VALIDATION OF BUILDING MODELS AGAINST LEGISLATION USING SHACL

Emma Nuyts* - Jeroen Werbrouck - Ruben Verstraeten - Louise Deprez

LDAC2023

UNIVERSITEIT GENT

DigiChecks

EMMA NUYTS

- Internship at Bricsys (summer 2019 & 2020)

- Master of Science in Engineering: Architecture (june 2022, Ghent University)
  Masterthesis on *Automated validation of building models against legislation using Linked Data*

- Assisting academic staff at Ghent University (september 2022-…)

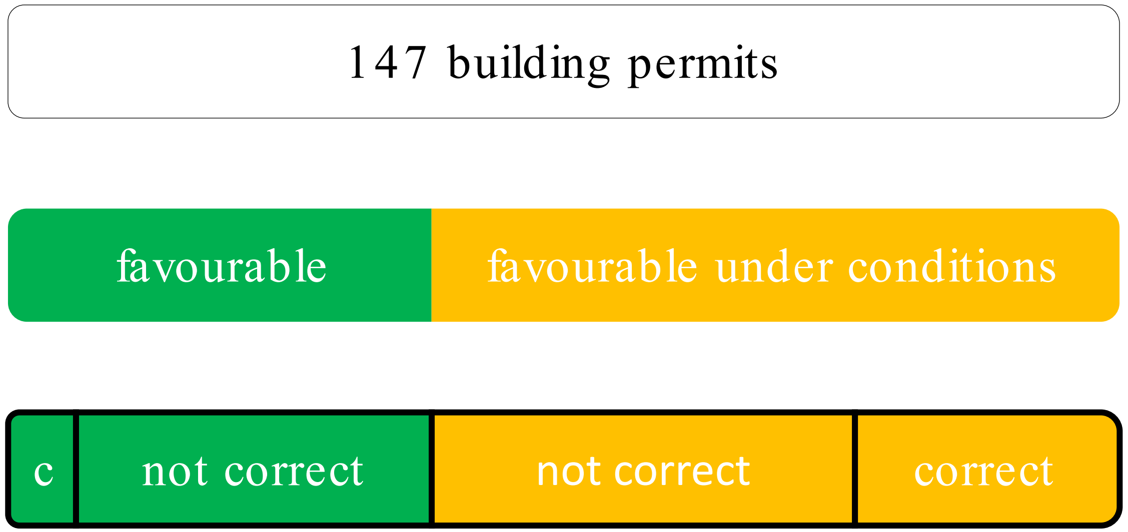
- Project engineer at Arch&Teco (september 2022-…)

emma.nuyts@ugent.be

PRESENTATION OVERVIEW

- Introduction
- Related work
- Building validation using SHACL
- Proof of concept
- Discussion

INTRODUCTION



(Inter, 2019)

RELATED WORK

STEPS IN THE ACC PROCESS

1. Interpretation of normative knowledge
2. Mapping applicabilities to ontologies
3. Creating machine-readable constraints

- Natural Language Processing (NLP) in combination with DMN
- Mark-up language, like RASE
- New methodology or framework

RELATED WORK

STEPS IN THE ACC PROCESS

1. Interpretation of normative knowledge
2. Mapping applicabilities to ontologies
3. Creating machine-readable constraints

A stair with a riser height and a tread length

A beo:StairFlight with a props:riserHeight IfcStairFlight and a props:treadLengthIfcStairFlight

cfr. LD-BIM web app (Rasmussen & Schlachter)

RELATED WORK

STEPS IN THE ACC PROCESS

1. Interpretation of normative knowledge
2. Mapping applicabilities to ontologies
3. Creating machine-readable constraints (Pauwels & Zhang)

- Hard-coded requirements
- Rule-checking by querying
- Dedicated rule language, like SWRL, N3 Logic...
- Shapes Constraints Language (SHACL)

BUILDING VALIDATION USING SHACL

SHACL allows checking procedures on building models converted to LBD ontologies (Oraskari, 2021)

Semantic Web rules (SWRL, SHACL) would increase the usage of KBs in AEC industry (Elshani, 2022)

Quality assurance of properties can be conducted using SHACL shapes (Zentgraf, 2022)

Using SHACL for compliance checking, by showing the shape for checking the thermal transmittance of a window (Kovacs & Micsik, 2021)

Can SHACL be used to evaluate more complex constraints for compliance checking purposes?

BUILDING VALIDATION USING SHACL

USE CASE ON ACCESSIBILITY

- Ensuring model quality
- Value constraints
- Relational constraints
- Mathematical constraints
- Conditional statements

1. Interpretation of normative knowledge: RASE
2. Mapping applicabilities to ontologies: manually
3. Creating machine-readable constraints: SHACL

BUILDING VALIDATION USING SHACL

ENSURING MODEL QUALITY

```
ex:DoorProperties
  a sh:NodeShape ; #apply the shape to a focus node
  sh:targetClass beo:Door ; #target all nodes with class 'Door'
  sh:property [ #target a property of each 'Door'
    sh:path props:overallHeightIfcDoor ; #target the height predicate
    sh:property ex:DoorHeightProperty ; #name the object of this predicate path
    sh:minCount 1 ; #each 'Door' should have at least one height property
    sh:maxCount 1 ; #each 'Door' should have at most one height property
    sh:message "Each door should have exactly one overallHeightIfcDoor" ; ] .

ex:DoorHeightProperty
  a sh:PropertyShape ; #target a property of the focus node
  sh:path schema:value ; #target the value predicate
  sh:minCount 1 ; #each 'overallHeightIfcDoor' property should have at least one value
  sh:maxCount 1 ; #each 'overallHeightIfcDoor' property should have at most one value
  sh:message "Each doorheight should have exactly one value" .
```

BUILDING VALIDATION USING SHACL

VALUE CONSTRAINT

```
For <a>entrances or doorways</a>, a <a>clear passage height</a> of <r>at least 2.09
    meters</r> must be guaranteed after finishing.
```

Article	Type	Applicability	Ontology
Art.22 §1	class	entrances or doorways	beo:Door
	property	clear passage height	props:overallHeightIfcDoor

```
ex:Door
  a sh:NodeShape ; #apply the shape to a focus node
  sh:targetClass beo:Door ; #target all nodes with class 'Door'
  sh:property [ #target a property of each 'Door'
    sh:path props:overallHeightIfcDoor ; #target the height predicate
    sh:property ex:DoorHeight ; ] . #name the object of this predicate path

ex:DoorHeight #the named object is now a subject
  a sh:PropertyShape ; #target a property of the focus node
  sh:path schema:value ; #target the value predicate
  sh:minInclusive "2.09"^^xsd:double ; #the object should be more than 2.09 m
  sh:message "Art.22.1: For entrances or doorways, a clear passage height of at least
      2.09 meters must be guaranteed after finishing" .
```

BUILDING VALIDATION USING SHACL

RELATIONAL CONSTRAINT

```
A <a>railing</a> should be fitted on <r>both sides</r> of the <a>stair</a> ...
```

Article	Type	Applicability	Ontology
Art.20 §4	class	stair	beo:Stair
	class	railing	beo:Railing

```
ex:Stair
  a sh:NodeShape ; #apply the shape to a focus node
  sh:targetClass beo:Stair ; #target all nodes with class 'Stair'
  sh:property [ #target a property of each 'Stair'
    sh:path bot:hasSubElement ; #target the subelement predicate
    sh:qualifiedValueShape [sh:class beo:Railing] ; #target the class 'Railing'
    sh:qualifiedMinCount 2 ; #two elements of this class should be present
    sh:message "Art.20.4: A railing should be fitted on both sides of the stair" ; ] .
```

BUILDING VALIDATION USING SHACL

MATHEMATICAL CONSTRAINT

```
... the <r>sum of twice the <a>riser</a> and once the <a>tread</a> of each step must
    be between 57 cm and 63 cm</r> ...
```

Article	Type	Applicability	Ontology
Art.20 §3	property	riser	props:riserHeightIfcStairFlight
	property	tread	props:treadLengthIfcStairFlight

```
ex:StairFormula
  a sh:SPARQLFunction ; #define a function
  sh:parameter [ #declare the first parameter
    sh:path ex:riser ; #first parameter is the riser height
    sh:datatype xsd:double ; #the riser is a decimal
  ] ;
  sh:parameter [ #declare the second parameter
    sh:path ex:tread ; #second parameter is the tread length
    sh:datatype xsd:double ; #the tread is a decimal
  ] ;
  sh:returnType xsd:double ; #the returned value is a decimal
  sh:select """ #start a SPARQL select query
    SELECT ( (2 * $riser + $tread) AS ?result)
    WHERE {
    }
    """ .
```

BUILDING VALIDATION USING SHACL

MATHEMATICAL CONSTRAINT <> VALUE CONSTRAINT

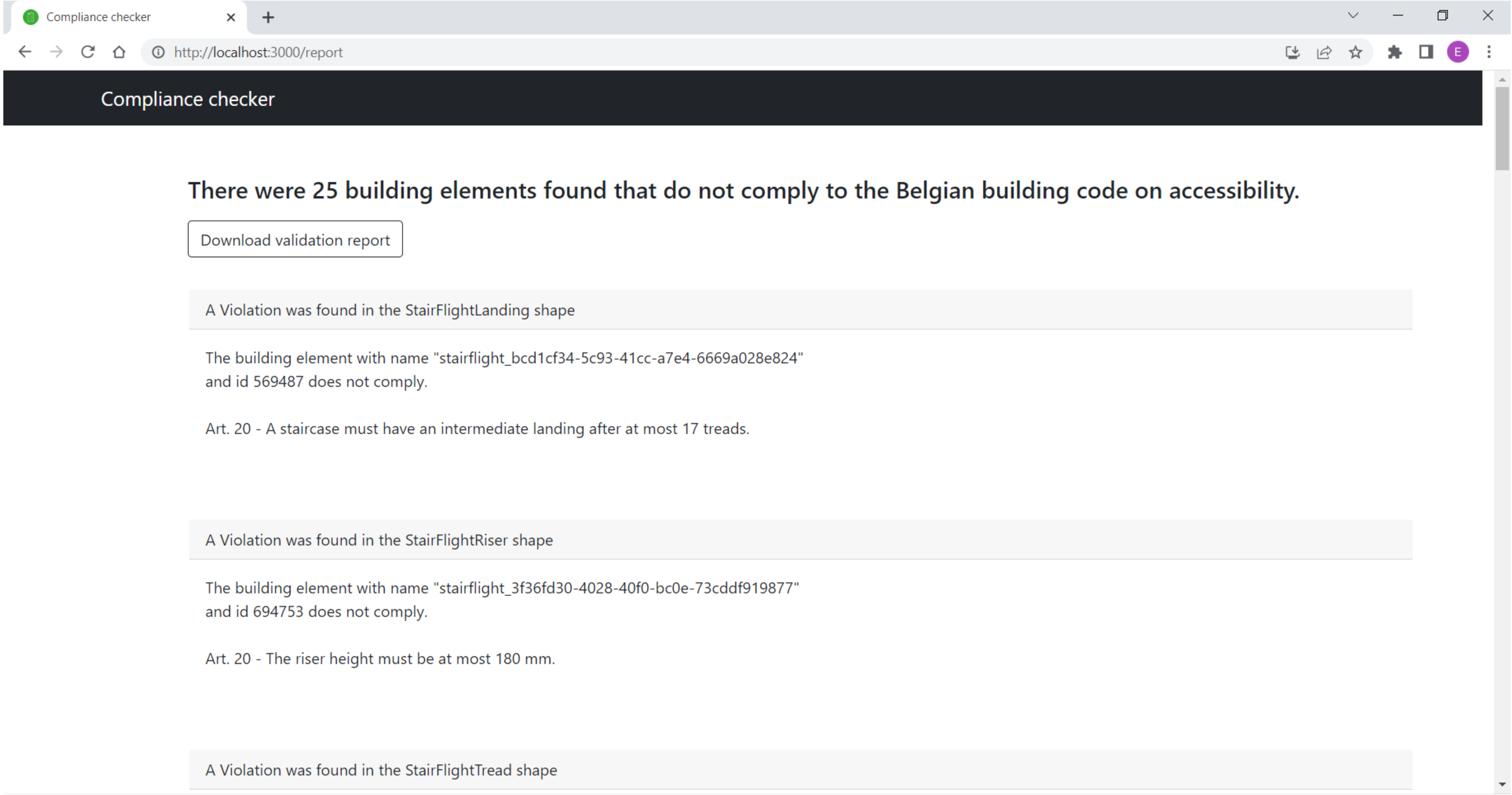
```
ex:StairSlope
  a sh:NodeShape ; #apply the shape to a focus node
  sh:targetClass beo:StairFlight ; #target all nodes with class 'StairFlight'
  sh:message "Art.20.3: the sum of twice the riser and once the tread of each step
      must be higher than 0.57 m." ;
  sh:expression [
    ex:LessThan ( #refer to the LessThan(a,b) function
      0.57 #first parameter of LessThan
      [ #second parameter of LessThan
        ex:StairFormula ( #refer to the StairFormula(riser, tread) function
          [sh:path (props:riserHeightIfcStairFlight schema:value)]
          [sh:path (props:treadLengthIfcStairFlight schema:value)])])] .
```

BUILDING VALIDATION USING SHACL

CONDITIONAL STATEMENT

```
ex:Slopes
  a sh:NodeShape ; #apply the shape to a focus node
  sh:targetClass beo:RampFlight ; #target all nodes with class 'Rampflight'
  sh:message "Art.19.1: The slope of a rampflight is maximally ten percent for level
      differences of up to 0.10 m" ;
  sh:and ( #both constraints should be fulfilled
   [sh:expression [ #first constraint: the slope is less than 10%
      ex:LessThan (
         [ex:Slope (
            [sh:path (props:heightIfcRampFlight schema:value)]
            [sh:path (props:lengthIfcRampFlight schema:value)])]
         10)]]
   sh:node ex:SlopeConstraint) . #if the slope is less than 10%, check the next
        constraint
```

PROOF OF CONCEPT



DISCUSSION

- The computer-programmed rules must be easily understood by the regulation authors;
- The lifecycle of the rule base must be independent of software and schema updates;
- All development must comply with Open Standards;
- Consideration must be given to the industry processes of model authoring.

(Greenwood et al.)

LIMITATIONS & FUTURE WORK

- Shapes are dependant on modeling complexity of RDF graph (L1 -L3)
- Low-level functions are needed ('LessThan')
- Only prescriptive legislation
- Consistent unit system needed or needs to be checked

- Compliance of geometry/relative positioning of elements
- Automation of the SHACL shapes
- Visual programming SHACL shapes creator (Senthilvel & Beetz)

REFERENCES

Inter, Evaluatieonderzoek Vlaamse Toegankelijkheidsverordening, Final Report, Agentschap Toegankelijk Vlaanderen, 2019.

M. H. Rasmussen, Schlachter, LD-BIM, 2023. URL: https://ld-bim.web.app/.

P. Pauwels, S. Zhang, Semantic Rule-checking for Regulation Compliance Checking: An Overview of Strategies and Approaches, in: 32nd CIB W78 Conference, Eindhoven, Netherlands, 2015, pp. 619–628.

J. Oraskari, M. Senthilvel, J. Beetz, SHACL is for LBD what mvdXML is for IFC, in: The 38th CIB W78 conference on Information and Communication Technologies for AECO, Luxembourg, 2021, pp. 693–702.

D. Elshani, A. Lombardi, A. Fisher, S. Staab, D. Hernández, Inferential Reasoning in Co-Design Using Semantic Web Standards alongside BHoM, in: 33. Forum Bauinformatik, München, Germany, 2022, pp. 89–97.

REFERENCES

S. Zentgraf, P. Hagedorn, M. König, Multi-requirements ontology engineering for automated processing of document-based building codes to linked building data properties, IOP Conference Series: Earth and Environmental Science 1101 (2022).

A. T. Kovacs, A. Micsik, BIM quality control based on requirement linked data, International Journal of Architectural Computing 19 (2021) 431–448.

D. Greenwood, S. Lockley, S. Malsane, J. Matthews, Automated compliance checking using building information models, in: The construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors, Dauphine Université, Paris, 2010.

M. Senthilvel, J. Beetz, A Visual Programming Approach for Validating Linked Building Data, in: EG-ICE 2020 Workshop on Intelligent Computing in Engineering, Berlin, Germany, 2020, pp. 403–411.

UNIVERSITEIT GENT

DigiChecks

fwo

TOWARDS AN ACCESSIBLE BUILT ENVIRONMENT