHACL provides a controlled vocabulary to describe validation reports in RDF.
- Validations report can be shared and queried on the web following Linked Data principles.

To go further ...

José Emilio Labra Gayo, Eric Prud'hommeaux, Iovka Boneva and Dimitris Kontokostas. “Validating RDF Data.” Validating RDF Data (2017).

Online version: https://book.validatingrdf.com



ISWC 2020 tutorial, Jose Emilio Labra Gayo:
http://www.validatingrdf.com/tutorial/iswc2020/

Tools

FAIR-Checker

- Aim 1: evaluating FAIR metrics with semantic web technologies
- Aim 2: empowering data providers to inspect and improve the quality of metadata
- http://fair-checker.france-bioinformatique.fr
  still under active development

Screenshots

Controlled vocabularies | Bioschemas

Bioschemas is a community effort aimed at reusing and extending Schema.org for better life science digital resource findability. Several profiles are defined for each kind of Life Science resources, specifying minimal, recommended or optional information. Am I missing minimal information ? Should I provide other information for better findability ?

At the moment, profiles supported are: ScholarlyArticle, Dataset, ComputationalTool

Check BioSchemas

Requirements
Property http://schema.org/identifier **must be** provided
Property http://schema.org/headline **must be** provided

Improvements
Property http://schema.org/backstory **should be** provided
Property http://schema.org/alternateName **should be** provided
Property http://schema.org/pageEnd **should be** provided
Property http://schema.org/citation **should be** provided
Property http://schema.org/pageStart **should be** provided
Property http://schema.org/dateModified **should be** provided
Property http://schema.org/about **should be** provided
Property http://schema.org/isBasedOn **should be** provided
Property http://schema.org/license **should be** provided
Property http://schema.org/isPartOf **should be** provided
Property http://schema.org/dateCreated **should be** provided

Annotate missing BioSchemas properties

Bioschemas-SHACL-validation

Developed during the Elixir BioHackathon 2021

Aim: instrumenting Bioschemas profiles with a generic validation tool

```
 main ?  python main.py -u "http://bio.tools/bwa"    ✔  3.9.7 (bioschemas-valid) Py  6345  15:07:20

Trying to validate https://bio.tools/bwa as a(n) http://schema.org/SoftwareApplication resource
Generating SHACL shape for sc:SoftwareApplication
ERROR: Property http://schema.org/name must be provided for https://bio.tools/bwa
ERROR: Property http://schema.org/description must be provided for https://bio.tools/bwa
ERROR: Property http://schema.org/url must be provided for https://bio.tools/bwa
WARNING: Property http://schema.org/additionalType should be provided for https://bio.tools/bwa
WARNING: Property http://schema.org/applicationCategory should be provided for https://bio.tools/bwa
WARNING: Property http://schema.org/author should be provided for https://bio.tools/bwa
WARNING: Property http://schema.org/license should be provided for https://bio.tools/bwa
WARNING: Property http://schema.org/softwareVersion should be provided for https://bio.tools/bwa
```

Supported profiles: CreativeWork, SoftwareApplication, Dataset, ScholarlyArticle, MolecularEntity, Gene, Study, Person, SoftwareSourceCode, Protein, SequenceAnnotation, SequenceRange

https://github.com/BioSchemas/bioschemas-validation

Hands-on

Hands-on session

Agenda

- Loading Bioschemas markup into an RDF Knowledge Graph with Jupyter notebook, Python, RDFLib
- Writing a simple SHACL shape
- Evaluating it through the PySHACL library
- Generate a human-friendly validation result
- Demo of the Bioschemas SHACL validator on some examples (or local execution)
- (If we have time) automate the generation of SHACL shapes with a textual template engine

Questions ?

https://github.com/BioSchemas/bioschemas-validation

https://mybinder.org/v2/gh/BioSchemas/bioschemas-validation/HEAD