Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220322

Towards Automated Compliance Checking of Building Regulations: smartNorms4BIM

Ignacio Huitzil [a,b,1], Marco Schorlemmer [a], Nardine Osman [a], Pere Garcia [a], Josep Coll [b] and Xavier Coll [b]

[a] *Artificial Intelligence Research Institute (IIIA), CSIC, Bellaterra (Barcelona), Spain*
[b] *Enginyeria i Project Management (EiPM), Barcelona, Spain*

Abstract. This paper describes a preliminary approach towards automating the compliance checking of constructions with respect to building regulations. We describe a prototype that supports such automated checking by specifying regulations in terms of an ontology, and reasoning with the Building Information Models (BIM) of constructions. The first step in our approach is to translate regulations into a machine-readable format with the support of controlled natural language specifications of rules. Then, we propose a formal specification of the building regulations in OWL2, the de facto standard for ontology engineering on the web. We subsequently populate this ontology with data of real-world BIM specifications based on Industry Foundation Classes (IFC) in order to check their compliance with the formalized regulations. Finally, our prototype offers to the end-users a verification report in text and a graphical visualiser with the results of the compliance check. To explain how our prototype works and to demonstrate its applicability, we show some examples taken from a concrete use case.

Keywords. Building Information Modeling, Building Regulations, Ontologies, Rule-Compliance Checking

1. Introduction

In the field of architecture, engineering and construction (AEC), more and more standards are being used for the digitized representation of the physical and functional characteristics of a building. The regulations that affect this sector, however, are usually expressed in natural language and published in the official bulletins of local, regional, national and international governmental bodies. Therefore, the verification that the design of a building actually conforms to a certain regulation continues to be an intrinsically manual process, subject to human errors of interpretation, and it requires the experienced consultation of extensive documentation and data related to the construction.

In this paper, we describe SMARTNORMS4BIM, a prototype tool by means of which we attempt to automate part of the compliance checking process of building models as developed according to Building Information Modeling (BIM) standards [1], with

[1] Corresponding Author, E-mail: ihuitzil@iiia.csic.es.

respect to construction norms and regulations. Any progress in the automation of this process can lead to a significant reduction in time, cost and risk of error. We show a proof of concept of our prototype tool for the *Decret d'Habitabilitat* of the Government of Catalonia [2] with respect to a particular BIM model of a building, carried out under the supervision of *Enginyeria i Project Management* (EiPM), an SME that has an extensive expertise in BIM project management.

We proceed as follows: Section 2 describes our proposal. Next, Section 3 discusses the main challenges we have encountered and the solutions we have adopted. In both sections, we show some illustrative examples. Then, Section 4 compares our approach with other related work, and, finally, Section 5 we describe our conclusions and future work.

2. Approach

We use the structure proposed by Eastman [3], which considers four stages for a rule-checking process : 1) translation of rules and regulations into a formal language; 2) preparation of the building model; 3) execution of the rule-checking process; and 4) reporting back the checking results. Figure 1 shows the architecture of our proposal.

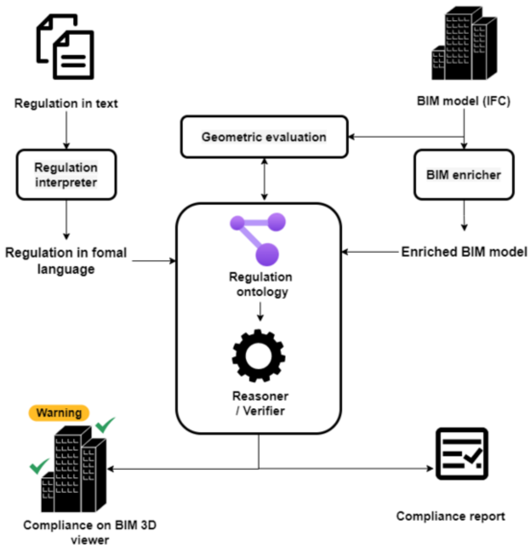

Figure 1. Architecture of the prototype.

2.1. Translation of rules and regulations into a formal language

For our approach we focused on Catalonia's building regulation and started by analyzing the "Decret 141/2012 sobre condicions mínimes d'habitabilitat dels habitatges i la cèdula d'habitabilitat" (from now on *Decret*), which specifies the minimal habitability requirements for dwellings [2]. Originally in Catalan, we translated it into English (our working language) and reviewed Annex 1 and Annex 2, resulting in a total of 51 rules.

Our approach rests on the fundamental idea to take rules and regulations as defining relevant ontological classes and to see rule-checking as the reasoning task that attempts to classify the building elements of a particular BIM model according to these ontological classes. Compliance of a particular building with the norm will be given by how the reasoner classifies particular building elements to their respective expected classes. Let's take the following example of a couple of rules taken from the *Decret*:

Example 1.
Annex 1, Rule 3.11.2:[2] ***Hygienic appliances*** *will be placed in* ***bathrooms*** *and their grouping is free (...).*
Annex 1, Rule 3.15:[3] *Equipment. All dwellings* ***must have****: (...) b) A* ***hygienic equipment*** *that consists of* ***at least one sink****,* ***one toilet*** *and* ***one shower****.*

We specify the rules in terms of ontological entities, namely by means of classes (e.g., *bathroom*, *shower*, etc.), and relations (e.g., a space, such as a bathroom, *having an equipment*, such as a shower). Furthermore, we take rules as defining our classes, in terms of necessary and sufficient conditions. For example, for Rule 3.15 we will define a class 3.15_validBathroom to be a bathroom that has as equipment at least one sink, one toilet, and one shower. Example 2 shows this definition in a description logic.

Example 2.

$$\begin{aligned} \mathit{3.15\_validBathroom} \equiv\ & \mathit{Bathroom} \\ & \sqcap\ \exists \mathit{hasEquipment.Toilet} \\ & \sqcap\ \exists \mathit{hasEquipment.Sink} \\ & \sqcap\ \exists \mathit{hasEquipment.Shower} \end{aligned}$$

Figure 2 shows the same definition for valid bathrooms, expressed in Manchester OWL Syntax, in the "Equivalent To" area of the Protégé ontology editor.

For each building entity to which some rule applies (e.g., a bathroom) we define three classes: one for *valid* entities, namely those that comply with the regulation, one for *invalid* entities, namely those that violate the regulation, and one for entities that *lack data* to be considered either valid or invalid.

The collection of all formal definitions of the *Decret*'s rules constitute an ontology with respect to which we attempt to classify the actual building elements of a particular BIM model. This classification process captures thus the compliance checking process of the BIM model with respect to the *Decret*.

[2]"Els aparells destinats a la higiene se situaran a les cambres higièniques i la seva agrupació és lliure (...)."
[3]"Dotació/equip. Tots els habitatges han de disposar de: (...) b) Un equip higiènic que estigui format, com a mínim, per un rentamans, un vàter i una dutxa."

2.2. Preparation of the building model

The BIM model we studied consists of one IFC file,[4] (in IFC2x3 schema, file size 11.2 MB). This IFC file specifies one building of four stories with seven dwellings. They are located in the following manner, two of them on the first floor and the rest (five) on the second.

We took the BIM model expressed in IFC and extracted the relevant instances of building elements that needed to be classified to the ontological classes as obtained when formalizing the *Decret*. We implemented the extraction algorithm proposed by Zhang [4]. The main idea is to populate the ontology with instances from an enriched IFC file. For each IFC class the algorithm gets the IFC property set of each instance and adds it into the corresponding ontology class. Instances are identified by a Globally Unique Identifier (GUID) and the property set of each instance is added to the ontology as data or object properties.

2.3. Execution of the rule-checking process

The classification task was done with a DL reasoner (HermiT v.1.3.8 [5]) by distinguishing those instances that are classified as valid according to a rule, from those that are classified as invalid because of some violation of a rule, and from those that are classified as lacking data. Figure 2 shows in light yellow the classification of BIM instances to the *Decret* ontology. These instances comply with Rule 3.15.

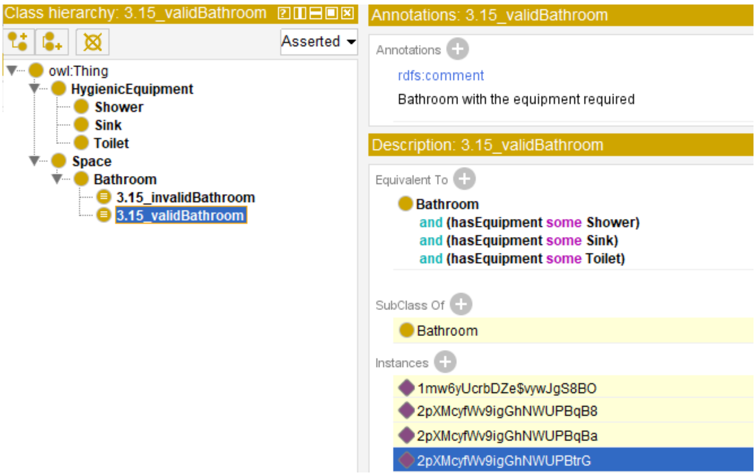

Figure 2. Example of ontology classification and instance evaluation (Rule 3.15).

2.4. Reporting back the checking results

Our approach provides two outputs for the end users, that is, a text and a visual format. The text format contains a data list. Every row of the list describes the instance GUID, the

[4]IFC file owner, http://www.eipm.es/es/

number of the evaluated rule, a tag that indicates the validation status, and the compliance description. The visual format is an IFC graphic representation. The visualiser gets the list of the text format, and classifies and shows the 3D instances coloring them according to the compliance status with respect to the regulation, namely, green if valid, red if invalid, and yellow if lacking data. Figure 3 shows the evaluation results for Rule 3.15. a) contains a list of bathrooms and c) the visualization of five valid spaces.

a) b)

Figure 3. Results: a) text format and b) graphic format.

3. Challenges

During the engineering of the *Decret* ontology for the building regulations on habitability, and the application of DL reasoners in order to check the compliance of particular BIM models with respect to the *Decret*, we encountered three main challenges: Extracting formal definitions from natural language, linking IFC specifications with ontologies, and checking geometric conditions. In this section we describe how we have addressed them:

3.1. Extracting formal definitions from natural language

Extracting formal definitions from natural language text is a challenging task due to the lack of mature natural language processing techniques. In addition, regulations may have several meanings, may be vague, may have ambiguities, and may have references to internal and external norms. Building regulations also have those problems. Let's consider Example 3, which provides additional details about hygienic equipment and bathrooms.

Example 3. *Annex 2, Rule 6.4:*[5] *Have hygienic equipment, understood as hygienic appliances that, with the corresponding running water and drainage, are intended for hygiene and evacuation of the human body, so that:*
a) It consists of at least one sink, one toilet, and either a shower or bathtub, all in good conditions.
b) The toilet must be included in a bathroom that can be made independent.

[5] "Disposar d'un equip higiènic, entès com els aparells higiènics que, amb la dotació d'aigua corrent corresponent i el desguàs, estan destinats a la higiene i l'evacuació del cos humà, de manera que:
a) Estigui format com a mínim per un lavabo, un vàter i una dutxa o banyera en bon estat.
b) El vàter ha d'estar inclòs en una cambra higiènica independitzable."

In a), the phrase *in good condition* lacks a clear definition, and so does the phrase *can be made independent* in b). Construction experts describe the latter as being isolated from other spaces by having walls or partitions as well as being accessible, for example, through a door.

In order to avoid ambiguity and facilitate the translation of building regulations into a formal representation such as description logics, we propose to use representations in some controlled natural language (CNL). We have explored Attempto Controlled English (ACE) and its sublanguage for representing description logics [6], but we have realised that current state-of-the-art CNLs are not expressive enough for many of the rules of building regulations such as the *Decret*. For instance, the representation of comparison operators and their numerical data are not supported by ACE, e.g. "height must not be less than 2.20 m" (Annex1, Rule 3.5).

3.2. Linking IFC specifications with ontologies

The second challenge encountered was linking low-level IFC building specifications with high-level ontological concepts obtained from normative text. IFC specifications are based on standard classes, but they do not describe all the detailed semantics of buildings that is needed for automatically checking their compliance with respect to building regulations. For example, Figure 4 shows an extract of an IFC specification that defines an instance with ID=#17185 using the IfcSpace class. IFC does not provide standard classes that define this instance explicitly as a bathroom.

```
#17185= IFCSPACE('2pXMcyfWv9igGhNWUPBtrG',#42,'CH',$,$,#17171,#17182,
'CH Cambra Higi\X2\00E8\X0\nica', .ELEMENT.,.INTERNAL.,$);
...
#17357= IFCPROPERTYSINGLEVALUE('ACO_Class_GuBIMclass',$,
IFCTEXT('[Uniclass 2015_SL 1.19] SL_35_80_08: Bathrooms'),$);
#17358= IFCPROPERTYSET('3EUBrWARjF5vYGum69v2OM',#42,'INCASOL_IDN','',(#17357));
...
#17385= IFCRELDEFINESBYPROPERTIES('07EPENH6bDi9T6Qfsv7eaC',#42,$,$,(#17185),#17358);
```

Figure 4. Extract of a real IFC file.

We propose to enrich the BIM model by adding properties and classification systems by inserting explicit information about IFC building elements. Figure 4 shows in the entity with ID=#17357 a property-value (the classification), which is linked to the instance with ID=#17185 in the IfcRelDefinesByProperties (ID=#17385). In particular, the entity with ID=#17357 holds a Uniclass[6] code SL_35_80_08 that states that we are dealing with a bathroom. In the extraction process of IFC data, all the explicit bathroom spaces will be classified under the Bathroom class of our ontology. Figure 5 shows the ID=#17185 instance as of class Bathroom, and its data properties in green (e.g., height, etc.) and its object properties in blue (three hygienic appliances).

[6]Uniclass is a unified classification system for the construction industry (https://uniclass.thenbs.com).

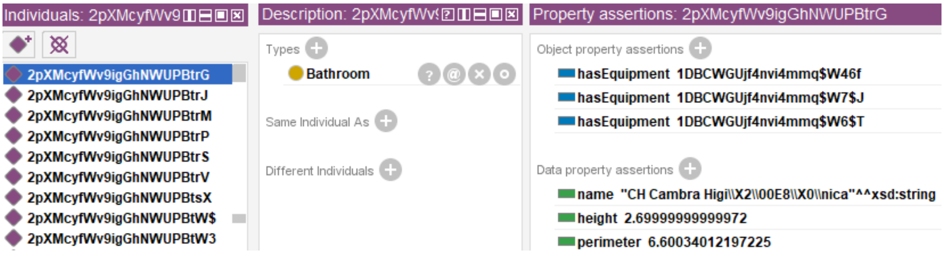

Figure 5. Example of a bathroom instance.

3.3. *Checking geometric conditions*

Some rules of building regulations require to evaluate geometrical and spatial relations between building entities. In the *Decret*, we found seven of such rules. Example 4 shows one of such type of rule, namely the one that defines when a space is practicable, which means that people with reduced mobility can access the relevant elements of the space (e.g., the bed in a bedroom).

Example 4. *Annex 1, Rule 3.4.1.e):*[7] *In practicable spaces, it should be possible to inscribe a circle of one meter and twenty centimeters (1.20 m) in dameter, free of the impact of the rotation of doors and of fixed equipment up to 0.70 m high (toilets and furniture). The interior routes of these spaces must have a minimum passage width of 0.80 m.*

When reading this rule, humans deduce that the interior routes of a space are required in order to allow access to the main elements of this space (e.g., a bed or wardrobe in a bedroom). Construction experts require that the access side of each element is specified, for example, the wardrobe's door would be its access side.

The compliance check of a geometrical norm with respect to an ontology is possible using external geometry algorithms and spatial reasoners. In other words, by means of geometric analysis and spatial processing on the IFC file, we update concrete ontological values of our populated ontology so as to reflect the compliance or not of the geometrical norm. Therefore, to provide a solution for addressing the geometrical requirements of the rule of Example 4, we propose employing geometrical algorithms from the CGAL library [8] for spaces in two dimensions, 2D polygons automatically extracted from IFC file. To check the requirement of the inscription of a circle, we use the skeleton algorithm for 2D polygons. The idea is to obtain a set of internal vertices, and to calculate circles of 1.20 m, with its secants and tangents as produced by the edges of the polygons. Our check looks of the absence of secants and lines inside the circle (i.e., it prevents a wall from crossing the circle). If the algorithm finds a valid circle then the ontology property of the instance is updated to reflect the compliance with the geometric condition. Figure 4 a) illustrates a 2D view of a bedroom, where the fixed equipment is in green and the inscribed circle in red.

[7] "En els espais practicables s'ha de poder inscriure un cercle d'un metre i vint centímetres de diàmetre (1,20 m), lliure de l'afectació del gir de les portes i dels equipaments fixos de fins a 0,70 m d'alçada (sanitaris i mobiliari). Els recorreguts interiors d'aquests espais han de tenir una amplada mínima de pas de 0,80 m."

[8] https://www.cgal.org.

A similar solution is implemented for the internal route. First, we employ the offset algorithm that generates a new internal area from an initial 2D polygon that shows the areas that are wider that the required passage width. We can then verify if the route has access to the fixed equipment delimited by its access sides (determined by construction experts in the design). The main door creates an access side using a bisector strategy considering the door segment. If the route connects all the access sides, then the ontology property is updated to reflect the compliance with the geometric condition. Figure 6 b) shows a 2D view of a bedroom, where the equipment is in green, an internal route is in blue, the main door segment is in orange, and access sides are indicated by the red arrow lines.

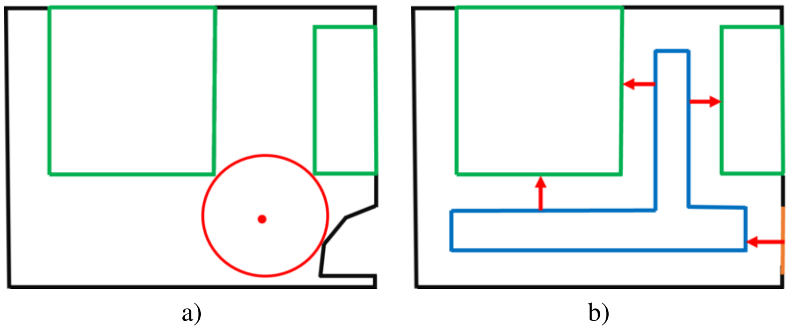

Figure 6. Geometric requirements: a) inscription of a circle b) internal route.

4. Related work

Hjelseth and Nisbet suggested a methodology for marking up the expressions of a rule according to four categories: Requirements and Applicability (i.e., what requirements apply to which entities), and Select and Exceptions (i.e., alternative subjects or other exceptions to the rule) [7] . They proposed a way to re-formulate norms based on this mark-up, and developed a software to check for rule compliance, where the 2021 version uses a conceptual graph to represent the semantics of rules. Our approach represents rules as definitions in an OWL2 ontology, and we explored the use of controlled natural languages for rule specification.

Zahng and El-Gohary use a rule-based, semantic natural language processing approach to extract norms in first-order logic from various construction regulatory documents in order to develop a compliance checking system [8,9]. In contrast, we do not focus on natural language processing, given the ambiguity of regulations, and advocate for rules to be written in some controlled natural language first, prior to extraction and formalization. In the classification of ontology instances, we use then an enriched BIM model.

A cloud-based solution to check rules called BIM-kit is proposed in [10]. The authors describe two interesting services, the rule editor and the model checking service. The first service allows the user to write the rules in a graphical mode, a restricted natural language, or a code representation system. The second service checks the regulation by employing an external app that returns the resulting list to a 3D visualizer. In our work, we use the ACE plugin[9] to write the regulation, with which we edit, view and create OWL2 ontologies.

Doukari et al. also proposed a framework for automated compliance checking based on BIM [11]. In the rule interpretation, they mention the immaturity of natural language processing technologies, and as a consequence, they code the regulation into if-then logical rules in C#. The BIM model is loaded as a data structure of partial states. Next, the rule and the building element are concatenated and evaluated as ‘pass’ or ‘fail’. The last step is a detailed report. Our approach offers a similar process with some variations, for example, using a controlled natural language that generates definitions (classes) of an ontology. From the BIM model we extract the enriched instances to populate and evaluate an OWL2 ontology, so as to make it reusable on the web. And, in addition to identifying those building elements that ‘pass’ or ‘fail’, we also categorize those that lack data to make a definite decision on their validity.

There are methods to evaluate geometrical errors (of design and modeling) and regulations concerning spatial aspects. Dinis et al. proposed a Virtual Reality check tool with which the end-users have an immersive experience in a 3D representation to find errors and to subsequently explain their findings [12]. From our experience, this evaluation is time consuming and requires experienced users. In order to evaluate spatial data of a BIM model, it would be possible to use a formal query language such as BimSPARQL [13]. In our work, we evaluate the spatial aspects of norms by resorting to libraries of computational geometry algorithms such as CGAL.

Recently, neuronal network approaches have also been explored to evaluate housing constructions based on a set of norms [14]. The authors formalize the regulation specified manually by architects to create the network. In contrast, we chose to follow a logic-based approach that does not need a training stage, as with learning techniques. In turn, we require the regulators to be familiar with controlled languages so as to be able to write the norms in a more constrained and unambiguous way, or to work collaboratively with specialists in the translation from the regulators’ natural language to the controlled one.

Finally, a popular commercial software is the Solibri Model Checker.[10] Version 2022 has 56 single rule templates to evaluate IFC models. These templates define standard checking procedures, delimited by a number of parameters. The end-users can edit the rule templates according to their requirements. To formulate the rule templates, there is a need for experienced users with an extensive knowledge of IFC.

5. Conclusions and future work

In this paper, we presented a prototype a case study for automated compliance checking of building regulations. First, we formalized a fragment of Catalonia’s *Decret d’habilitat*, which regulates the habitability requirements for buildings; we did this supported by a controlled natural language, in order to define an OWL2 ontology that captured the requirements for building elements as specified in the *Decret*. Next, we enriched a BIM model as described using IFC employing standards of classification systems such as Uniclass. This meant adding explicit information to the various building elements specified

[9] http://attempto.ifi.uzh.ch/aceview/
[10] https://www.solibri.com

in the BIM model. Then, we implemented an extraction algorithm for the enriched BIM model to populate the *Decret* ontology, and used a DL reasoner to classify the instances of the model to the ontology classes. In the last step, we generated reports for end-users, either in text or in a graphical mode, showing those instances of building elements that are correctly classified or are incorrectly classified because either they violate one or more rules of the regulation, or else lack data. As a future work, we would like to set up an empirical evaluation of our proposal using the entire *Decret*, checking the compliance of BIM models of different IFC file size, in order to evaluate our proposal in terms of expressiveness and computational cost. We also plan to study natural language processing techniques that are suitable for formalizing building regulations, seen as mathematical word problems [15].

References

[1] C. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, John Wiley & Sons, 2008.
[2] Generalitat de Catalunya, Portal Jurídic de Catalunya, DECRET 1412012, de 30 d'octubre, pel qual es regulen les condicions mrínimes d'habitabilitat dels habitatges i la cèdula d'habitabilitat, `https://portaljuridic.gencat.cat/eli/es-ct/d/2012/10/30/141,accessed:2021-01-15` (2012).
[3] C. Eastman, J. Lee, Y. Jeong, J. Lee, Automatic rule-based checking of building designs, Automation in Construction 18 (8) (2009) 1011–1033.
[4] L. Zhang, N. M. El-Gohary, Extracting Information from Building Information Models to Support Automated Value Analysis (2016) 527–534.
[5] R. Shearer, B. Motik, I. Horrocks, HermiT: A highly-efficient OWL reasoner, in: Proceedings of the 5th OWL: Experiences and Directions Workshop (OWLED 2008), 2008.
[6] K. Kaljur, N. E. Fuchs, Verbalizing OWL in Attempto Controlled English, in: In Proceedings of OWLED07, 2007.
[7] E. Hjelseth, N. N. Nisbet, Capturing normative constraints by use of the semantic mark-up rase methodology, 2011, pp. 1–10.
[8] J. Zhang, N. M. El-Gohary, Automated information transformation for automated regulatory compliance checking in construction, Journal of Computing in Civil Engineering 29 (2015) 1–16.
[9] J. Zhang, N. M. El-Gohary, Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking, Automation in Construction 73 (2017) 45–57.
[10] C. Sydora, E. Stroulia, BIM-kit: An extendible toolkit for reasoning about building information models, in: Proceedings of the 2021 European Conference on Computing in Construction, Vol. 2 of Computing in Construction, University College Dublin, Online, 2021, pp. 107–114.
[11] O. Doukari, D. Greenwood, K. Rogage, M. Kassem, Object-oriented compliance checking: an automated approach and a case study, in: Proceedings of the 2021 European Conference on Computing in Construction, Vol. 2 of Computing in Construction, University College Dublin, Online, 2021, pp. 293–302.
[12] F. M. Dinis, J. Martins, F. Sousa, B. Rangel, A. Guimarães, A. Soeiro, Virtual reality design quality-check tool for engineering projects, in: Proceedings of the 38th International Conference of CIB W78, 2021, pp. 912–922.
[13] C. Zhang, J. Beetz, de Vries, BimSPARQL: Domain–specific functional SPARQL extensions for querying RDF building data, Semantic Web (2018) 1–27.
[14] H. Arora, J. Bielski, V. Eisenstadt, C. Langenhan, C. Ziegler, K.-D. Althoff, A. Dengel, Consistency checker an automatic constraint–based evaluator for housing spatial configurations, in: International conference eCAADe – Education and research in Computer Aided Architectural Design in Europe, 2021, pp. 351–358.
[15] S. Ughade, S. Kumbhar, Survey on mathematical word problem solving using natural language processing, in: 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), 2019, pp. 1–5.