Converting Fire Safety Regulations to SHACL Shapes Using Natural Language Processing

Alex Donkers[1,*] and Ekaterina Petrova[1]

[1] *Information Systems in the Built Environment, Eindhoven University of Technology, De Groene Loper 6, Eindhoven, The Netherlands*

Abstract

Automated fire safety compliance checks require a digital representation of regulations; however, digitizing the regulations and capturing semantic concepts from regulatory texts remain challenging. This paper investigates whether the linguistic structure of a sentence can be used to automatically generate Shapes Constraint Language (SHACL) representations from natural language and enable standard-based checking and conformance of Building Information Models with fire safety requirements. We combine Semantic Web technologies and Natural Language Processing techniques to extract linguistic structures from text and automatically map the text to predefined SHACL templates. The results show that we can convert simple regulations and contextual regulations to SHACL shapes automatically. The latter enables performing automated compliance checking on Linked Building Data. Future research aims to improve the method and enable the conversion of more complex sentence structures.

Keywords

fire safety, SHACL, Natural Language Processing, Semantic Web, Ontologies, Linked Building Data, automated compliance checking

1. Introduction

Compliance with fire safety requirements is essential in building design and engineering. As such, fire safety regulations are a core part of building permit regulations. Both research and practice have been addressing fire safety over the past decades, leading to a decline in building-related fire deaths [1]. However, these regulations often come in non-machine-understandable formats, making it hard to perform automated compliance checking on these regulations. This results in fire safety checks being performed rather late in the design process, potentially causing significant delays, cost overruns and health hazards.

Fire safety regulations are typically complex sets of descriptive rules (codes) that often require expert assessment and interpretation. These regulations are highly specific and differ for various locations, building types, functions, and elements. The information necessary to assess the regulations or ensure compliance with them is often not properly digitized or is scattered across different local files by the various stakeholders involved in the design process. Contextualizing the regulations also requires expert knowledge or simulation results, which are difficult to capture in formal representations due to their complexity and the need for a deep understanding of the subject matter. Furthermore, the use of technical terms varies between stakeholders. For instance, the definitions and semantics of core concepts such as 'height' or

* Corresponding author.
NLP4KGC: 3rd International Workshop on Natural Language Processing for Knowledge Graph Creation, September 17, 2024, Amsterdam, Netherlands.
a.j.a.donkers@tue.nl (A. Donkers); e.petrova@tue.nl (E. Petrova)
0000-0002-8809-3277 (A. Donkers); 0000-0002-8651-0671 (E. Petrova)

‘area’ are often poorly defined and differ across regulations and countries, leading to ambiguities [1,2].

Semantic Web technologies have shown significant potential in overcoming some of these challenges [3]. Earlier work in this field led to the integration of heterogeneous data into software-agnostic linked data. The development of ontologies led to the explicit definition of concepts in the Architecture, Engineering, and Construction (AEC) domain, some of which focus specifically on fire safety [4,5]. The Semantic Web stack also introduced methods to perform compliance checking. Nuyts et al. [6] compared those methods (‘linked data approaches’) with methods relying on Industry Foundation Classes (IFC) and methods using JSON or XSD. The authors concluded that the linked data approaches utilizing Semantic Web Rule Language (SWRL), SPARQL Protocol and RDF Query Language (SPARQL), and Shapes Constraint Language (SHACL) are the most expressive and thus best suited for automated compliance checking in the AEC industry. Nuyts et al. [7] later manually created SHACL shapes to perform automated compliance checking. Manual SHACL shape generation is, however, a time-intensive task.

This paper proposes a method to convert fire safety regulations written in natural language to the machine-interpretable SHACL format and enrich those digital rules with explicitly defined fire safety knowledge in ontologies. The hypothesis is that the linguistic structure of the sentence, extracted using a combination of NLP techniques (part-of-speech tagging, named entity recognition, dependency parsing, and noun chunk creation), could be used to generate SHACL shapes based on predefined SHACL templates.

The remainder of the paper is organized as follows. Section 2 reviews relevant existing work. The methods adopted and developed in this work are described in section 3. Section 4 presents the results, followed by a discussion and conclusion in sections 5 and 6, respectively.

2. State-of-the-Art

2.1. Fire Safety Regulations and their Varying Complexities

Fire safety regulations come in varying complexities. A review of the Dutch, Danish, and Belgian fire safety regulations shows that these regulations differ based on i) their type (e.g., related to evacuation, structural safety of the building, smoke propagation, and so forth), ii) rule complexity (some are simple property value checks, some are dependent on relations with other objects or zones, some require geometric computations, some require simulations, and some need expert assessments), iii) information complexity (i.e., some regulations can be checked based on IFC models while others require more information), iv) ambiguity, v) completeness of the rule (e.g., in cases when the regulatory text refers to other documents), vi) explicitness (e.g., some regulations explicitly state a value that a building needs to comply with, some don’t), vii) the complexity of the sentence (based on the linguistic structure) and viii) the level of automation that can be reached in the checking procedure, as referred to in [8]).

2.2. Natural Language Processing and Fire Safety Regulations

NLP techniques help computationally analyze and understand speech and written text. These techniques have been previously applied to process regulations written in natural language. For example, Zhang and El-Gohary [9] developed a method to extract information from regulations using NLP and applied ontologies to help map words to semantic concepts. They later linked regulations to IFC models via patterns created using POS tags[10]. Li et al. [11] applied NLP algorithms to regulations and successfully extracted spatial terms from regulations to perform spatial reasoning (using gazetteer lists). Zheng et al. [12] transformed fire safety regulations into SPARQL queries by performing semantic labelling and parsing with BERT. They linked those parsed words to classes in an ontology using keyword matching. Chiappini et al. [13] used BERT and ChatGPT to convert building regulations to SPARQL queries.

2.3. Semantic Web Technologies and Fire Safety Regulations

The recent developments of Semantic Web technologies in the construction industry [3] have significantly impacted multiple other subdomains, including fire safety engineering. Fitkau et al. [4] created the FiSa ontology to describe preventive fire safety domain knowledge, after which this ontology was used for compliance checking using SWRL rules. The work is specifically designed for German regulations. Other researchers developed ontologies that are not specifically designed for fire safety regulations but incorporate knowledge related to fire safety. Guyo et al. [5] created an ontology that describes the information that firefighters need during a building fire emergency. Other contributions focus on operational fire protection, such as the building fire protection ontology [14], the Ofepac ontology [15], and the work by Li et al. [16]. Other ontologies focus on emergency situations, typically to help evacuation [17,18].

Research related to creating or applying SHACL shapes to evaluate fire safety is in its infancy [19]. However, multiple studies apply Semantic Web technologies to rule checking, typically for building performance or building permit reviews. Zentgraf et al. [20]created the OntoBPR ontology, which was used to generate SHACL shapes for building permit reviews using SPARQL queries. Fauth and Seiß [21] developed the ontology for building permit authorities to cover the process of building permit checking by authorities.

3. Methodology

3.1. The FireBIM ontology stack

To semantically enrich the digital representation of fire safety regulations, an ontology is designed to specifically serve the enrichment of fire safety regulations. The current contribution should not necessarily be construed as one ontology, but rather as a set of ontologies that collectively reach the goal of semantic enrichment. The set of ontologies has the primary goal of semantically describing fire safety regulations and building information across European borders, such that humans and machines can combine and understand those two domains.

To achieve the above, the FireBIM ontology stack consists of three main modules. The first module contains ontologies to structure regulations and add metadata to the regulations. This module is necessary to navigate through and between regulatory documents and understand the structure of regulatory documents in other countries or jurisdictions. This module is domain-agnostic; it merely aims to capture the structure of regulatory documents and could be used for multiple types of regulations, not exclusively fire safety regulations.

The second module contains ontologies to structure the content of rules and capture the semantics of regulations themselves. Those regulations contain deep contextual knowledge of a specific domain. As the focus of this work is on fire safety regulations, the ontology should therefore cover expert knowledge on fire safety. By semantically capturing the most important concepts in fire safety regulations and making them explicit, this second module allows comparing and reusing regulations across European borders. Secondly, the ontology aims to bridge concepts from the regulatory documents to building information models so that machines can understand both the regulations and the building information, enabling automated compliance checking in a later stage. This second module contains concepts from the Building Information Modelling (BIM) domain and is meant to be used in both the phases prior to and after construction.

The third module contains knowledge of the operational phase of a building. Although this module is out of scope for this work, it contains valuable information structures to apply the digital rulesets in later phases of the building's lifecycle (e.g., to perform as-built performance assessments).

3.1.1. Structuring Regulations

The regulation ontology maps one-to-one to parts of the AEC3PO ontology [22]; however, it is more lightweight, following best practices from the W3C Linked Building Data Community Group. Figure 1 shows the current version of the ontology. It consists of three main classes: the firebim:Authority (which represents the legal body that publishes and maintains the regulatory document), the firebim:DocumentSubdivision (which represents documents or parts of documents), and the firebim:Reference (which represents references to other representations of the regulation, or similar regulations). The firebim:DocumentSubdivision class has a subclass tree that defines a document, a section, an article, and a member. The latter typically holds the one or multiple bodies of text that an article exists of. We introduce multiple types of sections, such as chapters, subchapters, paragraphs, appendices, tables, and figures.

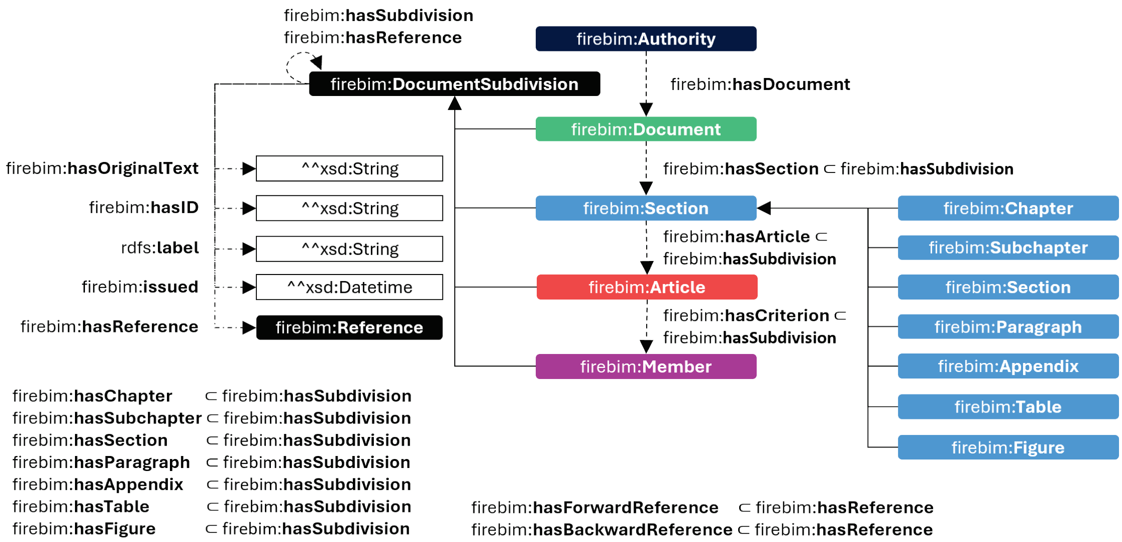

Figure 1: FireBIM regulation ontology

Figure 2 shows how the ontology can be instantiated for multiple Dutch fire safety articles. It clearly shows a document modeled as a tree structure with multiple members per article and multiple articles per paragraph. Members can have references to other members, using the firebim:hasBackwardReference and firebim:hasForwardReference object properties. This enables members to refer to other members if they, for example, contain constraints for the other member, as shown in Figure 2. This first part of the FireBIM ontology stack does not semantically enrich the regulation or the building itself; the regulatory member text is simply added to the graph as a literal.

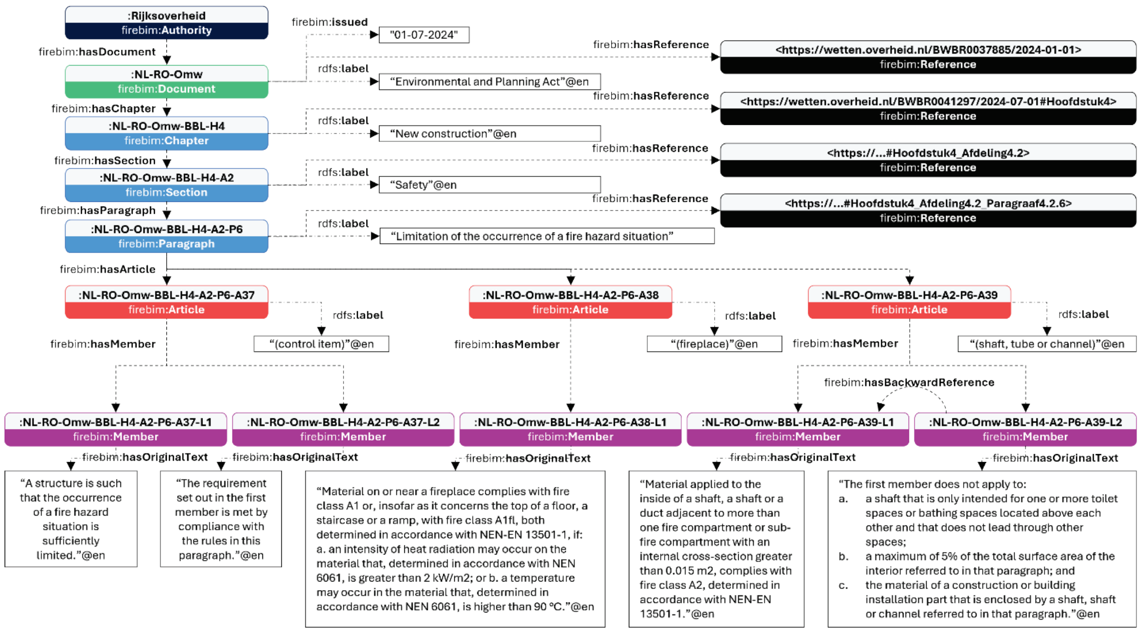

Figure 2: Instantiation of the FireBIM regulation ontology

3.1.2. Naming Convention

Different regulatory documents use different naming conventions, which could lead to difficulties when comparing or combining multiple regulations. The structure of those documents, however, often shows similarities. Earlier research shows that differences in those naming conventions are not problematic as long as parts of the name can be tagged with classes [23]. Instantiation of the FireBIM regulation ontology (using Python scripts) automatically leads to a document-specific naming convention for regulations. Figure 3 shows how the instantiation of the Dutch Environmental and Planning Act led to a naming convention for Dutch building regulations. The classes related to each part of the legal document are used to tag each part of the name so that this name gets a semantic structure. This enables the creation of different naming conventions in different countries (or different legal documents) while keeping the same semantics behind the name.

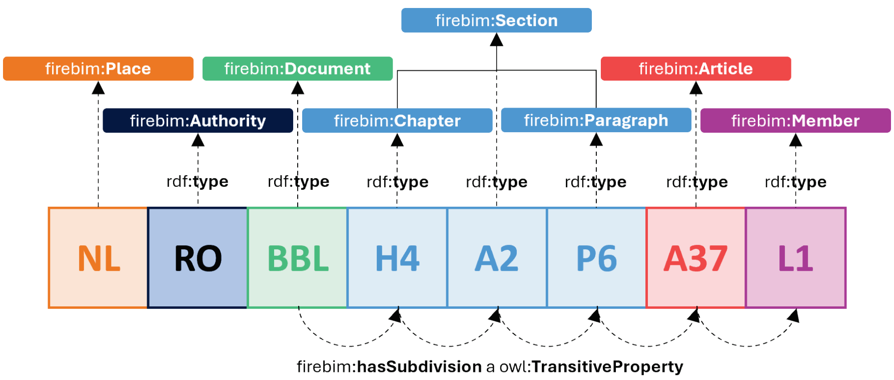

Figure 3: Naming convention of regulations based on the FireBIM ontology

3.1.3. Semantically Enriching the Regulations

The second module in the FireBIM ontology stack contains ontologies to semantically capture building-related concepts. These concepts can also be used to tag important concepts inside the member text of the regulations themselves. As the state-of-the-art section has shown, a significant body of knowledge on ontology development in the AEC industry already exists, and best practices suggest reusing those ontologies. Since a global standard for Linked Building Data does not exist yet, we stick to ontologies commonly used in the W3C Linked Building Data Community Group, being BOT [24], BPO [25], GeoSPARQL, OMG and FOG [26,27], OPM [28], NEO [29], and a simple OWL representation of IndoorGML. Figure 4 shows how the reused ontologies in the FireBIM stack align. No formal alignment modules were created; alignment is done by reusing some object properties (in the case of OPM, OMG, and NEO) or via a loose Abox connection (by simply using multiple rdf:type relationships on an instance).

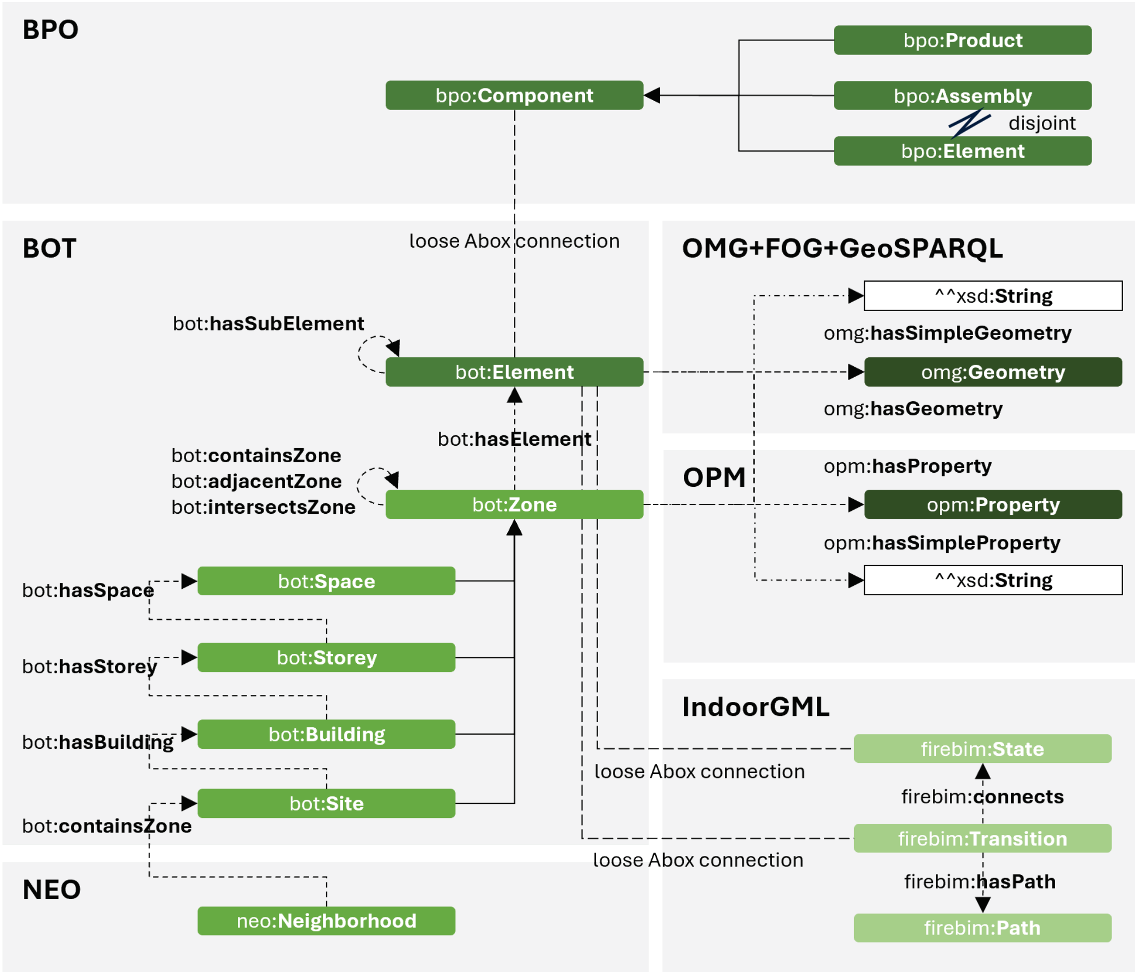

Figure 4: FireBIM building ontology

This top-level building ontology can be used as a generic framework to represent any kind of building information. The FireBIM ontology stack presents so-called taxonomy modules that extend classes of the top-level structure to enable more specific semantic descriptions of things. These taxonomy modules are essentially large subclass trees of a single class in the top-level ontology stack, including a subclass tree for bot:Element, opm:Property, and bot:Zone, representing different elements, properties, and zone types that occur in fire safety regulations. Using these taxonomies, the FireBIM ontology stack enables the description of theoretically every possible component and its properties (as shown in Listing 1 and 2). The taxonomies are constructed in collaboration with fire safety experts and allow extensions, while the top-level ontology is relatively stable.

```
Without taxonomies
bot:Space bot:containsZone bot:Space .
With taxonomies
firebim:FireCompartment bot:containsZone firebim:OccupiedSpace .
```

Listing 1: Zone-to-zone relationships using the FireBIM taxonomies

```
Without taxonomies
bot:Element opm:hasProperty opm:Property schema:hasValue "250" .
With taxonomies
firebim:RoadTunnelTube opm:hasProperty firebim:Length  schema:hasValue "250" .
```

Listing 2: Elements and their properties using the FireBIM taxonomies

3.2. Regulation Converter Pipeline

To convert (fire safety) regulations to a machine-understandable format, a regulation converter pipeline was created (Figure 5). The pipeline consists of three main parts. The first part – the regulation converter – aims to convert regulations into SHACL shapes. The second part of the pipeline – the structure parser – parses the regulatory document into a graph following the FireBIM regulation ontology and links this graph to the SHACL shape. The third part is a library with SHACL templates and ontologies. These templates are used to convert regulations to SHACL shapes, and the ontologies are used to semantically enrich the results of the generated RDF data. The following subsections describe the individual elements of the pipeline in more detail.

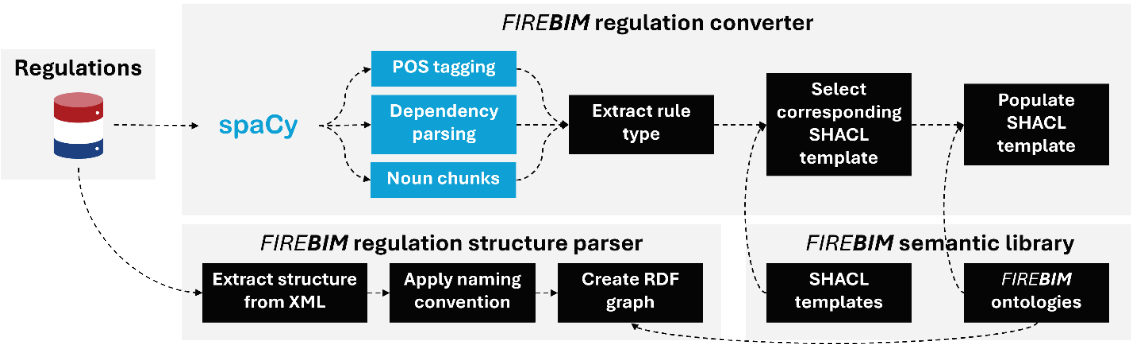

Figure 5: The pipeline to generate SHACL shapes from (Dutch fire safety) regulations

3.2.1. Regulation Structure Parser

To parse regulations into a structure similar to the instance data in Figure 2, Dutch fire safety regulations were first extracted from the Dutch online portal for regulations. An XML structure used to be available; however, the XML did not contain metadata to automatically extract regulations related to fire safety. Therefore, 374 fire safety regulations were manually selected from multiple regulatory documents and stored in a CSV file. Each column in the CSV file represented the next step in the document structure. Based on those columns, the developed naming convention (as shown in section 3.1.2) could be generated. The column headers were used to map items in the CSV to concepts in the ontology. The Dutch member text of the regulations is translated to English using Google Translate, and both the Dutch and English versions of the member text are stored in the graph.

3.2.2. Regulation Converter

The regulation converter is designed to convert regulations defined in natural language to regulations in SHACL. The initial assumption of this process is that the linguistic structure of the sentence can be used to:

1. extract the type of regulation, and, therefore, match this regulation with a corresponding SHACL template, and
2. extract the important concepts from the sentence to populate the SHACL template.

The regulation converter applies the spaCy [30] Python library to capture this linguistic structure of the sentence. spaCy has various linguistic features that can capture linguistic knowledge from sentences, such as part-of-speech (POS) tagging (which tags words with categories from the universal part-of-speech tag set [31]), dependency parsing (tagging the syntactic dependency relationship between a word and its parent), the creation of noun chunks (chunks of words that describe a noun), dependency tree creation (connecting child words to their parents or heads), named-entity recognition (tagging words or groups of words with a label that gives a semantic meaning), and vector-based similarity comparisons of words. Depending on the chosen pipeline, the language of this pipeline, and how this pipeline is trained, the results of spaCy's linguistic features will be more efficient or accurate.

This paper mainly uses three spaCy features: POS tagging, dependency parsing, and noun chunk creation. A stepwise approach is followed to extract the rule type and populate a SHACL shape. Figure 6 visually explains each step of this approach using a real regulation.

1. Based on the dependency parsing, the root word of the sentence is extracted (*'located').*
2. The group of words that the root is directly related to, such as auxiliary clauses ('AUX') or adpositions ('ADP') are extracted and grouped with the root (*'is located in'*). This group of words is used to determine the rule type based on a mapping table that was created for this project.
3. The nominal subject (or equivalent, such as the passive nominal subject) that is the child of the root is extracted (*'area'*).
4. The noun chunk of the nominal subject is created (*'An occupied area'*), and in this noun chunk, it is checked whether there are any special determiners (*'An'*), such as quantifiers.
5. All child relations of the nominal subject that modify the nominal subject, such as verbs, adjectival modifiers, or noun compounds are extracted (*'occupied'*).
6. These child relationships are combined with the nominal subject (*'occupied area'*).
7. Based on the results of step 2, step 7 and onwards changes. For this regulation, the other nouns that are children of the root are extracted (*'compartment'*).
8. Steps 4 and 5 are repeated, but now for this new noun (*'protected, sub, -, fire).*
9. The child relationships are combined with the noun (*'protected sub-fire compartment'*).

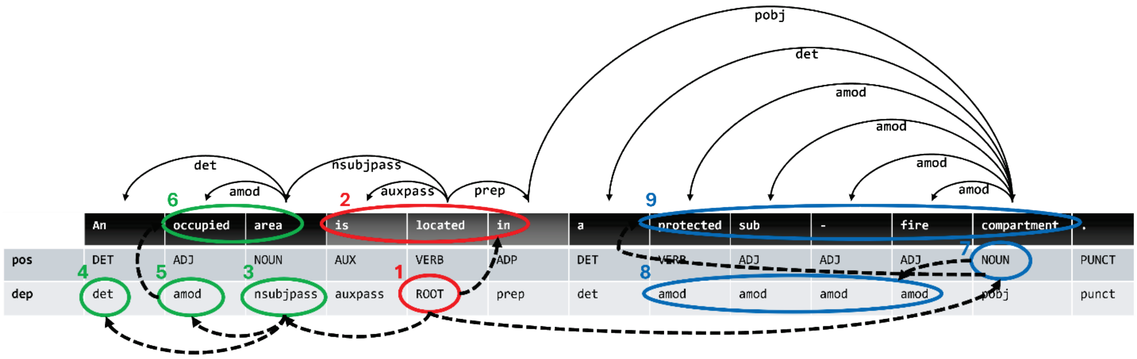

Figure 6: Example of the stepwise approach to extract RDF-based patterns out of a sentence based on its linguistic structure

3.2.3. SHACL Shape Creation

The extraction of the linguistic structure is then used to convert the sentence to a machine-understandable format. Based on the creation of dependency trees and noun chunks of various fire safety regulations (see example in Figure 7), it is concluded that regulations follow a triple-like structure with a group of words that are related to the subject of the sentence (the left side), a group of words that are related to the root of the sentence (typically also containing the finite verb, the middle), and a group of words that are related to the object of the sentence (the right side). It may not be a coincidence that these sentences follow a structure similar to RDF triples, consisting of a subject, predicate, and object. Internally, the structure of the left, middle, and right parts of the sentence shows variations, ranging from very simple structures (as shown in the example in Figure 6), to more complex structures with exceptions, constraints, conjunctions, numerical modifiers, and so forth.

SHACL shapes, also following the RDF model, are created using a similar pattern. Shapes consist of a target, a property, and a constraint, to which we could map the left side, the middle, and the right side of the sentence. Similarly to the variations in the internal structure of those three parts of the sentence, SHACL enables the creation of simpler or more complex targets, properties, and constraints. It is possible to stack parts of SHACL shapes by adding multiple properties and constraints to a single target.

This paper introduces a library of basic, manually created SHACL template shapes for various types of regulations, including:

1. Regulations where a target needs to have a property with a specific object class.
2. Regulations where a target needs to have a minimum or maximum number of a specific outgoing property.
3. Regulations where a target needs to have a property with a specific datatype.
4. Regulations where a target needs to have a property with a specific value.
5. Regulations where a target needs to have a property with a value in a specific range.

After selecting the correct SHACL template, these templates could be filled by inserting the right variables based on the results of section 3.2.2.

3.2.4. Semantic Enrichment

The variables inserted in the SHACL shapes should be first converted to concepts that also exist in the FireBIM building ontology. Adding those semantics to the SHACL shape allows for checking the RDF-based BIM model later. We see two options to convert concepts from the natural language regulations to concepts from the ontology. In option one, words from the regulation are selected based on their linguistic features and directly prefixed, making them classes or properties directly. Then, a mapping service (such as the buildingSmart Data Dictionary (bSDD) or a similar local variant) should be used to manually map the newly defined classes to the FireBIM building ontology. A second option is to extract the semantics of words in the regulation during the NLP pipeline and find the most similar concept in the ontology, mapping the word to the correct concept directly. The latter option would be the most sophisticated one; however, this option requires training a large language model (LLM) on fire safety concepts. As such a model is not available at this stage, this project applies the first approach. Based on the type of regulation, words are combined and converted to classes, object properties, or datatype properties. Next to this, a subclass structure is automatically generated based on the occurrence of verbs, adjectival modifiers, or noun compounds in front of a noun. Finally, all classes are prefixed. The regulation in Figure 6 would get the following output, as described in Listing 3:

```
firebim:isLocatedIn .
firebim:OccupiedArea rdfs:subClassOf firebim:Area .
firebim:ProtectedSub-FireCompartment rdfs:subClassOf firebim:Sub-
FireCompartment .
firebim:Sub-FireCompartment rdfs:subClassOf firebim:FireCompartment .
firebim:FireCompartment rdfs:subClassOf firebim:Compartment .
```

Listing 3: Automatic creation of properties, classes and their subclasses in the NLP pipeline

4. Results

The developed approach was tested on multiple regulations. Structuring the regulations using the structure parser is not dependent on the complexity of the regulation itself, as the plain text is stored in a literal in the RDF representation. Listing 4 shows a snippet of the resulting graph with instances of the regulatory document, a chapter, and an article in this chapter.

```
:NLROBBL
      rdf:type            firebim:Document ;
      rdfs:label          "Buildings and Living Environment Decree"@en ;
      firebim:issued      "01-07-2024" ;
      firebim:reference   <https://wetten.overheid.nl/BWBR0041297/2024-01-01>
;
      firebim:hasChapter  :NLROBBLH4 .

:NLROBBLH4
      rdf:type            firebim:Chapter ;
      rdfs:label          "Chapter 4: New construction"@en ;
      firebim:reference   <https://wetten.overheid.nl/BWBR0041297/2024-01-
                          01#Hoofdstuk4_Afdeling4.2> ;
      firebim:hasSection  :NLROBBLH4A2 .

...

:NLROBBLH4A2P9A58
      rdf:type            firebim:Article ;
      rdfs:label          "Article 4.58 (protected sub-fire compartment:
                          location)"@en;
      firebim:hasMember   :NLROBBLH4A2P9A58L1 .
```

Listing 4*:* Resulting RDF graph of member text NLROBBLH4A2P9A58L1

The conversion of the regulations into SHACL shapes remains a more complex process. This work tested the regulation converter for multiple regulations, and at this stage, it can generate SHACL shapes for regulations with lower complexity (as defined in section 2.1). The regulations that were successfully converted were simple rules (i.e., whether a subject has a property of a certain kind or with a certain value) or relational rules (i.e., simple relationships between a subject and an object, often based on their location). The resulting SHACL shapes were tested on small RDF datasets using pySHACL.

Figure 7 shows the creation of the dependency tree and noun chunks of a relational regulation. It clearly consists of a left, middle, and right side of the sentence. Based on the chunk of text around the root ('is located in'), a simple SHACL template could be selected and populated. The result is shown in Listing 5. The listing links to the graph in Listing 4, enabling querying this SHACL shape using the document structure of this legal document.

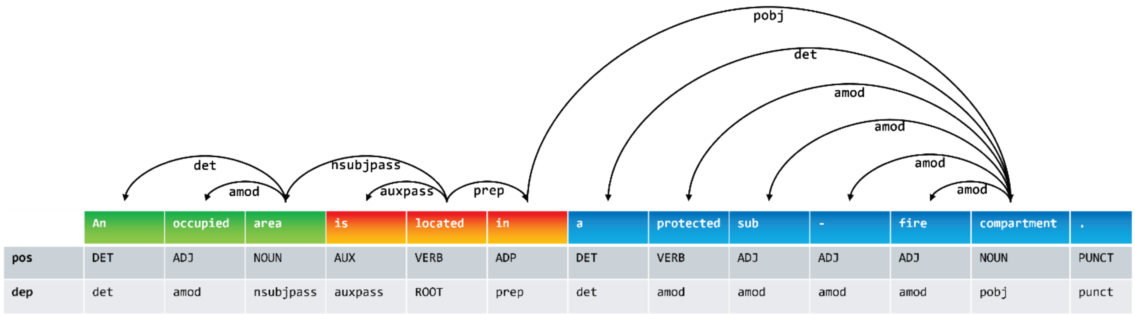

Figure 7: Dependency tree and noun chunks of a fire safety regulation

```
:NLROBBLH4A2P9A58L1
       a                    sh:NodeShape , firebim:Member ;
       rdfs:comment         "Relational rule" ;
       sh:targetClass       firebim:OccupiedArea ;
       sh:property [
           sh:path          firebim:isLocatedIn ;
           sh:class         firebim:ProtectedSub-FireCompartment ;
       ];
       firebim:hasOriginalText     "Een verblijfsgebied ligt in een beschermd
                                   subbrandcompartiment."@nl,
                                   "An occupied area is located in a protected
                                   sub-fire compartment."@en ;
       firebim:hasID               "NLROBBLH4A2P9A58L1" ;
       firebim:hasReference
       <https://wetten.overheid.nl/jci1.3:c:BWBR0041297&hoofdstuk=4&afdeling=
4.2&paragraaf=4.2.9&artikel=4.58&z=2024-01-01&g=2024-01-01> ;
       rdfs:hasLabel               "Article 4.58 (protected sub-fire
                                   compartment: location) Member 1"@en .
```

Listing 5: Resulting SHACL shape of member text NLROBBLH4A2P9A58L1

Figure 8 shows the dependency table and noun chunks of a second regulation. This regulation still clearly has a left ('A road tunnel tube with a length of more than 250 m'), a middle ('is located in'), and a right side ('a fire compartment'); however, the left side is clearly more complex. This results in selecting a different target method in SHACL, nesting a SPARQL query in the target class. Listing 6 shows the resulting SHACL shape.

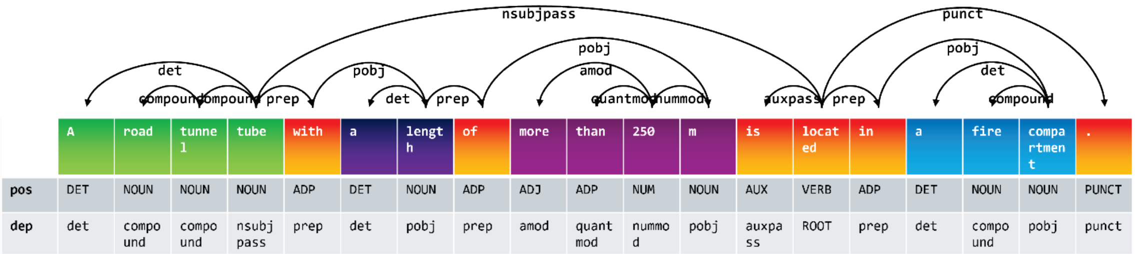

Figure 8: Dependency tree and noun chunks with a more complex object

```
: NLROBBLH4A2P8A50L3
        a             sh:NodeShape , firebim:Member ;
        rdfs:comment  "A road tunnel tube with a length of more than 250 m is
                      located in a fire compartment."@en ;
        sh:target     [
                      a             sh:SPARQLTarget;
                      sh:prefixes   firebim: , bop: , quantitykind: , bot: ;

                      sh:select     """" SELECT ?target
                                    WHERE {
                                            ?target a firebim:RoadTunnelTube ;
                                            firebim:length ?value ;
                                            FILTER (?value > 250) . """" ;]
        sh:property [
                      sh:path       firebim:isLocatedIn ;
                      sh:class      firebim:FireCompartment ; ]
```

Listing 6: Resulting SHACL shape of member with a slightly more complex target

5. Discussion

This work presents an automated pipeline to convert fire safety regulations to SHACL shapes using NLP. It improves existing work that was able to extract knowledge from regulations using NLP, but did not generate SHACL shapes [9], and work that did generate SHACL shapes, but only manually [7]. The work also presents a general-purpose ontology stack for building regulations that was applied to fire safety regulations but could also be applied to other types of regulations. This extends existing work on ontologies that are either very generic and do not capture detailed concepts from the building information [22] or very specific and only capture concepts from a specific use case or country [4]. However, the present work has some limitations, which should be addressed in future research.

First, this work uses the en_core_web_sm model from spaCy, which is a small, pre-trained English dataset. Although the POS tagging and the dependency tree creation performed well for most of the simple regulations that were tested, the model had difficulties with some ambiguous statements in regulations. An example is the noun chunk 'portable fire extinguisher', where 'portable' should be an adjectival modifier to 'extinguisher' (it's the extinguisher that is portable); however, the implemented model interpreted 'portable' as a modifier to 'fire' (while the fire is obviously not the portable concept in this noun chunk). Future work should take into account this ambiguity as it does affect the creation of the SHACL shapes and, thus, the correctness of the digital rule [2]. Future work should also explore the use of more novel LLMs in the SHACL generation pipeline.

This work assumes that semantic mapping to a core ontology will be performed after the SHACL shape generation, via a bSDD-like system. Another approach would be to map to correct classes directly in the NLP pipeline, for example by performing named-entity recognition and similarity matching to directly match raw text to semantic concepts. It is essential to correctly map to those semantic concepts to capture the meaning and context of words in regulations [1].

The SHACL shapes in this work have been validated against some manually created Linked Building Data graphs. However, to scale up this approach, the creation of such graphs should rely less on manual actions, and therefore converters need to be able to map the semantic concepts in the ontology to data generated by designers.

6. Conclusion

Performing automated compliance checking on fire safety regulations requires a digital representation of the regulations and a thorough understanding of the semantics of those regulations by both humans and machines. Although much research has investigated automated compliance checking of (fire safety) regulations, the goal of capturing the regulations in a machine-understandable format has yet to be fully achieved. This study utilizes Semantic Web technologies and NLP to digitize fire safety regulations into machine-understandable rules. The paper first presents two ontologies in the FireBIM ontology stack – the FireBIM regulation ontology and the FireBIM building ontology. These ontologies aim to capture the semantics of both regulatory documents and the buildings they are describing. The paper then presents a pipeline to convert natural text from regulations, by using the linguistic structure of the sentence. A spaCy-based script captures part-of-speech tags, noun chunks, and dependency trees, after which a rule-base system populates SHACL templates. The regulations are then transformed into RDF using a custom-built converter and linked to the accompanying SHACL shape. The result is an RDF graph capturing the regulatory document, its internal structure, and its regulations in a machine-understandable format. In the future the SHACL shapes could be used to perform automated compliance checking on Linked Building Data.

The pipeline in this paper is used to convert regulations that cover simple property checks or relations between objects, typically with a low sentence complexity. Future research should determine whether the pipeline can also convert more complex regulations, whether different linguistic structures of sentences require different step-by-step NLP approaches, and what functionalities of SHACL could be used to convert those regulations to a machine-understandable format. Furthermore, we plan to investigate how to handle regulations that require input data that is difficult to store in an RDF format, such as geometric information, simulations, or expert judgment. One example is to combine Semantic Web technologies (such as SHACL) with open-source software from the AEC domain (such as IfcOpenShell).

References

[1] G. Spinardi, L. Bisby, J. Torero, A Review of Sociological Issues in Fire Safety Regulation, Fire Technology. 53 (2017). https://doi.org/10.1007/s10694-016-0615-1.

[2] Z. Zhang, N. Nisbet, L. Ma, T. Broyd, Capabilities of rule representations for automated compliance checking in healthcare buildings, Automation in Construction. 146 (2023). https://doi.org/10.1016/j.autcon.2022.104688.

[3] P. Pauwels, S. Zhang, Y.C. Lee, Semantic web technologies in AEC industry: A literature overview, Automation in Construction. 73 (2017) 145–165. https://doi.org/10.1016/j.autcon.2016.10.003.

[4] I. Fitkau, T. Hartmann, An ontology-based approach of automatic compliance checking for structural fire safety requirements, Advanced Engineering Informatics. 59 (2024). https://doi.org/10.1016/j.aei.2023.102314.

[5] E.D. Guyo, T. Hartmann, S. Snyders, An ontology to represent firefighters data requirements during building fire emergencies, Advanced Engineering Informatics. 56 (2023). https://doi.org/10.1016/j.aei.2023.101992.

[6] E. Nuyts, M. Bonduel, R. Verstraeten, Comparative analysis of approaches for automated compliance checking of construction data, Advanced Engineering Informatics. 60 (2024). https://doi.org/10.1016/j.aei.2024.102443.

[7] E. Nuyts, J. Werbrouck, R. Verstraeten, L. Deprez, Validation of building models against legislation using SHACL, in: CEUR Workshop Proceedings, 2023.

[8] J. Yeung, T. Beach, R. Lavikka, M. Kiviniemi, C.-R. Raitviir, G. Costa, E. Pla, M. Morilla, P. Patlakas, F. Cheung, ACCORD D1.1 Landscape Review Report, 2023.

[9] J. Zhang, N.M. El-Gohary, Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking, Journal of Computing in Civil Engineering. 30 (2016). https://doi.org/10.1061/(asce)cp.1943-5487.0000346.

[10] J. Zhang, N.M. El-Gohary, Extending Building Information Models Semiautomatically Using Semantic Natural Language Processing Techniques, Journal of Computing in Civil Engineering. 30 (2016). https://doi.org/10.1061/(asce)cp.1943-5487.0000536.

[11] S. Li, H. Cai, V.R. Kamat, Integrating Natural Language Processing and Spatial Reasoning for Utility Compliance Checking, Journal of Construction Engineering and Management. 142 (2016). https://doi.org/10.1061/(asce)co.1943-7862.0001199.

[12] Z. Zheng, Y.C. Zhou, X.Z. Lu, J.R. Lin, Knowledge-informed semantic alignment and rule interpretation for automated compliance checking, Automation in Construction. 142 (2022). https://doi.org/10.1016/j.autcon.2022.104524.

[13] F. Chiappini, D. Napps, S. Mastrolembo Ventura, M. König, A. Ciribini, Automatic verification of requirements in BIM models for building permit, in: Digital Building Permit Conference 2024 - Book of Abstracts, Barcelona, 2024: pp. 134–139.

[14] L. Jiang, J. Shi, C. Wang, Z. Pan, Intelligent control of building fire protection system using digital twins and semantic web technologies, Automation in Construction. 147 (2023). https://doi.org/10.1016/j.autcon.2022.104728.

[15] C. López, J. Botia, A system of cooperation based on ubiquitous environments for protection against fires in buildings, in: Advances in Soft Computing, 2009. https://doi.org/10.1007/978-3-540-85867-6_38.

[16] Y. Li, J. Yi, X. Zhu, Z. Wang, F. Xu, Developing A Fire Monitoring and Control System Based on IoT, in: 2016. https://doi.org/10.2991/aiie-16.2016.40.

[17] M. Da, T. Zhong, J. Huang, Knowledge Graph Construction to Facilitate Indoor Fire Emergency Evacuation, ISPRS International Journal of Geo-Information. 12 (2023). https://doi.org/10.3390/ijgi12100403.

[18] J. Neto, A.J. Morais, R. Gonçalves, A.L. Coelho, Context-Based Multi-Agent Recommender System, Supported on IoT, for Guiding the Occupants of a Building in Case of a Fire, Electronics (Switzerland). 11 (2022). https://doi.org/10.3390/electronics11213466.

[19] T. Acke, Research on the applicability of language models for automated building code compliance checks, Ghent University, 2024.
[20] S. Zentgraf, J. Fauth, P. Hagedorn, S. Seiss, K. Smarsly, M. König, J. Melzner, OntoBPR : Ontology-based workflow and concept for building permit reviews, in: 30th EG-ICE: International Conference on Intelligent Computing in Engineering, 2023.
[21] J. Fauth, S. Seiß, Ontology for building permit authorities (OBPA) for advanced building permit processes, Advanced Engineering Informatics. 58 (2023). https://doi.org/10.1016/j.aei.2023.102216.
[22] A. Dridi, P. Patlakas, M. Lefrançois, T. Beach, H. Petr, E. Vakaj, AEC3PO: A Knowledge Model for Machine Executable Construction Regulations, Preprint. (2024).
[23] L. Chamari, J. Van Der Weijden, L. Boonstra, S. Hoekstra, E. Petrova, P. Pauwels, Metadata Schema Generation for Data-driven Smart Buildings, in: W. Terkaj, M. Poveda-Villalón, P. Pauwels (Eds.), Proceedings of the 11th Linked Data in Architecture and Construction Workshop, CEUR-WS.org, Matera, Italy, 2023: pp. 139–150.
[24] M.H. Rasmussen, M. Lefrançois, G.F. Schneider, P. Pauwels, BOT: The building topology ontology of the W3C linked building data group, Semantic Web. 12 (2020) 143–161. https://doi.org/10.3233/sw-200385.
[25] A. Wagner, U. Rüppel, BPO: The building product ontology for assembled products, in: M. Poveda-Villalón, P. Pauwels, R. De Klerk, A. Roxin (Eds.), Proceedings of the 7th Linked Data in Architecture and Construction Workshop, CEUR-WS.org, Lisbon, 2019: pp. 106–119.
[26] M. Bassier, M. Bonduel, J. Derdaele, M. Vergauwen, Processing existing building geometry for reuse as Linked Data, Automation in Construction. 115 (2020). https://doi.org/10.1016/j.autcon.2020.103180.
[27] N.J. Car, T. Homburg, GeoSPARQL 1.1: Motivations, Details and Applications of the Decadal Update to the Most Important Geospatial LOD Standard, ISPRS International Journal of Geo-Information. 11 (2022). https://doi.org/10.3390/ijgi11020117.
[28] M.H. Rasmussen, M. Lefrançois, M. Bonduel, C.A. Hviid, J. Karlshø, OPM: An ontology for describing properties that evolve over time, in: M. Poveda-Villalón, P. Pauwels, A. Roxin (Eds.), Proceedings of the 6th Linked Data in Architecture and Construction Workshop (LDAC 2018), CEUR-WS.org, London, 2018: pp. 24–33.
[29] S. De Meij, A. Donkers, D. Yang, M. Klepper, Making Urban Energy Use More Intelligible Using Semantic Digital Twins, in: W. Terkaj, M. Poveda-Villalón, P. Pauwels (Eds.), Proceedings of the 11th Linked Data in Architecture and Construction Workshop., CEUR-WS.org, Matera, Italy, 2023: pp. 110–122. https://linkedbuildingdata.net/ldac2023/files/papers/papers/LDAC2023_paper_6625.pdf.
[30] M. Honnibal, I. Montani, S. Van Landeghem, A. Boyd, spaCy: Industrial-strength Natural Language Processing in Python, (2020). https://doi.org/10.5281/zenodo.1212303.
[31] M.C. de Marneffe, C.D. Manning, J. Nivre, D. Zeman, Universal dependencies, Computational Linguistics. 47 (2021). https://doi.org/10.1162/COLI_a_00402.

Acknowledgments

This paper is funded by the Eureka ITEA4 FireBIM project (22003) and the Netherlands Enterprise Agency.