Data Validation using SHACL

Madhumitha Senthilvel
RWTH Aachen University | TenneT Offshore GmbH

Hello ☺

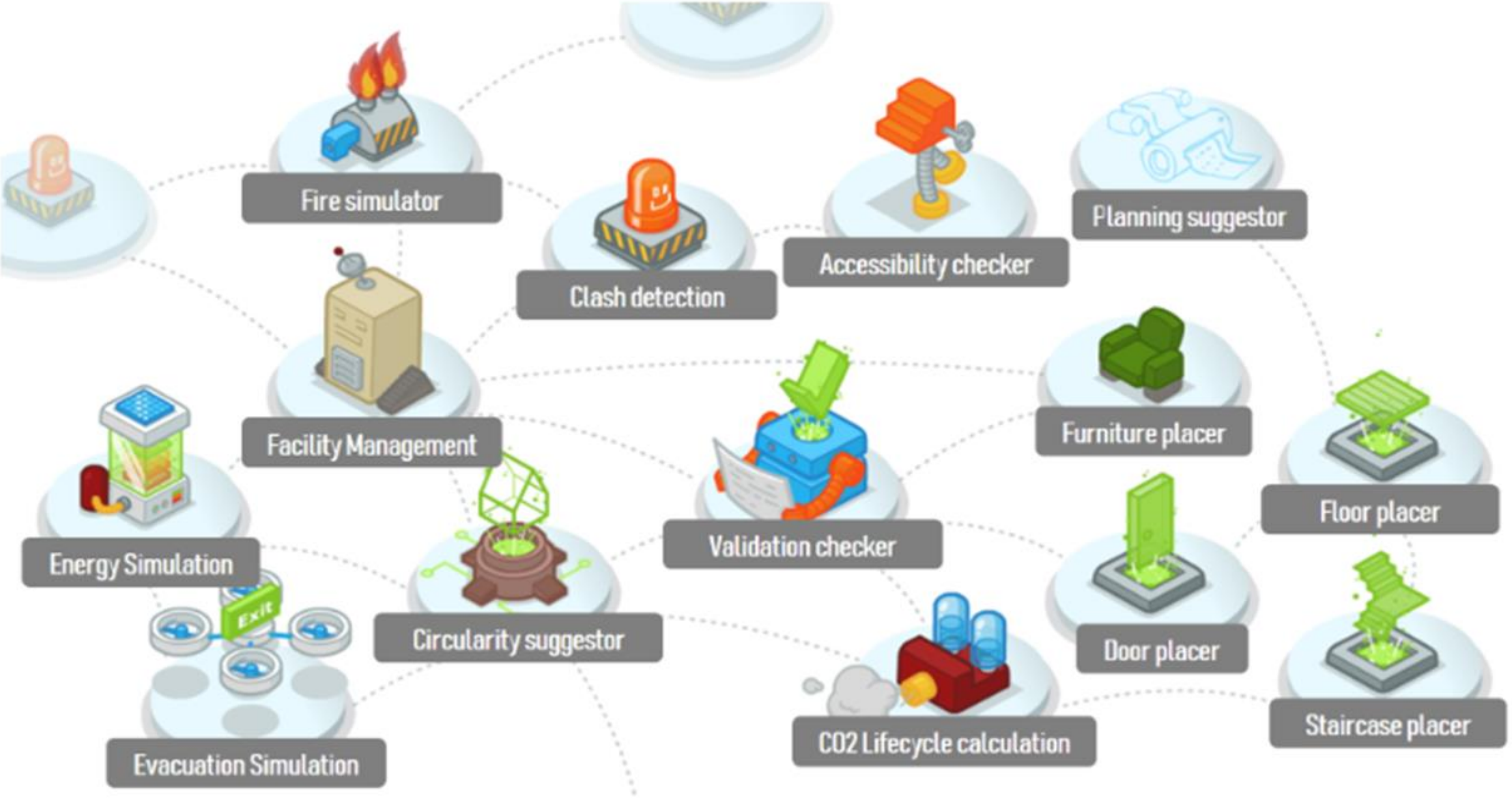
- PhD Candidate | RWTH Aachen University | Germany
- Linking and Managing heterogeneous data in Information Containers in CDEs
- Information and Document Manager | TenneT 2GW German projects
- M.S in Building Technology and Construction Management | IIT Madras
- Bacherlors in Civil Engineering

Why validate data?

Data Quality & Integrity

Local/National/International Codes & Regulations

Why validate data?



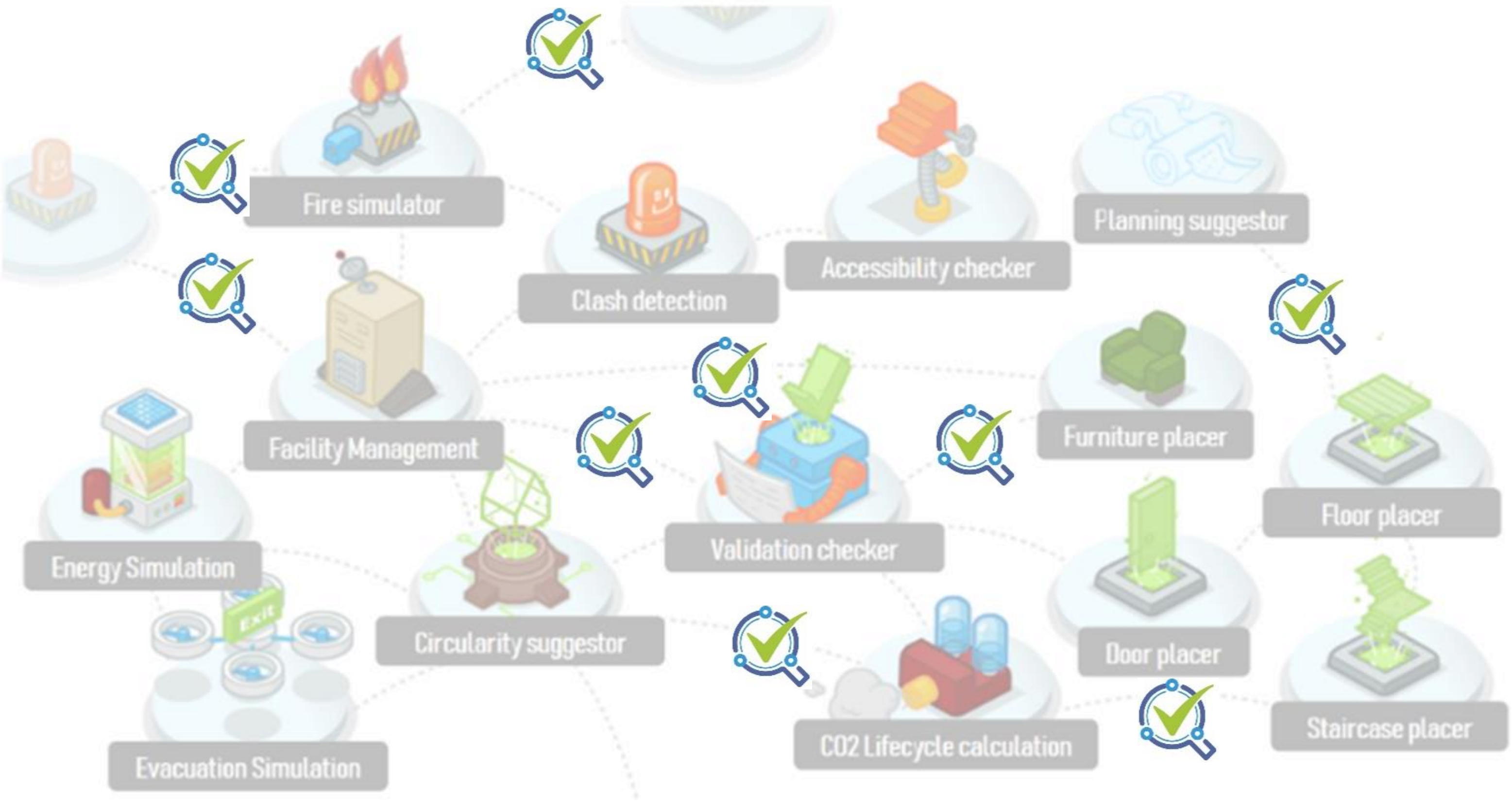
ELASSTIC EU FP7 project (Berlo, Jagt, Walsum, Klein, & Müllers, 2016)

Why validate data?

- Catch errors in data
- Check for missing information
- Check if incoming data contains all information required for analysis by the tool

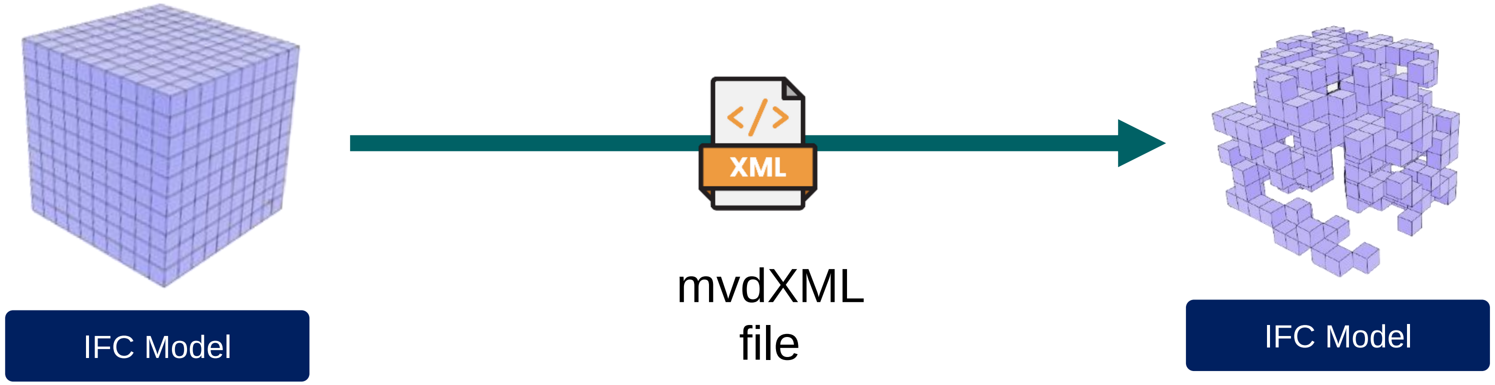
- What data is missing? “ThermalTransmittance” property is missing
- Are the values within the specified range? ThermalTransmittance > user-defined value

Why validate data?



ELASSTIC EU FP7 project (Berlo, Jagt, Walsum, Klein, & Müllers, 2016)

Current Approaches for Validating AEC Data



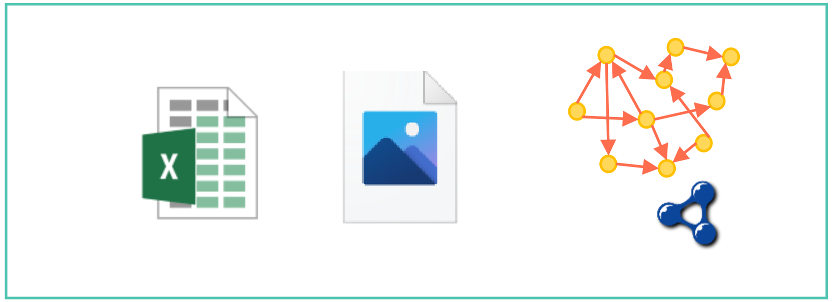
IFC Model

mvdXML file

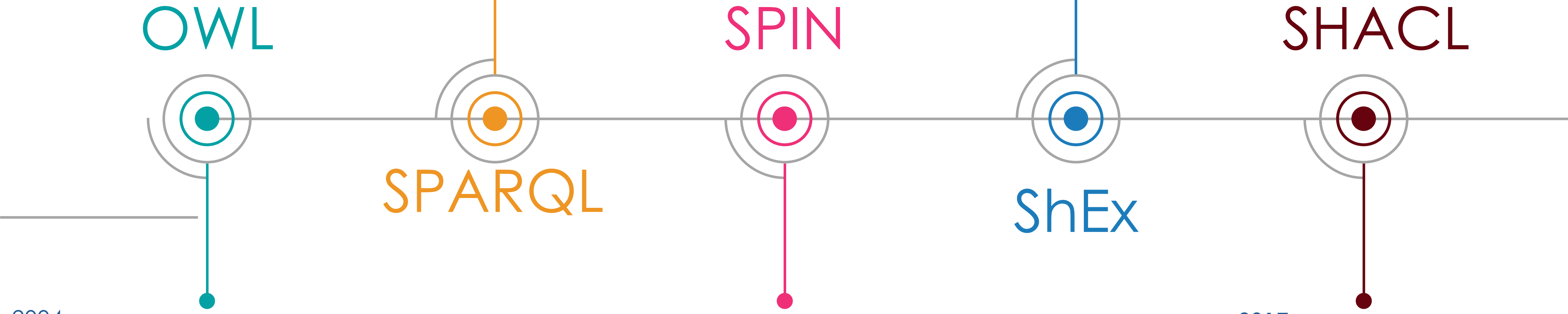
IFC Model

☑ IFC Data

How to check other kinds of linked data?



Before SHACL



OWL

2004
Properties - owl:maxCardinality
Designed for inferencing
Not actual data constraints
Describe inferences to be applied on constraints
OWA

SPARQL

2008 SPARQL Queries with conditions in the
WHERE clausequery returns asserted triples
Adhoc rules for data validation

SPIN

SPARQL Inference Notation
Inspired by SPARQL's data
validation popularity

ShEx

Shape Expressions 2013
Intended as schema for RDF Graphs
Main goal – human readability for
describing grammar of RDF graph

SHACL

2017
Merges XML resource shapes
SPIN
Dedicated constructs for riche
restriction definition

https://spinrdf.org/shacl-and-owl.html#:~:text=A%20difference%20between%20OWL%20and,for%20example%20using%20sh%3AtargetObjectsOf%20.

SHACL Scope: OWA and CWA

- OWA

- Absence of a certain statement does not mean that the statement is false

- EGsample
- When a triple does not have existence of a value, i.e. Owl:minCardinality 1 - > Not an error

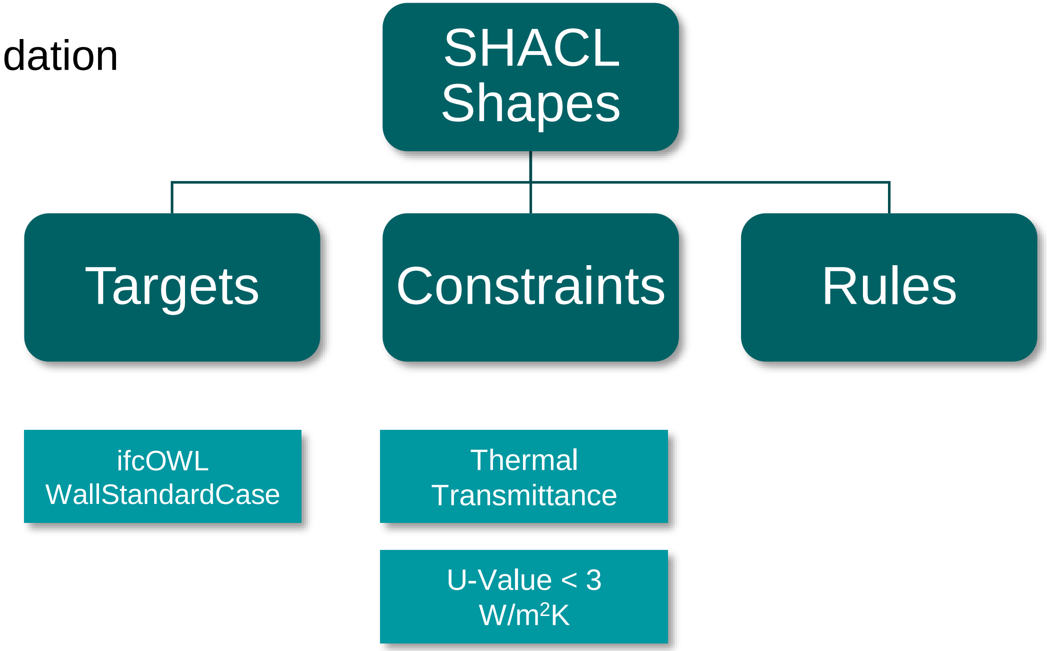
- More data can appear at any time, as in the Web open world, any RDF resource can link to another at any point.

- CWA

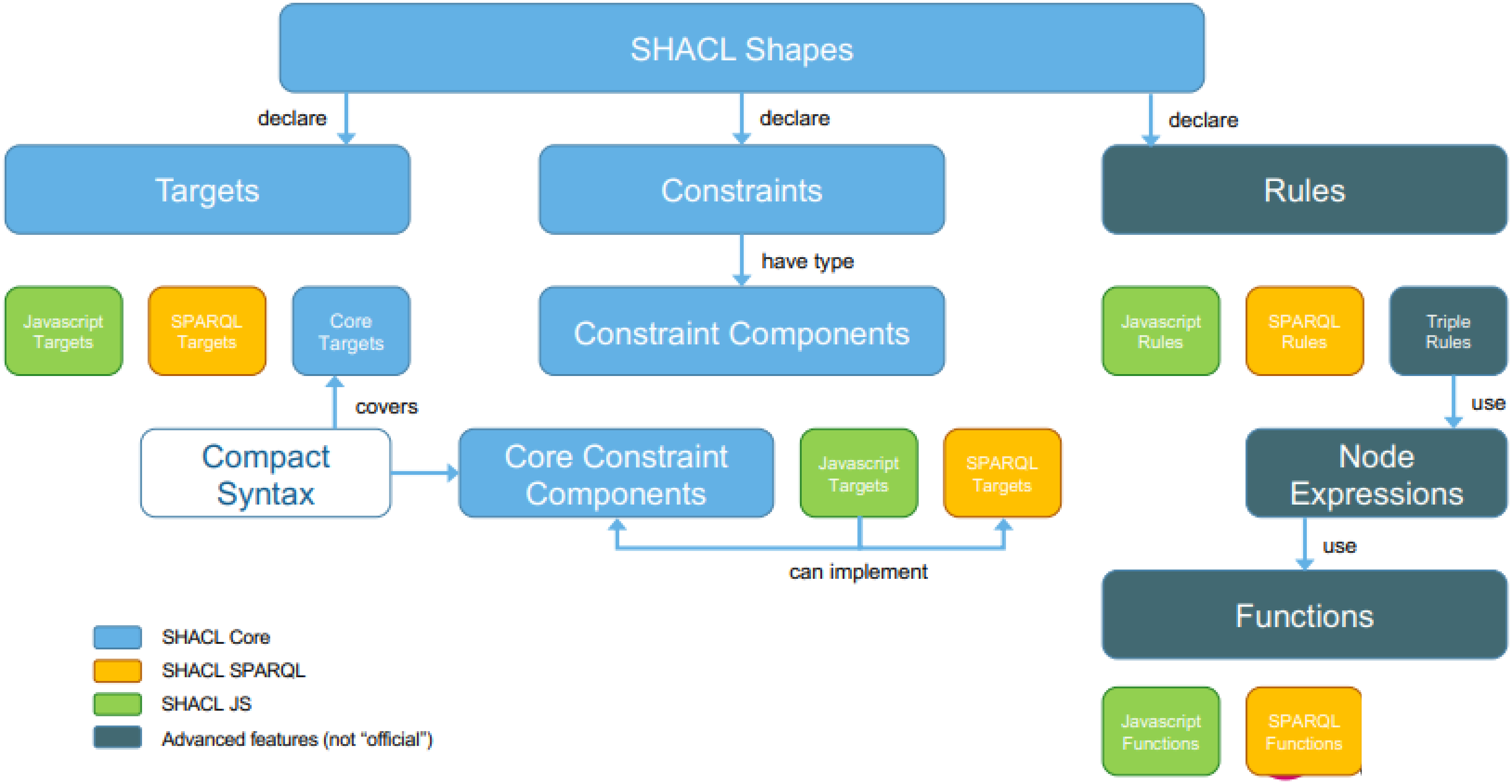
What is SHACL?

SHApes Constraint Language - W3C recommendation

- Inspiration: SPIN, OSLC, ShEx
- Shape: contains targets & constraints
- Targets: which nodes in data graph must conform to a SHACL shape
- Constraints: what kind of validation- cardinality, data type, etc.
- Shape Graphs: RDF graph containing shapes
- Data Graphs: RDF graph with data to be validated



Overview of SHACL (Core and Advanced Features)



[1]

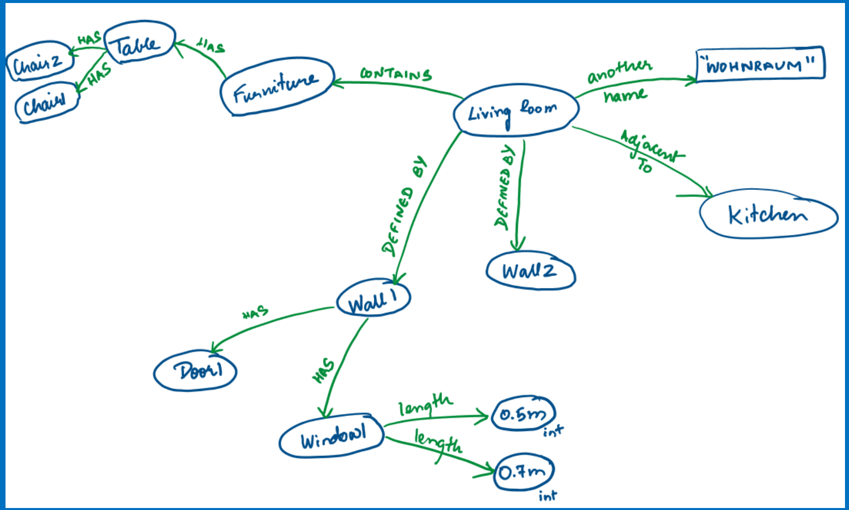
SHACL Constructs: What can be checked?

(1) Data existence, cardinality	Cardinality	All walls have a property “ThermalTransmittance/U-Value” All walls have only one value for the property “Thermal Transmittance”
(2) Data Type	Value types	Property “Thermal Transmittance” is an integer
(3) Values	Value ranges, Value restrictions, language constraints	Property “Thermal Transmittance” has value within 1-3 W/(m^2·K)
(4) Conditional Validity	Logical constraints, shape-based constraints, closed shapes	Property “Thermal Transmittance” exists for all Walls, only in the first floor
(5) Pattern Validation	SPARQL	Check that all instances of bot:Space have bot:SubElement associated with them, with the predicate pattern “bot:Space - > bot:containsElement -> bot:containsSubElement”

[2]

Hands-on Session - 1

Describe the room we are in – as a Graph



Instance Graph

What are the constraints we can put on the Graph nodes

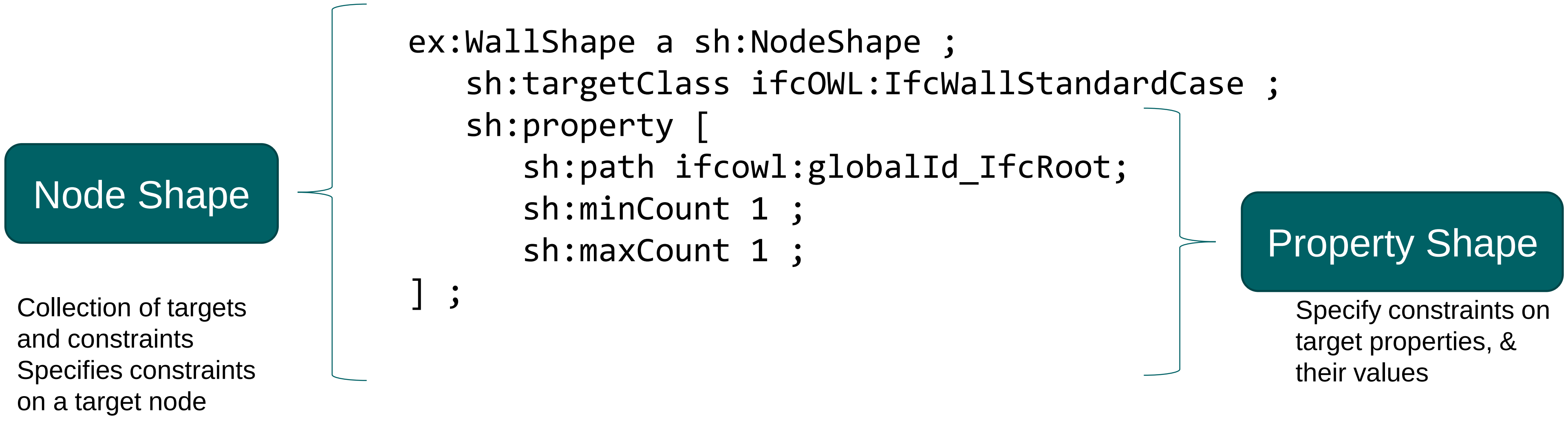
Write constraints in Natural Language

All walls should have only one GUID associated with it

Rule definition

SHACL Shape

Collection of targets and constraints:



Node Shape

Collection of targets and constraints Specifies constraints on a target node

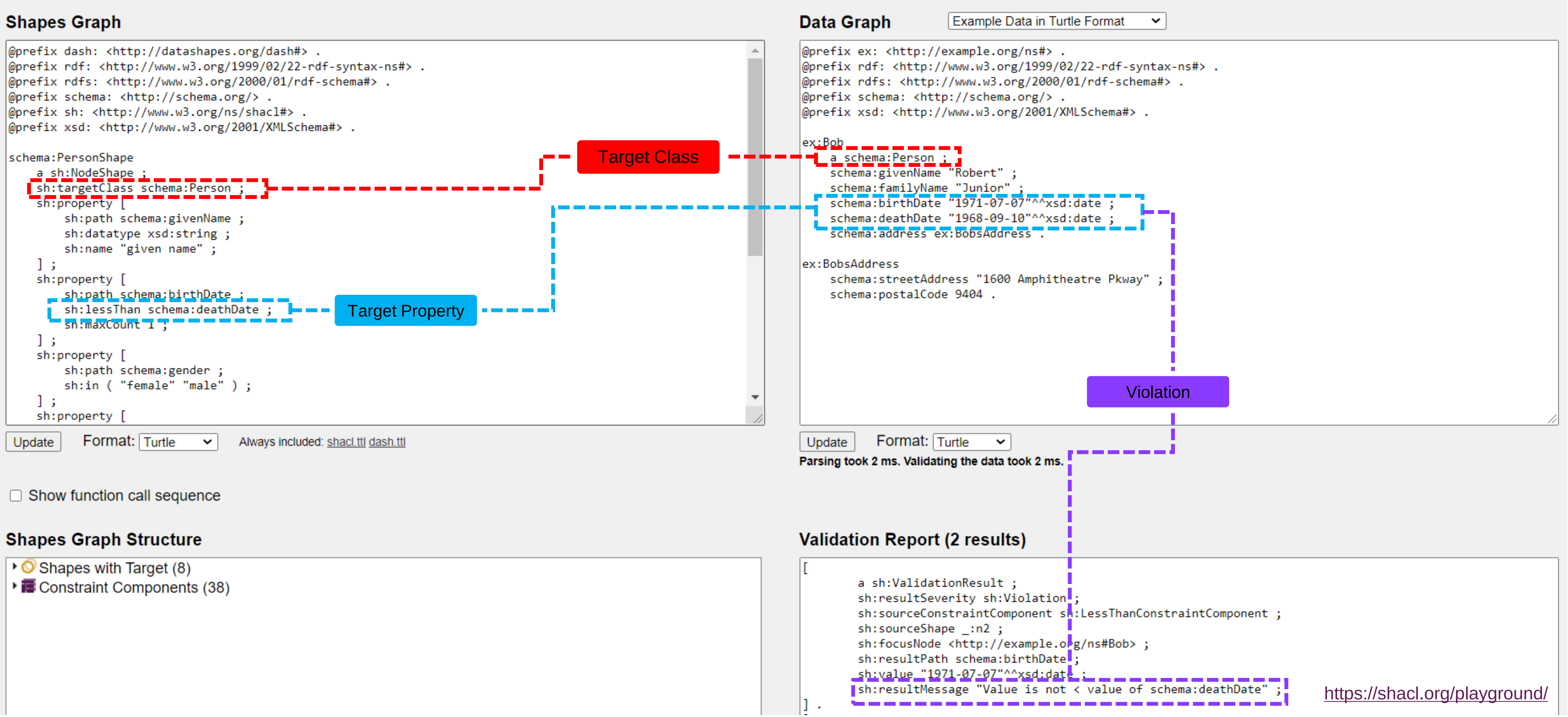
```
ex:WallShape a sh:NodeShape ;
   sh:targetClass ifcOWL:IfcWallStandardCase ;
   sh:property [
      sh:path ifcowl:globalId_IfcRoot;
      sh:minCount 1 ;
      sh:maxCount 1 ;
] ;
```

Property Shape

Specify constraints on target properties, & their values

- Semi-colon – uses the same subject as previously declared triple
- Comma – uses the same subject and predicate as the previously declared triple
- Parentheses - lists or collections
- Brackets - blank nodes (anonymous resources)

Anatomy of SHACL Shapes



Shapes Graph

```
@prefix dash: <http://datashapes.org/dash#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

schema:PersonShape
    a sh:NodeShape ;
    sh:targetClass schema:Person ;
    sh:property [
        sh:path schema:givenName ;
        sh:datatype xsd:string ;
        sh:name "given name" ;
    ] ;
    sh:property [
        sh:path schema:birthDate ;
        sh:lessThan schema:deathDate ;
        sh:maxCount 1 ;
    ] ;
    sh:property [
        sh:path schema:gender ;
        sh:in ( "female" "male" ) ;
    ] ;
    sh:property [
```

Format: Turtle   Always included: shacl.ttl dash.ttl

☐ Show function call sequence

Shapes Graph Structure

- Shapes with Target (8)
- Constraint Components (38)

Data Graph   Example Data in Turtle Format

```
@prefix ex: <http://example.org/ns#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

ex:Bob
    a schema:Person ;
    schema:givenName "Robert" ;
    schema:familyName "Junior" ;
    schema:birthDate "1971-07-07"^^xsd:date ;
    schema:deathDate "1968-09-10"^^xsd:date ;
    schema:address ex:BobsAddress .

ex:BobsAddress
    schema:streetAddress "1600 Amphitheatre Pkway" ;
    schema:postalCode 9404 .
```

Format: Turtle

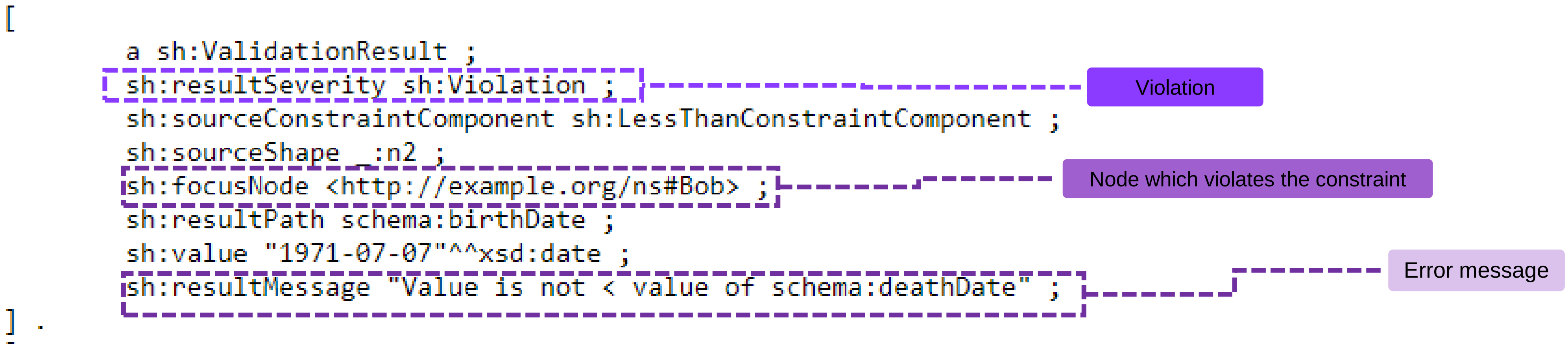
Parsing took 2 ms. Validating the data took 2 ms.

Validation Report (2 results)

```
[
        a sh:ValidationResult ;
        sh:resultSeverity sh:Violation ;
        sh:sourceConstraintComponent sh:LessThanConstraintComponent ;
        sh:sourceShape _:n2 ;
        sh:focusNode <http://example.org/ns#Bob> ;
        sh:resultPath schema:birthDate ;
        sh:value "1971-07-07"^^xsd:date ;
        sh:resultMessage "Value is not < value of schema:deathDate" ;
] .
```

https://shacl.org/playground/

SHACL Validation Report



```
[
        a sh:ValidationResult ;
        sh:resultSeverity sh:Violation ;
        sh:sourceConstraintComponent sh:LessThanConstraintComponent ;
        sh:sourceShape _:n2 ;
        sh:focusNode <http://example.org/ns#Bob> ;
        sh:resultPath schema:birthDate ;
        sh:value "1971-07-07"^^xsd:date ;
        sh:resultMessage "Value is not < value of schema:deathDate" ;
] .
```

Violation

Node which violates the constraint

Error message

https://shacl.org/playground/

Hands-on Session - 2

Write constraints in Natural Language

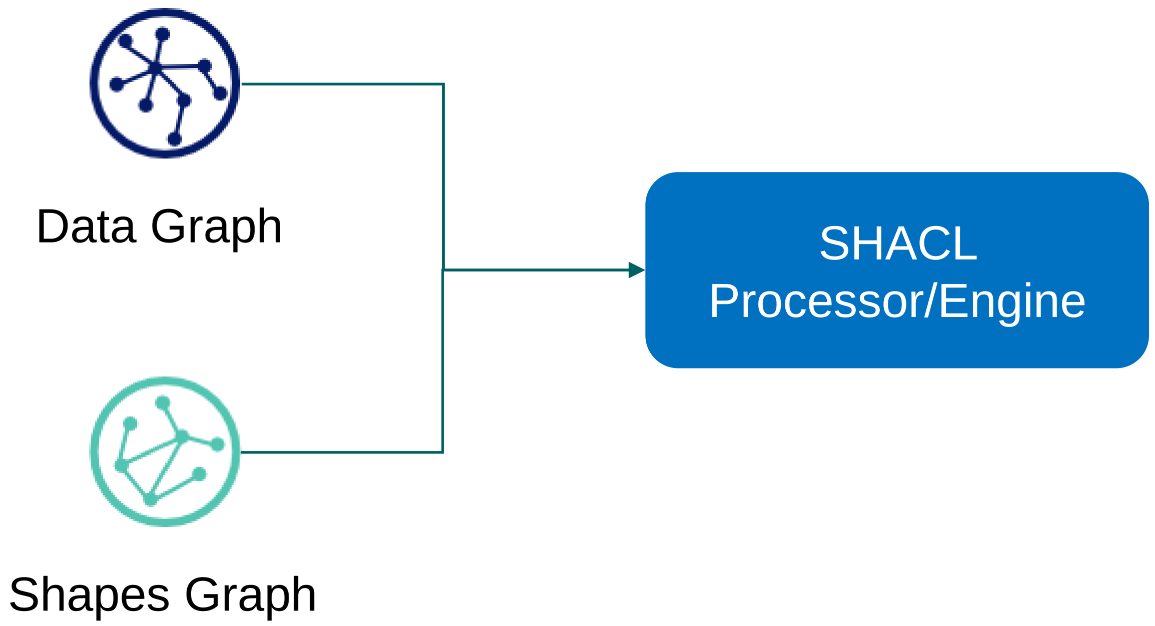
All walls should have only one GUID associated with it

Rule definition

Translate constraints to a SHACL Shape

```
ifcowl:WallShape a sh:NodeShape ;
   sh:targetClass ifcowl:IfcWallStandardCase ;
   sh:property [
      sh:path ifcowl:globalId_IfcRoot;
      sh:minCount 1;
      sh:maxCount 1;
].
```

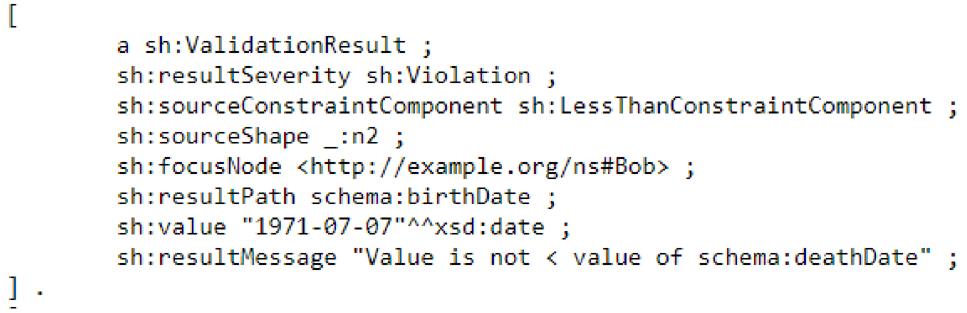
SHACL Validators and Processors



Data Graph

Shapes Graph

SHACL Processor/Engine



```
[
        a sh:ValidationResult ;
        sh:resultSeverity sh:Violation ;
        sh:sourceConstraintComponent sh:LessThanConstraintComponent ;
        sh:sourceShape _:n2 ;
        sh:focusNode <http://example.org/ns#Bob> ;
        sh:resultPath schema:birthDate ;
        sh:value "1971-07-07"^^xsd:date ;
        sh:resultMessage "Value is not < value of schema:deathDate" ;
] .
```

Online Validators

- SHACL Playground
- SHACL Play! (sparna.fr)
- SHACL Online Editor (liu.se)
- SHACL Validator (europa.eu)
- SHACL Playground by Zazuko

Hands-on Session - 3

Translate constraints to a SHACL Shape

```
ex:WallShape a sh:NodeShape ;
   sh:targetClass ifcOWL:IfcWallStandardCase ;
   sh:property [
      sh:path ifcowl:globalId_IfcRoot;
      sh:minCount 1;
      sh:maxCount 1;
] ;
```

Validation

```
[
 a sh:ValidationResult ;
 sh:resultSeverity sh:Violation ;
 sh:sourceConstraintComponent
sh:MinCountConstraintComponent ;
 sh:sourceShape _:n902 ;
 sh:focusNode inst:IfcWallStandardCase_92 ;
 sh:resultPath ifcowl:globalId_IfcRoot ;
 sh:resultMessage "Less than 1 values" ;
] .
```

Describe the instance graph – snippet of the duplex model

```
inst:IfcWallStandardCase_91
  rdf:type  ifcowl:IfcWallStandardCase;
  ifcowl:globalId_IfcRoot inst:IfcGloballyUniqueId_27659.

inst:IfcWallStandardCase_92
  rdf:type ifcowl:IfcWallStandardCase ;
  ifcowl:globalId_IfcRoot inst:IfcGloballyUniqueId_27663;
  ifcowl:globalId_IfcRoot inst:IfcGloballyUniqueId_12365.
```

Target constraints

Applied to Node Shapes - specify the nodes that are to be validated against some property shape

Ways to define target:

- targetClass – All instances of a class
- targetNode – Specific nodes
- targetObjectsOf – All object of a specific property
- targetSubjectsOf – All subjects of a specific property

```
ex:WindowShape1 a sh:NodeShape ;
        sh:targetClass bot:Window .
ex:WindowShape2 a sh:NodeShape ;
        sh:targetNode bot:opening bot:window.
ex:WindowShape3 a sh:NodeShape ;
        sh:targetSubjectsOf bot:window .
```

Datatype Constraints

(2) Data Type

Property “Thermal Transmittance” is a float

```
ex:WallShape a sh:NodeShape ;
   sh:targetClass ifcOWL:IfcWallStandardCase ;
   sh:property [
      sh:path ns1:thermalTransmittancevalue;
      sh:datatype xsd:float;
] ;
```

```
inst:IfcWallStandardCase_91
  rdf:type  ifcowl:IfcWallStandardCase;
  ns1:thermalTransmittanceValue “2.1”^xsd:string.

inst:IfcWallStandardCase_92
  rdf:type ifcowl:IfcWallStandardCase ;
  ns1:thermalTransmittancevalue 1.8^xsd:float.
```

Value Constraints

(3) Values

Property “Thermal Transmittance” has value within 1-3 W/(m^2·K)

```
ex:WallShape a sh:NodeShape ;
   sh:targetClass ifcOWL:IfcWallStandardCase ;
   sh:property [
      sh:path ns1:thermalTransmittancevalue;
      sh:minValue 1;
      sh:maxValue 3;
] ;
```

```
inst:IfcWallStandardCase_91
   rdf:type  ifcowl:IfcWallStandardCase ;
   ns1:thermalTransmitancevalue 3.5^xsd:float.

inst:IfcWallStandardCase_92
        rdf:type  ifcowl:IfcWallStandardCase ;
        ns1:thermalTransmittancevalue 1.8^xsd:float.
```

SPARQL Constraints

(5) Pattern Validation

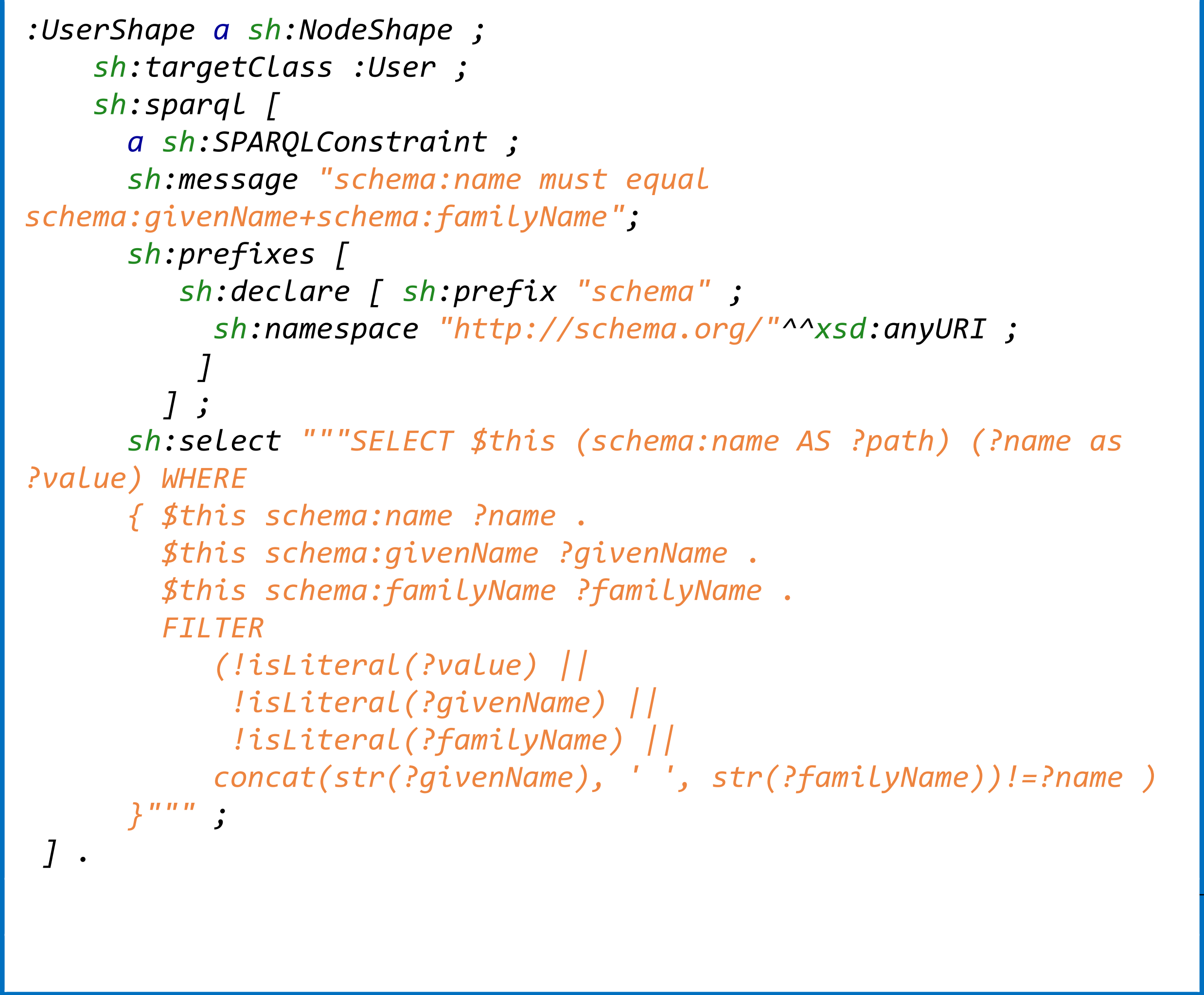
Check that all instances of bot:Space have bot:SubElement associated with them, with the predicate pattern “bot:Space - > bot:containsElement -> bot:containsSubElement”

```
ex:WallShape a sh:Node
   sh:targetClass 
ifcOWL:IfcWallStandardCase ;
   sh:sparql [
 a sh:SPARQLConstraint;
sh:message “global id not present!”;
sh:prefix[
Sh:declare[
sh:prefix “ifcowl”;
sh:namespace “”;
]
];
sh:select
“”””
SELECT …..
WHERE {
}
“”””;
].
```

```
inst:IfcWallStandardCase_91
   rdf:type ifcowl:IfcWallStandardCase ;
   ifcowl:globalId_IfcRoot inst:IfcGloballyUniqueId_27659 .

inst:IfcGloballyUniqueId_27659
   rdf:type ifcowl:IfcGloballyUniqueId ;
   express:hasString "2O2Fr$t4X7Zf8NOew3FL9r" .

inst:IfcWallStandardCase_92
        rdf:type  ifcowl:IfcWallStandardCase ;
        ifcowl:globalId_IfcRoot inst:IfcGloballyUniqueId_27663.

inst:IfcGloballyUniqueId_27663
        rdf:type ifcowl:IfcGloballyUniqueId ;
        express:hasString "2O2Fr$t4X7Zf8NOew3FLIE" .
```

Hands-on Session - 3

Translate constraints to a SHACL Shape



```
:UserShape a sh:NodeShape ;
    sh:targetClass :User ;
    sh:sparql [
      a sh:SPARQLConstraint ;
      sh:message "schema:name must equal
schema:givenName+schema:familyName";
      sh:prefixes [
         sh:declare [ sh:prefix "schema" ;
           sh:namespace "http://schema.org/"^^xsd:anyURI ;
          ]
        ] ;
      sh:select """SELECT $this (schema:name AS ?path) (?name as
?value) WHERE
      { $this schema:name ?name .
        $this schema:givenName ?givenName .
        $this schema:familyName ?familyName .
        FILTER
           (!isLiteral(?value) ||
            !isLiteral(?givenName) ||
            !isLiteral(?familyName) ||
           concat(str(?givenName), ' ', str(?familyName))!=?name )
      }""" ;
 ] .
```

Describe the instance graph – snippet of the duplex model

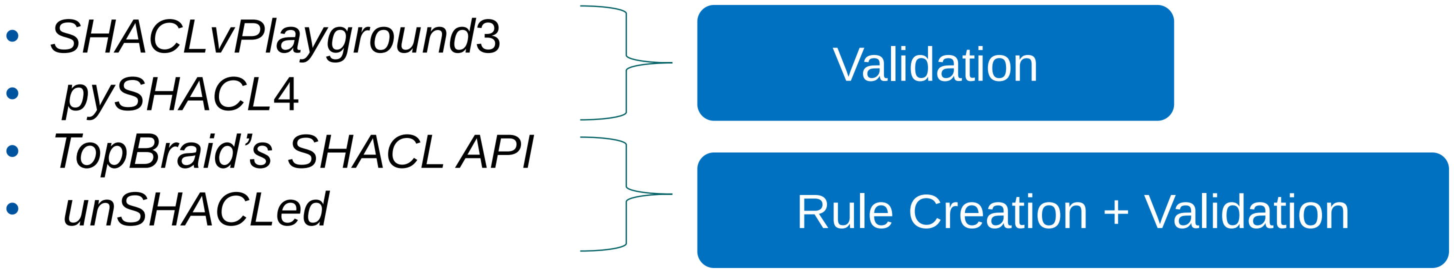
```
:alice a :User ; #Passes as a :UserShape
   schema:givenName "Alice" ;
   schema:familyName "Cooper" ;
   schema:name "Alice Cooper" .

:bob a :User ; #Fails as a :UserShape
   schema:givenName "Bob" ;
   schema:familyName "Smith" ;
   schema:name "Robert Smith" .
```

SHACL Libraries and tooling

- RDF4J
- rdfLIB
- Topbraid
- pySHACL
- Protégé
- …
- …

Existing Implementations

SHACL



- *SHACLvPlayground3*
- *pySHACL4*
- *TopBraid's SHACL API*
- *unSHACLed*

Validation

Rule Creation + Validation

expert semantic web knowledge

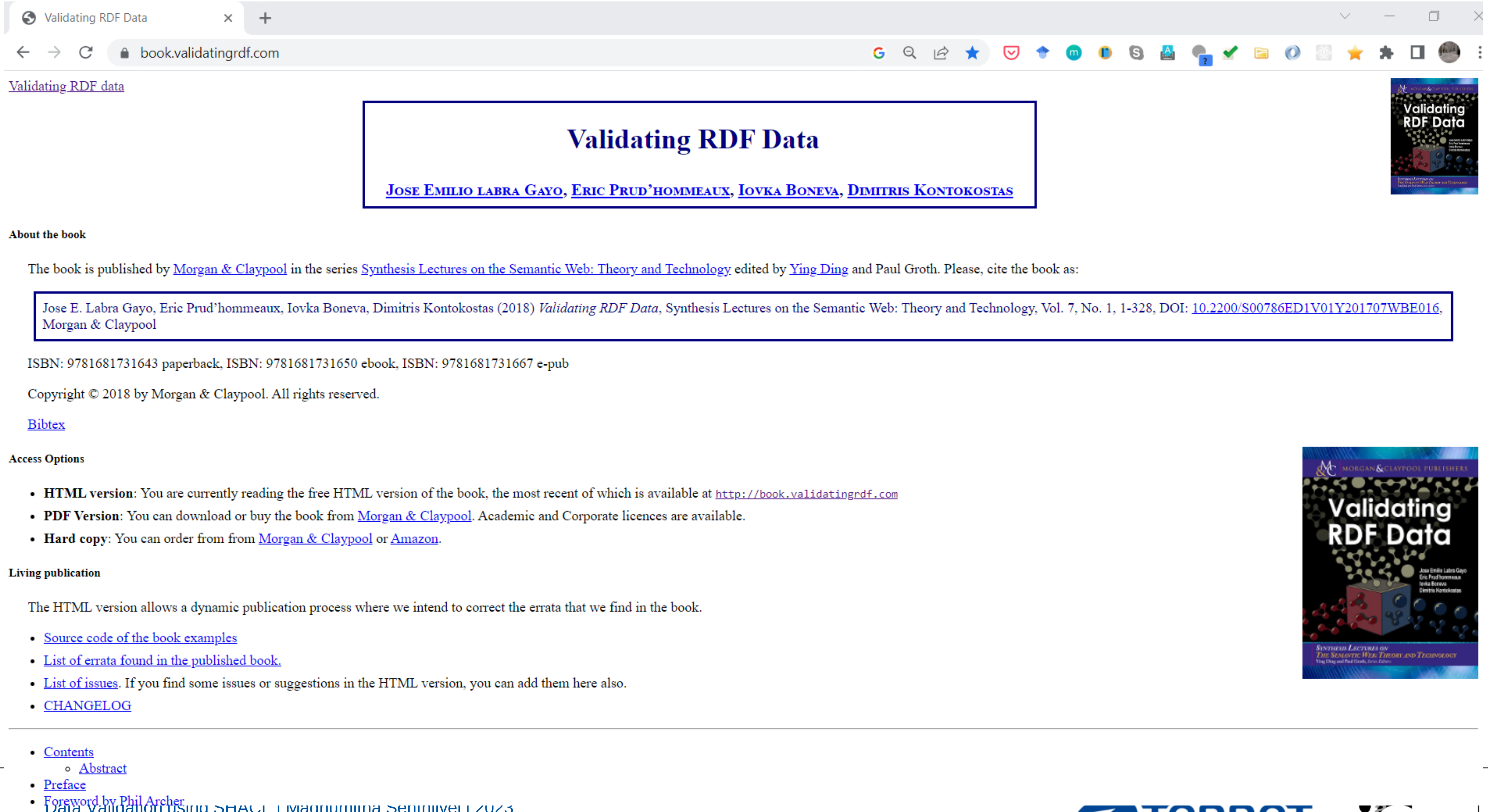
+

knowledge on assessing the applicability of these rules for AEC ontologies and use cases

OWA vs CWA

- SHACL operates under Closed World Assumptions supported by:
  - Comprehensive constraint vocabulary
  - Includes error reports in RDF, with dedicated terminologies
  - Distributed/Federated graphs query
  - Javascript functions for advanced features

Additional reading



Validating RDF data

Validating RDF Data

Jose Emilio labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas

About the book

The book is published by Morgan & Claypool in the series Synthesis Lectures on the Semantic Web: Theory and Technology edited by Ying Ding and Paul Groth. Please, cite the book as:

> Jose E. Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas (2018) *Validating RDF Data*, Synthesis Lectures on the Semantic Web: Theory and Technology, Vol. 7, No. 1, 1-328, DOI: 10.2200/S00786ED1V01Y201707WBE016, Morgan & Claypool

ISBN: 9781681731643 paperback, ISBN: 9781681731650 ebook, ISBN: 9781681731667 e-pub

Copyright © 2018 by Morgan & Claypool. All rights reserved.

Bibtex

Access Options

- **HTML version**: You are currently reading the free HTML version of the book, the most recent of which is available at `http://book.validatingrdf.com`
- **PDF Version**: You can download or buy the book from Morgan & Claypool. Academic and Corporate licences are available.
- **Hard copy**: You can order from from Morgan & Claypool or Amazon.

Living publication

The HTML version allows a dynamic publication process where we intend to correct the errata that we find in the book.

- Source code of the book examples
- List of errata found in the published book.
- List of issues. If you find some issues or suggestions in the HTML version, you can add them here also.
- CHANGELOG

Thanks for your attention!

Design Computation

RWTH AACHEN UNIVERSITY

References

- [1] Luiz Bonino 2022 “SHAPES CONSTRAINT LANUGAGE - SHACL”, University of Twente
- [2] https://www.w3.org/TR/shacl/
- [3] Gayo et. al 2021 Validating RDF data.

Hands-on Session - 4

Translate constraints to a SHACL Shape

```
SELECT ?s ?val2
WHERE {
?s rdf:type ifcowl:IfcWallStandardCase;
?s ifcowl:globalId_IfcRoot ?val2;
?val2 rdf:type ifcowl:IfcGloballyUniqueId;
?val2 express:string ?o3;
}
```

Validation

Describe the room we are in – as a Graph

```
inst:IfcWallStandardCase_91
   rdf:type ifcowl:IfcWallStandardCase ;
   ifcowl:globalId_IfcRoot
inst:IfcGloballyUniqueId_27659 .

inst:IfcGloballyUniqueId_27659
   rdf:type ifcowl:IfcGloballyUniqueId ;
   express:hasString "2O2Fr$t4X7Zf8NOew3FL9r" .

inst:IfcWallStandardCase_92
        rdf:type  ifcowl:IfcWallStandardCase ;
        ifcowl:globalId_IfcRoot
inst:IfcGloballyUniqueId_27663.

inst:IfcGloballyUniqueId_27663
        rdf:type ifcowl:IfcGloballyUniqueId ;
        express:hasString "2O2Fr$t4X7Zf8NOew3FLIE".
```