

# Automating building regulations conformance checking using a semantic approach - preliminary results

M. Bernert<sup>2,†</sup>, F. Ramparany<sup>1,\*</sup> and T. Hassan<sup>3,†</sup>

<sup>2</sup>*HawAI.tech, 7 Rue Antoine Polotti, 38000 Grenoble, France*

<sup>1</sup>*Orange 3 Massifs, 22 chemin du Vieux Chêne, 38244 Meylan, France*

<sup>3</sup>*Orange Atalante, 2 Av. de Belle Fontaine, 35510 Cesson-Sevigné, France*

## Abstract

Construction project management has improved in the last decades thanks to the Building Information Modeling paradigm (BIM), whose main principle is to use an exhaustive numerical representation of a building. In this paper, we focus on the application of Semantic Web technologies for checking building conformance to legal regulations. In our case, buildings are modeled using an ontology based RDF graph, and our goal is to check its conformity to accessibility regulations. We use the SHACL semantic data validation language, to perform this automated checking. We selected some relevant regulations to be modeled in SHACL and tested the method on a test building model. We were able to validate this method and plan to develop a methodology aiming at producing a SHACL model for any kind of regulations. Using our approach makes it possible to check conformance of a building even before the building is constructed or at early stages of its construction, introducing a form of “Accessibility by design”.

## Keywords

accessibility, BIM, semantic web, OWL, RDF, SPARQL, SHACL, Thing’In

## 1. Introduction

Assessing building conformance to norms and regulation is tedious and prone to error due to the complexity of law texts or incompleteness of checking process. The assessment process requires a substantial amount of expertise. The target conformity standard takes the form of synoptic tables, timetables, global and detailed diagrams. Those documents summarize the obligations and procedure: diagnosis, authorizations, certificates, technical inspection, possibilities of exemptions, etc. It also details the logical order of the chain of travel, the technical prescriptions to be observed for all types of buildings, including establishments open to the public, public facilities, collective residential buildings and individual houses.

---

*SemIIM’23: 2nd International Workshop on Semantic Industrial Information Modelling, 7th November 2023, Athens, Greece, co-located with 22nd International Semantic Web Conference (ISWC 2023)*

\*Fano Ramparany


†These authors contributed equally.

✉ marie.bernert@hawaitech.com (M. Bernert); fano.ramparany@orange.com (F. Ramparany);

thomas.hassan@orange.com (T. Hassan)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

The evolution of laws and regulations, requires conformance assessment professionals to regularly update their expertise. Besides, existing buildings structures and layout might evolve, to satisfy their occupants needs. Keeping the building respectful of the of current regulation and laws, is a challenging necessity.

In this paper we propose to automate this process using the following elements:

- A digital model of the building based on a semantic model
- A logical representation of the conformance regulation
- A Digital Twin (DT) platform called Thing'In providing the above features
- A proof mechanism for checking the conformance of the building to the conformance regulation

Thus, complying to new regulation will simply amount to modify or add new conformance rules. Similarly if the building has been reworked, its model and associated conformity checking service, can easily be reused to check that the building still conforms to the regulation.

Using our approach makes it possible to check conformance of a building even before the building is constructed or at early stages of its construction, introducing a form of “Accessibility by design”. Taking all dimensions of the design space into account early, when an artifact has proved cost effective in many domains, as it is generally expensive to modify the artifact afterwards, if a wrong decision has been taken.

## 2. State of the Art

Not much work has addressed the objective described above. In the following, we describe bodies of work that are related to our topic and have partially addressed our target.

Josefiak and his team have developed the SWOP [1] platform, which is a Semantic Web based Open Engineering Platform ontology that allows parametric modeling of products rules set, to model user requirements. According to the latest *Description of Work* Josefiak recommends and facilitates a Semantic Product Modelling approach for engineering including the choice of technologies and tools to be used in SWOP.

The Product Modelling Ontology (PMO) developed by SWOP has sufficient power to make an end-user product ontology for any parametric/configurable product type. This ontology models the product from a solution perspective ('what is possible'). The same ontology can be extended with an end-user rule set representing User Requirements (UR) like fixed single values, continuous or discrete ranges (logical OR-sense), including enumerations, regarding all relevant product/component datatype/object properties modelled taking into account their underlying datatypes/ranges.

Ontologies such as BOT [2] and ifcOWL [3], and graph based models such as Brick [4] and RealEstateCore [5] have recently been defined that target the building and architectural domain.

In [6], Fahad and Andrieux use SWRL to ensure conformance of digital building models to rules and standards regulation. They introduce a framework for mapping certification rules over

BIM to enable the compliance checking of the repository through the digital building model. They introduce an extension of IfcOWL ontology with bSDD vocabulary (i.e., synonyms and description) as enriched elifcOWL ontology to deal with the same abstract concepts or physical objects and compare mvdXML and SWRL technologies for the model instance verification and conformance checking of Industry Foundation Classes (IFC) models. However they mention limitations of a pure rule based and attribute restrictions approaches, as they are unable to handle some types of constraints such as cumulative constraints.

In [7] Greenwood and his team address automated compliance to building regulations. They review previous research into automated code compliance, identifies the key issues for future development and examines the causes of information paucity for compliance checking in the current generation of BIM tools. They conclude with an incentive to accelerate technical developments in Building Information Modelling (BIM), which offers the potential for a new generation of software tools that can automate the checking of compliance with building codes, thus improving the efficiency of building design and procurement. To attain these efficiencies designers must change their working practices and move away from the definition of a building in multiple and disparate documents to a single coherent building model from which the documentation is generated. Theoretically, this building model could contain sufficient information to respond to interrogation at the level of building code compliance, though in practice only a percentage of the required information is normally present.

In [8] Sijie Zhang addresses construction safety by applying automated safety rule checking to Building Information Models (BIM). Algorithms that automatically analyze a building model to detect safety hazards and suggest preventive measures to users are developed for different cases involving fall related hazards. As BIM is changing the way construction can be approached, the presented work and case studies extend BIM to include automated hazard identification and correction during construction planning and in certain cases, during design. A rule-based engine that utilizes this framework is implemented on top of a commercially available BIM platform to show the feasibility of the approach. As a result, the developed automated safety checking platform informs construction engineers and managers by reporting, why, where, when, and what safety measures are needed for preventing fall-related accidents before construction starts. The safety area reviewed is fall protection. An example case study of such a system is also provided.

In [9], Chi Zhang and his group report on a prototypical implementation of a model view checker for model instance validation of Industry Foundation Classes (IFC) models. This checker is developed based on the open standards mvdXML as the format for structuring validation rules and the BIM Collaboration Format (BCF) to issue reports as a result of the checking process. The checker is implemented on top of the open source bimserver.org framework. The research presented here has two main aims: (1) to develop an open source IFC validation tool based on flexible and standardized method; (2) to identify issues and capabilities of the current mvdXML rules based on real-world scenarios and to develop stable and easy-to-use IFC validation methods using open standards. Two BIM operational standards required by local building regulations and laws, the Dutch Rgd BIM Norm, and the Norwegian Statsbygg BIM Manual are used to validate both the mvdXML standards capabilities and the tools implementation. The rules from these

standards are categorized into different rule types and converted to mvdXML templates and rules. These rules are then tested using a prototypical, open source software tool. By combining this tool with a BCF server they demonstrate the deployment of such automated checking procedures in real working processes. Based on these experiences, a detailed discussion about identified issues is provided as the starting point for the future research and a feedback to standardization organizations. Similarly to Fahad [6] they advocate for exploring models and languages which expressiveness go beyond that of mvdWML and rules's ones.

In [10], Werbrouck and al. report a first attempt to apply SHACL with a checking approach for distributed building data. Currently, the BIM focus lies on file-based collaboration, although with the rise of semantic web technologies, the benefits of web- and data-based collaboration for the Architecture, Engineering and Construction (AEC) industry come within reach. A web-based AEC industry that relies on Linked Data can provide various advantages compared to 'classic' BIM practice, e.g. regarding interdisciplinarity, linking across domains and logical reasoning. In their paper, they investigate Linked Data rule checking mechanisms on decentralized building datasets. The Semantic Web standard Shapes Constraint Language (SHACL) is used to check a Linked Data building model that is hosted on multiple data pods. The Social Linked Data (Solid) ecosystem is used as set of conventions and tools for creating decentralized applications.

In [11] Fierro and al. introduce the notion of "semantic sufficiency" as a practical principle for metadata model creation that ensures that the model contains the necessary information to support a given set of applications. Their use of SHACL has proven to be an efficient approach when formalizing and automating architectural constraints and technical requirements for supporting smart building applications.

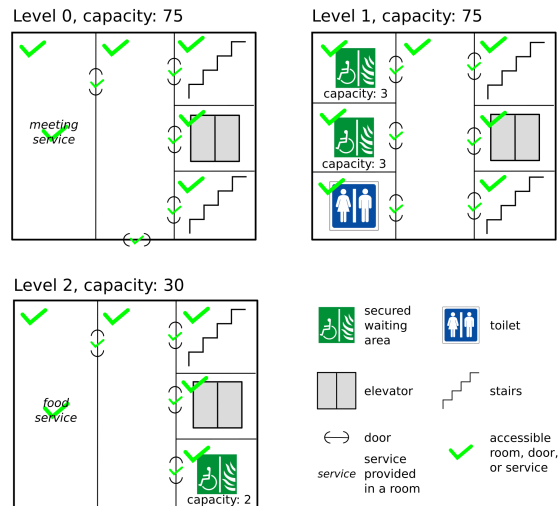
From our first analysis of the works reported above, a common ground is to use Building Information Modelling (BIM) as a baseline. Since the early 2000s, the building industry has been steadily embracing the concept of BIM. Although Werbrouck and his team have successfully used SHACL to ensure a seamless modeling of a building over distributed pods, we believe that SHACL can also beneficially be used to overcome the limitations of tools such as mvdXML and rule based systems for automated building legalx regulations checking.

### **3. Use Case**

We have defined a small 3 storeys building, which plan is displayed on figure 1. Our building is fictional but representative of a real one. Our motivation is to keep it simple enough to be used as a pedagogical tool that we will use to illustrate our solution.

The main entrance of the building is located at the ground floor. When entering the building you have on your left two staircases leading to the upper floors and an elevator in between. On the same floor you have a meeting room on the left. On the first floor you have two EAS on the right and the restrooms on the left. When arriving at the second floor from the stairs, you find the cafeteria in front and a third EAS on the left corner.

The traditional way of checking the building conformance to accessibility regulation is to request the assistance of a specialized agency. This agency will send an expert which will visit



**Figure 1:** Building Map

and carefully examine the building, to make sure that its infrastructure, facilities and equipments satisfy the rule in force in the building accessibility regulation.

In figure 4, we show such a rule (for now, just ignore the annotations made on the document, we will come to them later in section 5.3). In contrast to the simplicity of the building described above, here we have decided to choose one complex description. Our motivation is that if we are able to support a complex description we can handle simpler ones: as one says “who can do more can do less”

The description we have selected is drawn from a self-diagnostic tool called OCARA<sup>1</sup>. OCARA connects to a web server which makes available the content used for audits in the form of benchmarks. A benchmark group together a set of questionnaires which present and control the rules that apply to a given building object (parking spaces, paths, barriers, inclined plane, intercoms, room, door, etc). With OCARA, the expert is guided through the different steps to follow to carry out an audit. The expert enters the route to be audited as an ordered list of building object to be checked. Each object encountered in during the route can be evaluated immediately or recorded to be evaluated later.

As a result of an audit, the expert has to fill an audit report which summarizes the obstacles encountered by type of disability and to provide an accessibility score. She/He has also to specify which accessibility faults she/he has encountered. With OCARA the expert can illustrate or comment his finding with texts, photos or audio recording.

In the following we explain how the different building objects mentioned above can be formally described, using Building Information Models (BIM) and stored as a graph in the Thing’In [12] semantically enhanced graph database. As introduced earlier, such models will make it possible to automate the checking process.

<sup>1</sup>OCARA is an in-house mobile application which inventories buildings accessibility and security rules. OCARA helps audit officers to check building conformance in-situ.

## 4. Requirements

Thing'in [12] is a DT platform developed by Orange, that we have used to instantiate the model of the building. This model contains the structural and semantic description the building introduced in section 3. This description is based on a graph, similar to a social network, whose nodes are, in place of people, objects (not necessarily connected), physical entities and systems of all kinds that make up these environments. Thing'in makes it possible for different and so-far-vertically-integrated IoT applications in these environments to locate and share their data sources on the basis of the information stored in the Thing'in graph, much as a search engine does for the Web. Thing'in is based on a high-level information model (NGSI-LD <sup>2</sup>, now standardized by ETSI <sup>3</sup>), geared to its comprehensive value-added property graph store, but supports import and export of open data in standard "linked data" formats.

One particular policy that Thing'In adopts is to ensure that the nodes and relations contained in its graph database comply to well defined ontologies.

As prescribed by cognitive engineering methodologies and good practices, we have looked up existing ontologies that cover our universe of discourse, i.e. the scope of concepts and issues that we need to model in our application. They include the following ontologies: IoTA <sup>4</sup> to model IoT devices and data measurements, SAREF <sup>5</sup> to model Smart appliances, DogOnt <sup>6</sup> and IFC <sup>7</sup>.

We indirectly use some higher level ontologies such as the time ontology <sup>8</sup> and the WGS84 (positioning and geolocation) ontology <sup>9</sup>, because they are themselves imported by the IoTA and SAREF ontologies.

Figure 2 displays the Thing'In knowledge graph which models the building introduced in section 3. Nodes of the graph represent components of the building. For example the node under focus on the right of the figure is a door and is an instance of the class "Door". Each node of the graph is an instance of one class defined in the ontology. The classes required to represent the entire building are listed in the left legend box.

In order to discover more about the semantic of that door, we have to navigate around the neighbourhood of the node in the knowledge graph. By clicking on the node we can focus on its immediate vicinity and find out that this door gives access to "Room1" as shown in figure 3. This information is modeled as a link labeled "leadsTo" between the node representing the door and the node representing the door. The list of available link labels is given at the bottom of the left legend box in figure 2.

Such semantic modeling of the building makes it possible to apply other semantic web technologies such as SHACL, as will be shown in the following.

---

<sup>2</sup>NGSI-LD: [https://www.etsi.org/deliver/etsi\\_gs/CIM/001\\_099/009/01.04.01\\_60/gs\\_cim009v010401p.pdf](https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.04.01_60/gs_cim009v010401p.pdf)

<sup>3</sup>ETSI: <https://www.etsi.org>

<sup>4</sup><https://github.com/supdey/IoTA-ontologies>

<sup>5</sup><https://saref.etsi.org/>

<sup>6</sup><http://iot-ontologies.github.io/dogont/>

<sup>7</sup><https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/>

<sup>8</sup><https://www.w3.org/TR/owl-time/>

<sup>9</sup><https://www.w3.org/2003/01/geo/>

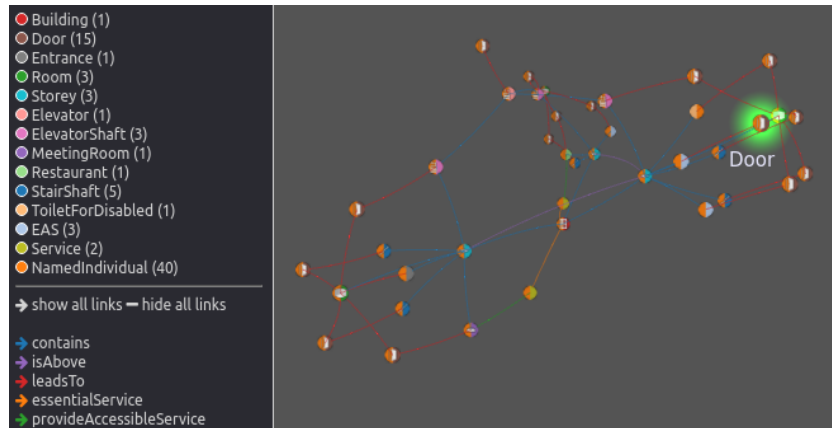


Figure 2: Building Thing'In graph model

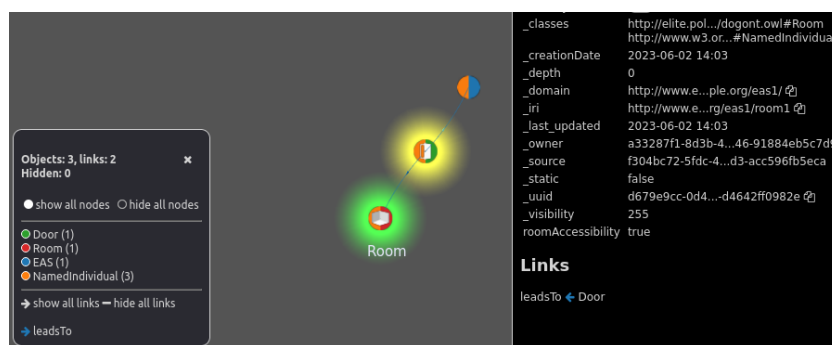


Figure 3: Door context

## 5. Our Solution

We now describe our solution. As concluded in the state of the art section 2 we will define SHACL shapes for each of the chosen regulations, in order to check if the building model has the desired properties. To be able to assert some accessibility properties about the building. We thus added some accessibility modeling elements to our building ontology. We also introduce inference rules to infer implicit accessibility properties. The following paragraph describes these three elements of our implementation

### 5.1. Accessibility Ontology

Our accessibility ontology contains the elements described in table 1. This ontology introduces a class EAS for secured waiting areas (in French: *“Espace d’Attente Sécurisé”*), a class Service and a class ServiceType representing service type. The DataProperties introduced allow to set the accommodation capacity of a room, a storey or a building, and to assert that an element is compliant with accessibility regulation. It also introduces the concept of service in a wide sense and in particular services provided in buildings.

Classes	Subclass of	Comments		
EAS Service BuildingService	Room Service	Secured Waiting Area (Espace d'Attente Sécurisé)		
ObjectProperties	SubPropertyOf	Domain	Range	OWL types
essentialService provideService provideAccessibleService accessibleWayTo accessibleEvacuationTo	provideService accessibleWayTo	Building Room or Door Room or Door	Service Service Service Room or Door Room or Door	Symmetric, Transitive Symmetric, Transitive
DataProperties	SubPropertyOf	Domain	Range	OWL types
storeyCapacity roomCapacity roomAccessibility doorAccessibility		Storey Room Room Door	integer integer boolean boolean	Functional Functional Functional Functional

**Table 1**  
Accessibility ontology

The data properties allows to assert that a building should provide some essential service (`essentialService`), actually provides a service (`provideService`), and provides a service in an accessible manner (`provideAccessibleService`). The properties `accessibleWayTo` and `accessibleEvacuationTo` allows to assert that a disabled person is able move from one place to another. The difference is that elevators are not allowed for evacuation. These properties are not given in the model and should be inferred by some reasoning. The data properties `storeyCapacity` and `roomCapacity` allows to assert the accommodation capacity of a storey or a room. The data properties `roomAccessibility` and `doorAccessibility` allow to assert that a room or a door is accessible according to the corresponding regulations (these properties are given in our test case).

## 5.2. Inference Rules

There are several properties that we want to infer from the given building model, before checking conformance using SHACL shapes. They should reassure with respect to the following questions:

- Is it possible for a disabled person to move from one room to another room?
- Is it possible for a disabled person to move from one room to another room in evacuation condition?
- It is possible for a disabled person to use the essential services of the building, taking into account the path to access the service?

As it is not possible to answer these questions using OWL reasoning, we added some inference rules to the ontology. Three rules were defined for each of these questions. First, we know which rooms connect each door thanks to the `leadsTo` property. If a door leads to a room and both the room and the door are accessible, we consider that there is an `accessibleEvacuationTo` relation between the door and the room. The inverse (evacuation from the room to the door) can be inferred as the property is symmetric. Evacuation between distant rooms and doors can then be inferred thanks to the transitivity of the property. The `AccessibleEvacuation` rule can thus be written as follows:

```
leadsTo(?d,?r) AND roomAccessibility(?r,true) AND doorAccessibility(?d, true) THEN accessibleE-
vacuationTo(?d,?r)
```



accessibleEvacuationTo is a sub-property of accessibleWayTo. Indeed, when a disabled person can evacuate from one place to another, it can also use the same path when not in evacuating. Additionally, it is possible to use elevators. We consider that there is an accessible path between two elevator shafts if they contains the same elevator and if the elevator is accessible. All accessible paths can then be inferred by transitivity. We thus write the Accessible Way rule as follows:

```
Elevator(?elev) AND ElevatorShaft(?shaft1) AND ElevatorShaft(?shaft2) AND roomAccessibility(?elev,true) AND contains(?shaft1,?elev) AND contains(?shaft2,?elev) THEN accessibleWayTo(?shaft1,?shaft2)
```

Finally we consider that a building provide an accessible service if it contains at least one room which provides the service in an accessible manner. The room should be accessible from at least one entrance. This Accessible Service rule can be written as the following inference rule:

```
Building(?b) AND room(?r) AND Storey(?st) AND Service(?s) AND Entrance(?e) AND provideAccessibleService(?r, ?s) AND contains(?b,?st) AND contains(?st,?e) AND accessibleWayTo(?e,?r) THEN provideAccessibleService(?b,?s)
```

### 5.3. SHACL Shapes

In order to define SHACL shapes proceeded with a knowledge elicitation phase where we have analyzed the regulations documents using a technique called “protocol analysis” [13]. In a nutshell, this technique consists of annotating the document with labels that defines types of information such as type of constraints, on which these constraints should be applied. Those labels are assigned to words in the documents which are semantically related to these classes or constraints. A typical result of this technique is displayed in figure 4. On this figure tokens or paragraphs in the text have been tagged as “classes”, “properties” or “constraints”, and relations have been drawn between tokens which are involved in the same constraint.

This analysis provides a structured way to identify the constraints that will be formulated as SHACL shapes, and components such as the type of the target node as well as its relations with neighbor nodes that will be involved in the shape. These elements are detailed in paragraph 5.4.

### 5.4. SHACL Concepts and Principles

SHACL shapes are designed to check if an RDF data satisfies some constraints. They can thus be used for our conformance checking problem. We implemented several shapes for each of the regulation item to be checked. Three such items are: The number of Secured Waiting Areas, the secured Waiting Areas’Capacity and the service accessibility. Due to space constraint, we will detail only the second item (Secured Waiting Areas’Capacity) here.

We defined the following SHACL Shapes to check the capacity of secured waiting areas:

```
ex:EasCapacity
  a sh:NodeShape ;
  sh:targetClass access:EAS ;
  sh:property [
```

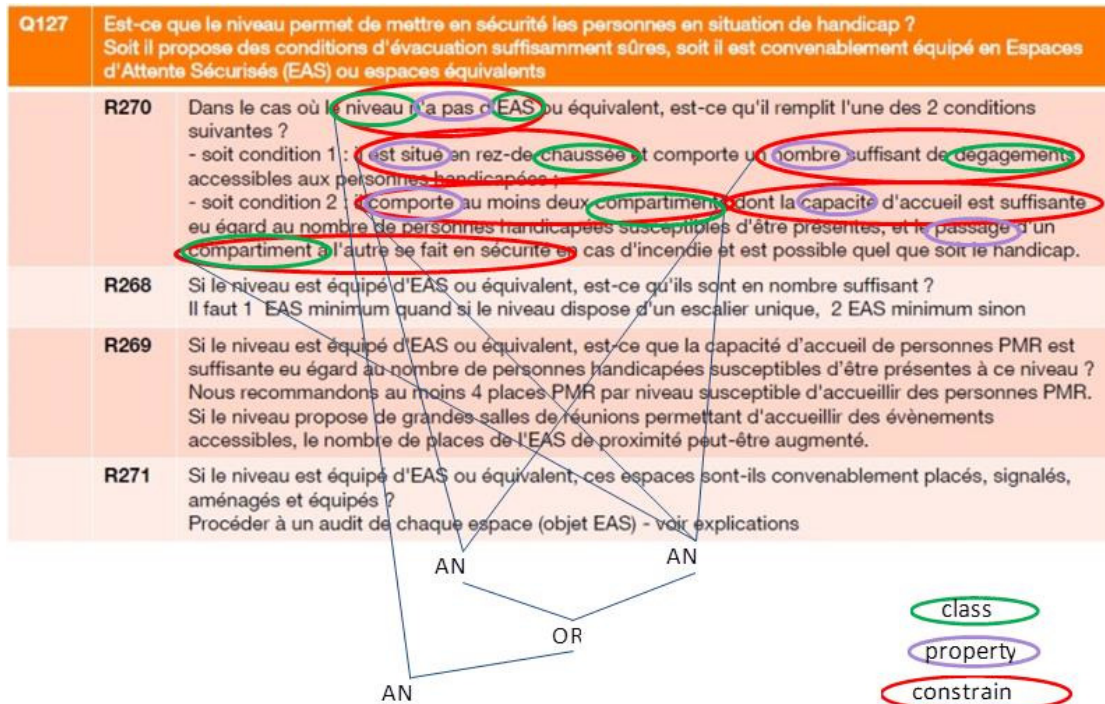


Figure 4: Accessibility regulation document protocol analysis

```

sh:path access:roomCapacity;
sh:minCount 1;
sh:maxCount 1;
sh:message "Eas capacity is unknown";
];
sh:property [
sh:path access:roomCapacity ;
sh:minInclusive 2 ;
sh:message "Eas capacity should be at least 2";
] .

```

This first shape EasCapacity targets secured waiting areas and ensures that their capacity is known and at least 2.

The second shape CumulatedEasCapacity is a complex one as it will use a SPARQL constraint, targeting storeys.

```
ex:CumulatedEasCapacity
  a sh:NodeShape ;
  sh:targetClass dogont:Storey ;
  sh:property [
    sh:path access:storeyCapacity;
    sh:minCount 1;
    sh:maxCount 1;
    sh:message "Storey capacity is unknown";
  ];
  sh:sparql [
    sh:message "Cumulated Eas capacity is insufficient in the storey";
    sh:select """
      SELECT $this
      WHERE {
        $this access:storeyCapacity ?storeyCapa; dogont:contains ?eas .
        ?eas a access:EAS; access:roomCapacity ?easCapa .
      }
      GROUP BY $this ?storeyCapa
      HAVING (?storeyCapa>(SUM(?easCapa)-1)*50)
      """ ;
  ] .
```

Its first constraint simply ensure that the storey capacity is known. The second constraint is a SPARQL constraint. The SPARQL request retrieve the capacity of the storey and the capacities of all secured waiting areas contained in this storey. The GROUP BY and HAVING commands allows to compute the sum of the secured waiting areas capacities and filter out the storeys having the desired property. If the request return a non-empty result, the shape raise an error.

## 5.5. Prototype Implementation

To test our automated conformance checking method, we focus on French accessibility regulations. We selected some rules about emergency evacuation and essential service accessibility, which are rules that involve the global structure of the building and are thus not trivial to check. These global rules requires to check some local accessibility conformance, such as if a door or a room is accessible to a disable person. Here we simply assume that information about these local accessibility is given. The checking of these local accessibility conformance could be also automatically checked in a future work, or left to a human expert.

The rules we selected for automated conformance checking are the following : – Secured waiting area number : each level of a building should have a minimal number of secured waiting area. If the level has an accessible access to the outdoor, no secured waiting area is required. Otherwise, if the level is such that only one stair is required for evacuation, one secured waiting area is required. Otherwise two secured waiting area are required. – Secured waiting area capacity : The minimal capacity of a secured waiting area depends on the level accommodation

capacity. This minimal capacity is 2 when the level capacity is less than 50. It is increased by one for each additional slot of 50 people in the level capacity. — Service accessibility : A building is intended to provide some essential service. These services should be accessible for disabled people. In particular, toilets should be accessible.

## 6. Results

Our conformance-checking process using OWL, SWRL, and SHACL, was tested using both the Protégé software [14] and the Thing'In platform. Protégé provides a graphical interface to explore ontologies and RDF data. It also provides some OWL reasoners and plugins to handle SWRL rules and SHACL shapes. Using the method described in section 3.4, we were able to validate that the test building conforms to the regulations, and that the expected violations arise when it is modified in a certain way. Three different OWL reasoners were tested: Fact++, HermiT and Pellet. Pellet was the only reasoner able to infer all necessary information, combining OWL axioms and SWRL rules. We found that experimenting with Protégé is a convenient preliminary approach to test and compare different tools.

For a more streamlined and industrial deployment of the process, we migrate the Protégé project to the Thing'In platform. Although Thing'In provides less semantic web tools, it provides the ones we needed and we could benefit from additional functions such as rich visualisation interfaces which enable a userfriendly navigation throughout knowledge graphs.

To our knowledge our approach is the first building compliance checking method using exclusively standard semantic web technologies. Some other works such as Werbrouck et al. [10] and Fierro and al. [11] also used SHACL but not for regulation compliance checking purpose.

Modeling a few regulations gave us some methodology elements to model more rules. Building regulations often state that an object in some context should satisfy some constraints. Our method to model such a rule was to implement an intermediate shape for the context in which the rule applies, and an intermediate shape for the constraints that have to be satisfied. We then implemented a validating shape, targeting the above-mentioned object, and declaring that if the focus node satisfies the context shape it should also satisfy the constraint shape.

## 7. Discussion and Conclusion

So far our preliminary results are promising. However, future work has to be done to define a precise methodology, able to handle different kinds of regulations, such as regulations involving numerical computations. It could also be interesting to implement a tool to assist non-SHACL experts in modeling regulations. Another weak point of this conformance-checking process is that it involves three different technologies. This causes some reasoners to fail to validate the data. It is also more difficult to define a clear method to create rules when different languages are combined. We plan to investigate the possibility to unify the rules modeling in a single language, which would improve the quality of the method.

Finally, it would be interesting to test the performance of this method on more realistic building models. For comparison, the model for the Orange building in Meylan, which can be

found in Thing'In, contains about 6000 elements, amongst which about 800 rooms. Our test building contains only 40 elements and 17 rooms.

## References

- [1] F. Josefiak, H. Böhms, P. Bonsma, M. Bourdeau, Semantic product modelling with SWOP's PMO, 2008, pp. 95–104. doi:10.1201/9780203883327.ch11.
- [2] M. H. Rasmussen, M. Lefrançois, G. Schneider, P. Pauwels, Bot: the building topology ontology of the w3c linked building data group, *Semantic Web* (2020). doi:10.3233/SW-200385.
- [3] P. Pauwels, T. Krijnen, W. Terkaj, J. Beetz, Enhancing the ifcowl ontology with an alternative representation for geometric data, *Automation in Construction* 80 (2017) 77–94.
- [4] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M. B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick: Towards a unified metadata schema for buildings, in: *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, BuildSys '16*, Association for Computing Machinery, New York, NY, USA, 2016, p. 41–50. URL: <https://doi.org/10.1145/2993422.2993577>. doi:10.1145/2993422.2993577.
- [5] K. Hammar, E. O. Wallin, P. Karlberg, D. Hälleberg, The realestatecore ontology, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), *The Semantic Web – ISWC 2019*, Springer International Publishing, Cham, 2019, pp. 130–145.
- [6] M. Fahad, F. Andrieux, Towards mapping certification rules over bim (2016).
- [7] D. Greenwood, S. Lockley, S. Malsane, J. Matthews, Automated compliance checking using building information models, *COBRA 2010 - Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors* (2010).
- [8] S. Zhang, J. Teizer, J.-K. Lee, C. M. Eastman, M. Venugopal, Building information modeling (bim) and safety: Automatic safety checking of construction models and schedules, *Automation in Construction* 29 (2013) 183–195. URL: <http://www.sciencedirect.com/science/article/pii/S0926580512000799>. doi:<https://doi.org/10.1016/j.autcon.2012.05.006>.
- [9] C. Zhang, J. Beetz, M. Weise, Interoperable validation for ifc building models using open standards, *Electronic Journal of Information Technology in Construction* 20, Special issue ECPPM 2014 - 10th European Conference on Product and Process Modelling (2015) 24–39. <https://www.itcon.org/2015/2>.
- [10] J. Werbrouck, M. Senthilvel, J. Beetz, P. Pauwels, A checking approach for distributed building data, in: *Proceedings of the 31. Forum Bauinformatik*, Berlin, 2019, pp. 173–181.
- [11] G. Fierro, A. Saha, T. Shapinsky, M. Steen, H. Eslinger, Application-driven creation of building metadata models with semantic sufficiency, in: *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, BuildSys '22*, Association for Computing Machinery, New York, NY, USA, 2022, p. 228–237. URL: <https://doi.org/10.1145/3563357.3564083>. doi:10.1145/3563357.3564083.
- [12] "<https://hellofuture.orange.com/fr/thingin-la-plateforme-du-graphe-des-objets/>", 2020.
- [13] N. Shadbolt, P. R. Smart, Knowledge elicitation: Methods, tools and techniques, in: J. R.

Wilson, S. Sharples (Eds.), *Evaluation of Human Work*, CRC Press, 2015, pp. 163–200. URL: <https://eprints.soton.ac.uk/359638/>.

- [14] M. A. Musen, The protégé project: a look back and a look forward, *AI Matters* 1 (2015) 4–12. URL: <https://doi.org/10.1145/2757001.2757003>. doi:10.1145/2757001.2757003.