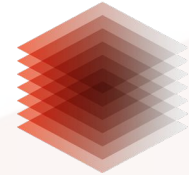


LEIBNIZ-INFORMATIONSZENTRUM  
TECHNIK UND NATURWISSENSCHAFTEN  
UNIVERSITÄTSBIBLIOTHEK



TIB

# Representing Normative Regulations in OWL DL for Automated Compliance Checking Supported by Text Annotation

Ildar Baimuratov, Denis Turygin

LDAC 2025

# Problems

**Manual** building compliance checking:

- Regulations are in **natural language** X
- Requires significant **time** and **expertise** X
- Prone to **human error** X
- **Redundancy** and **inconsistency** of building codes X
- **Hardcoded** model checkers (Solibri) are difficult to develop and maintain X
- **AI** is inherently unreliable (PAC learning\*) X

**Symbolic** reasoning:

- Absolutely **accurate** ✓
- **Explainable** ✓
- Formalizing regulations is **timely** and **skill-intensive** X

\*Probably approximately correct

# Representation of regulations

Language	Approach
SPARQL	Yurchyshyna & Zarli, 2009
N3-Logic	Pauwels et al., 2011
SWRL	Beach et al., 2015; Fahad et al., 2016; Shi et al., 2017
Prolog	Zhang&El-Gohary, 2017
SHACL	Stolk&McGlenn, 2020; Kovacs&Micsik, 2021; Zentgraf et al., 2022; Nuyts et al., 2023; Patlakas et al., 2024; Donkers&Petrova, 2024
DL query	Fitkau & Hartmann, 2024
OWL*	Nuyts et al., 2024

\*only information availability

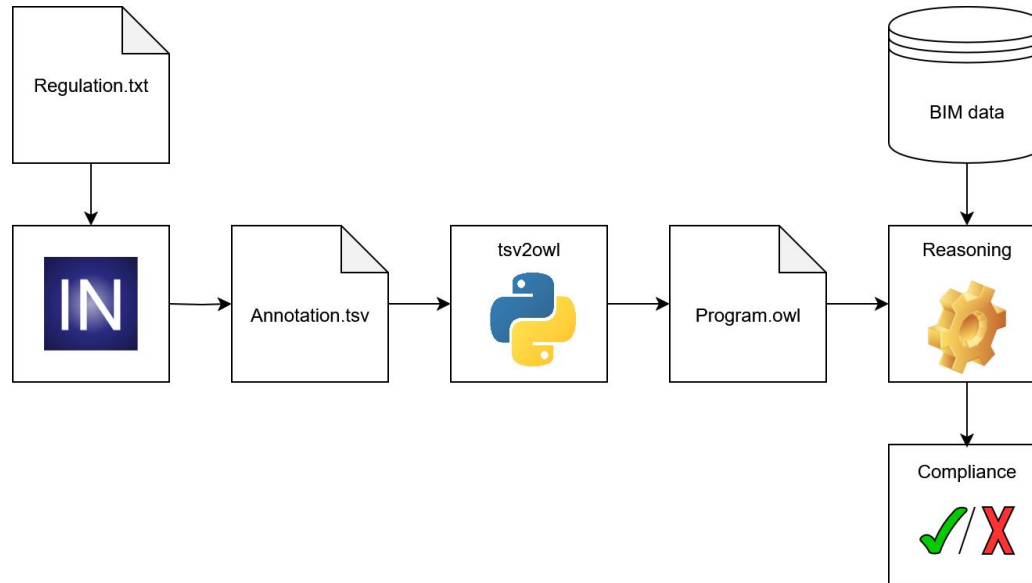
# SHACL vs OWL

	SHACL	OWL
<b>World assumption</b>	Closed	Open*
<b>DL</b>	ALCOIQ(°)	SROIQ
<b>Finite decidability</b>	✓	✓
<b>Infinite decidability</b>	X	✓
<b>Explainability</b>	Validation reports provide limited information	Reasoning explanations offer detailed information
<b>Human understanding</b>	Still RDF code	Natural-language like Manchester syntax

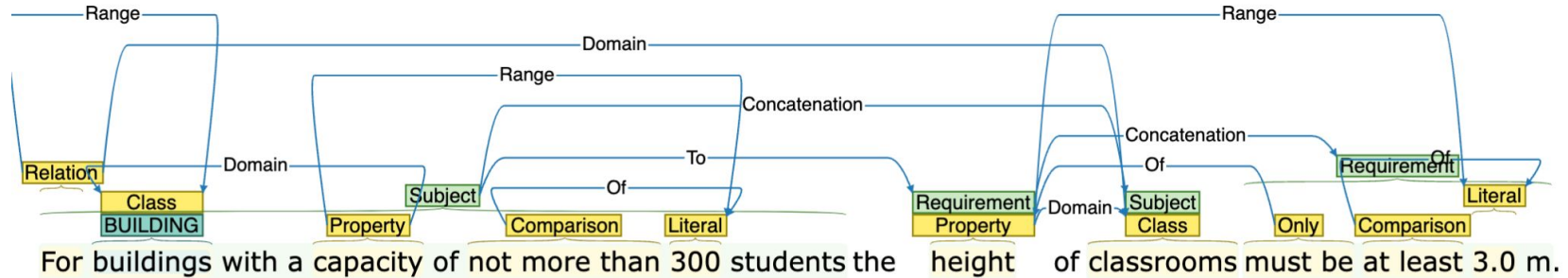
\*Can be closed

## Goal and approach

**Goal:** Develop a symbolic reasoning system for automated compliance checking using annotated text-to-OWL transformation



# Annotation schema



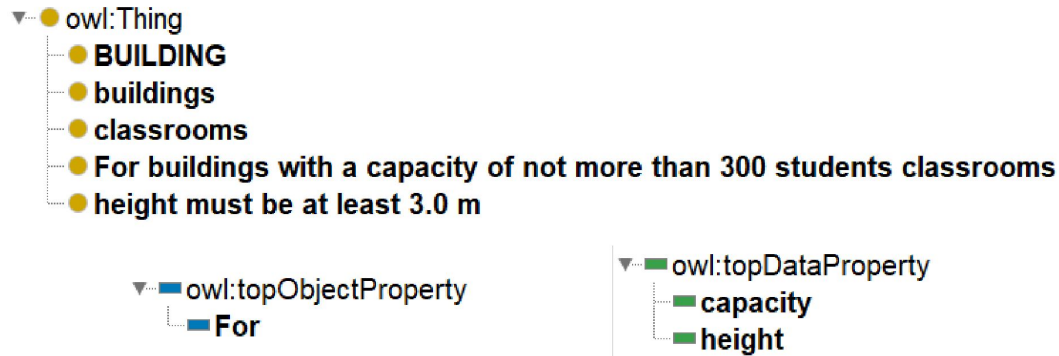
- **Terms layer:** ifcOWL
- **Semantic types layer:** OWL syntax
- **Semantic arrows:** supports construction of OWL restrictions
- **Semantic roles layer:** slots of an OWL axiom
- **Linguistic arrows:** supports construction of tags

$$\text{subject } C_1 \sqsubseteq \exists R.C_2 \text{ restriction}$$

Annotation	OWL
Class	owl:Class
Relation	owl:ObjectProperty
Property	owl>DataProperty
Literal	xsd:string...
Not	owl:complementOf
Or	owl:unionOf
Some	owl:someValuesFrom
Only	owl:allValuesFrom
Number	owl:Cardinality...
Comparison	xsd:minInclusive...

# Translating annotation into OWL code

## 1. Creating elementary entities



## 2. Entity alignment based on textual token inclusion

```
ifc:IfcBuilding owl:equivalentTo :buildings
```

# Translating annotation into OWL code

## 3. Vocabulary mapping

“not more than” => `xsd:maxInclusive`  
“at least” => `xsd:minInclusive`

## 4. Constructing OWL restrictions

- For some  
(buildings  
and (capacity some `xsd:int[<= "300"^^xsd:int]`))
- height only `xsd:float[>= 3.0]`

## 5. Constructing an axiom

Equivalent To 

- classrooms  
and (For some  
(buildings  
and (capacity some `xsd:int[<= "300"^^xsd:int]`)))

SubClass Of 

- 'height must be at least 3.0 m'

## Compliance checking demo

1. Import OWL code representing the regulation  $R$
2. Create individuals complying with  $R$
3. Close individuals
4. Run reasoner => ***No inconsistencies should occur***
5. Modify the individuals so they don't comply with  $R$
6. Run reasoner => ***An inconsistency should be detected***

# Compliance checking demo

## Creating instances

Building of capacity 300

```
[11] building_capacity_300 = onto.Building('building_capacity_300')
      building_capacity_300.capacity = 300
```

Classroom in the building of capacity 300

```
[9] classroom_in_building_capacity_300 = onto.Classroom('classroom_in_building_capacity_300')
     onto_for[classroom_in_building_capacity_300].append(building_capacity_300)
     classroom_in_building_capacity_300.height = 3
```

Closing individuals and saving

```
▶ for indiv in onto.individuals():
  close_world(indiv)

onto.save(f'{path}{req_val_name}', format='rdfxml')
```

<- Enables closed-world-like reasoning

# Compliance checking demo

## Building of capacity 300

✓ Closure  
Classified

Types +	Object property assertions +
<ul style="list-style-type: none"> <li>Building</li> <li><b>for only owl:Nothing</b></li> <li>IfcBuilding</li> </ul>	<ul style="list-style-type: none"> <li>capacity 300</li> </ul>
<ul style="list-style-type: none"> <li>Data property assertions +</li> </ul>	

## Classroom in building of capacity 300 with height 3 m

✓ Closure  
Classified

Types +	Object property assertions +
<ul style="list-style-type: none"> <li>Classroom</li> <li><b>for only ({building_capacity_300})</b></li> <li>'For buildings with a capacity of not more than 300 students classrooms'</li> </ul>	<ul style="list-style-type: none"> <li>for building_capacity_300</li> <li>height 3</li> </ul>
<ul style="list-style-type: none"> <li>Data property assertions +</li> </ul>	

# Compliance checking demo

## Changing height to 2.5 m

Types +

- Classroom ? @ x o
- for only ? @ x o
- ({building\_capacity\_300})

Object property assertions +


- for building\_capacity\_300 ? @ x o

Data property assertions +

- height 2.5 ? @ x o

**X** not compliant

## Inconsistency is detected

 InconsistentOntologyException: Cannot do reasoning with inconsistent ontologies!  
Reason for inconsistency: Literal value "2.5"^^decimal does not belong to datatype decimal[>= "3.0"^^decimal]

OK

Explanation(s):

- 1) classroom\_in\_building\_capacity\_300 height 2.5  
    Subject subclassOf Restriction  
    classroom\_in\_building\_capacity\_300 type Subject  
    Restriction equivalentTo height only decimal[>= 3.0]

## Conclusion

- ✓ Regulations are represented as OWL code, both *human-* and *machine-readable*
- ✓ Formalization is *streamlined* by replacing manual modeling with text annotation
- ✓ Aligning natural language text of regulations with their semantics enables application of *NLP* to predict the annotations, still *controllable* by human
- ✓ OWL reasoning provides *trustworthy* and *explainable* compliance checking
- ✓ Due to infinite decidability, OWL representation enables *redundancy* and *inconsistency* checking of regulations\*

\*not demonstrated yet

## Future work

- Consistency and redundancy checking of building codes
- Facilitating the annotation with NLP, ML and LLM

Ildar Baimuratov, Denis Turygin and Dmitrii Pliukhin

Towards the Automated Annotation of Regulatory Text for OWL-Based Compliance Checking

- Aligning BIM data with regulations in OWL

**Ueli Saluz**

Research Associate  
Leibniz University  
Hannover

semio: Large-Language-Model-based Building-Information-Model Alignment for Automatic-Compliance-Checking – Towards Closing the Gap between Model Authoring and Model Checking for Kit-of-Parts Architecture

Final Paper 

**Thanks for your attention!**

Questions?