

# Capabilities of rule representations for automated compliance checking in healthcare buildings

Zijing Zhang, Nicholas Nisbet, Ling Ma<sup>\*</sup>, Tim Broyd

The Bartlett School of Sustainable Construction, University College London, London WC1E 6BT, UK

## ARTICLE INFO

### Keywords:

Automated compliance checking  
Knowledge representation  
Healthcare building design  
Building regulations

## ABSTRACT

A suitable rule representation is essential to enable automated compliance checking of building design. It encapsulates engineering knowledge and facilitates an adequate interpretation of design standards. However, existing methods have achieved limited capabilities to represent rules for automated compliance checking. Thus, they merely worked for limited types of rules. This paper aims to identify capabilities needed for rule representation by using healthcare design regulations as an example. It can serve as a foundation for developing rule engines and compliance-checking systems in the future. A four-step process was used to systematically analyse six healthcare building regulations in rule-oriented and implementation aspects. The results showed 18 capabilities for healthcare rule representation, where 16 are required, and two are desirable. This research is valuable to researchers and practitioners by providing a checklist for future representation development and criteria for assessing rule representation methods.

## 1. Introduction

In the construction industry, before the design is finalised and moving on to the construction stage, the building design must be reviewed and checked against standards and codes that are typically found in laws, regulations, requirement statements and recommendations [13,52]. Traditionally, compliance checking is a manual process conducted by domain experts. It is expensive and often leads to project delay [20]. The manual compliance checking is also error-prone; mistakes in design may lead to costly rework and poor building performance [13,33,44,51].

Automated compliance checking (ACC) has been researched in the past decades to improve the efficiency and reliability of compliance checking [64]. The whole process involves numerous actors and knowledge exchanges, as shown in Fig. 1. Among them, rule interpretation is a non-trivial task, where the rules written in natural language need to be interpreted to a machine-readable form without losing meaning [27,50,52]. The semantics, logic and the knowledge embedded in rules can only be analysed and revealed using experts' domain knowledge [50]. According to the CORENET project, such manual interpretation is time-consuming, taking up to 30% of the time to implement a design checking [49,50].

Rule interpretation typically needs to be repeated from scratch when

establishing a new ACC system, as rules are typically hardcoded. Furthermore, the old interpretations need to be modified when design regulations are updated every few years. Such modifications can only be manually conducted by domain experts [39], often with coding and modelling experts. Repeating the interpretation is time-consuming and expensive, and inconsistency and credibility issues are concerning as different experts tend to have different interpretations based on their own experience and locality [18,50].

Rule representation has been suggested as a viable solution for effectively interpreting design regulations. It does so by formalising rules and capturing embedded knowledge revealed during interpretation [50,57]. Such a representation helps retain and communicate knowledge among multiple actors in developing ACC systems. Nevertheless, there is no universally accepted representation method for rules [33], as existing methods failed to include all required capabilities for rule representation [33,62,63].

The authors believe that developing a more well-rounded representation would first need to understand what capabilities are required and desired to fill this research gap. In this study, we aim to identify a list of capabilities for rule representation through an inductive analysis of healthcare-related regulations in England. We choose healthcare regulations as subjects for three reasons: 1) healthcare projects are often very complex, with many inter-dependent sub-systems that need to be

<sup>\*</sup> Corresponding author.

E-mail address: [l.ma@ucl.ac.uk](mailto:l.ma@ucl.ac.uk) (L. Ma).

checked [14]; 2) There are many healthcare regulations and guidance issued by different agencies [37]; 3) healthcare regulations can be very confusing; they have different levels of constraints and complex hierarchies; some rules have overlap and duplications [21]. As a result, healthcare regulations are good examples for comprehensively identifying rule capabilities.

The remainder of this paper is structured as follows: Section 2 provides a review of related work, Section 3 introduces the method proposed by the authors, Section 4 identifies the required capabilities, Section 5 provides a discussion, and Section 6 concludes this paper.

## 2. Related work

A considerable amount of research has been conducted on rule interpretation and representation, mainly focusing on the classification of rules (e.g., [39,49,51,52]), rule organisation (e.g., [57]) and rule representation (e.g., [33]). Previous research on these topics is shown in Table 1. Research on rule classification and rule organisation has been closely related to rule representation studies. Firstly, rule classification studies have been frequently employed to understand rule characteristics to assist in developing rule representation methods. Secondly, rule organisation has been an important consideration when developing rule representations. Because of these close relationships, although the following review mainly focused on various representation methods, attention was also paid to analysing the rule classification and organisation methods.

Some early efforts adopted production rules to represent building codes. Production rules typically take the form of “if <conditions> then <actions>”. Ma et al. [31,32] devised a mathematical matrix representation to encapsulate civil engineers' knowledge about classifying bridge components based on their geometric features and pairwise relationships. When a model lacks object classification information, the matrix representation enables semantic enrichment through mathematical calculations. Fenves [15] proposed a decision table approach to represent rules concisely and unambiguously using production rules.

Decision tables indicate conditions applicable to a specific situation and appropriate actions based on condition values [16]. Although individual rule clauses were represented, the main defect of this method is that the relationships among rules were not represented in decision tables.

Similarly, Tan et al. [53] enhanced the conciseness and extended the expressiveness of this representation method in [15] using a new decision table to represent envelope design regulations. A set of parameters (e.g., location, type) were extracted from the rules and placed as sub-headings of the decision table. The reference index is also included, linking to the original rules and the cross-reference. However, typically only rules with a very similar set of conditions and actions can be represented in this way. In addition, this method still failed to show the logical relationships among rules.

The initial decision tables were improved to address this issue, and a SASE (Standards Analysis, Synthesis and Evaluation) model was proposed [17]. It is a four-level representation method, including an organisational network (building codes organisation), an information network (dependency relationships among provisions) that includes an individual provision level (decision tables) and data items referred to in the provisions. It is also an independent representation of the rule engine, thus allowing easy creation and modification by non-programmers. However, the main problems with this model lie in the lack of applicability to data items and overly complex precedence relationships [34].

Some studies developed a logic-based approach using parametric tables to represent building codes. Commercial software such as Solibri Model Checker (SMC) [47] was an example using this approach, where interpreted rules were programmed into computer codes by software engineers. Thus, rules are embedded in the rule engine, and there is no separate representation. A similar approach has also been seen in multiple academic studies. For example, Lee [30] adopted parametric tables to represent USA courthouse circulation rules. Soliman-Junior, et al. [52] checked UK hospital rules using SMC. To decide whether or not to check specific rules using SMC, they proposed a framework to classify rules against the nature of rules (i.e., qualitative, quantitative and

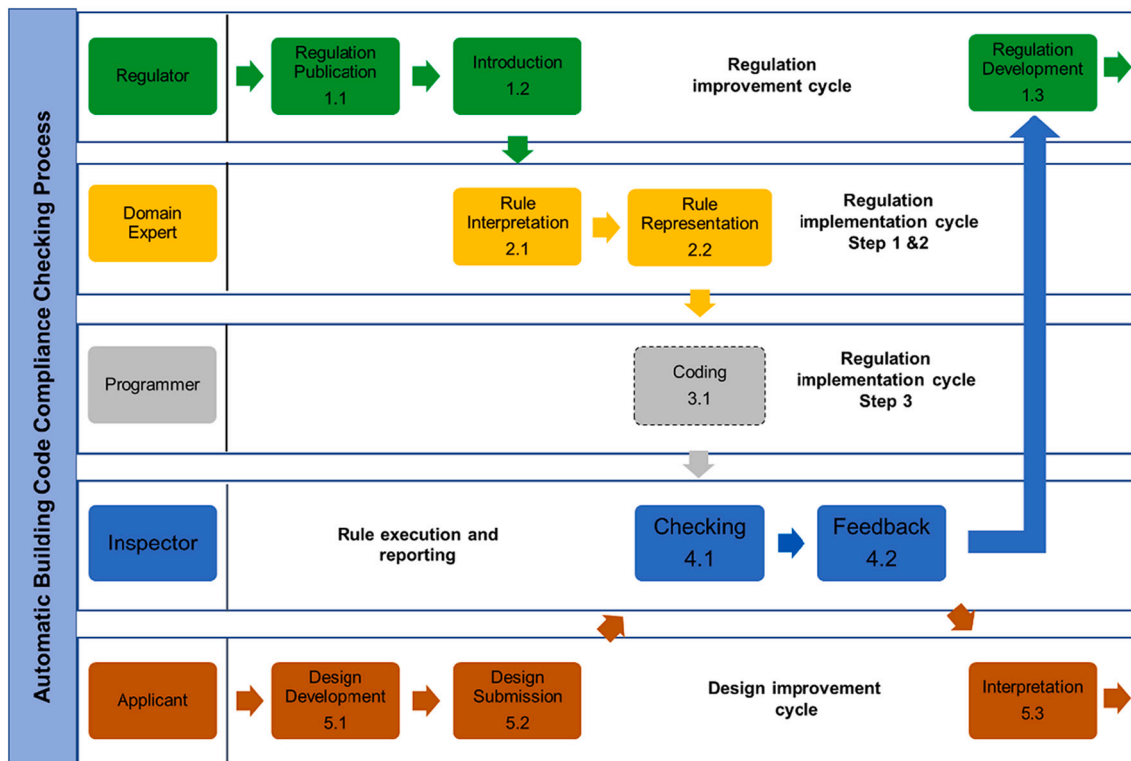


Fig. 1. Automatic building code compliance checking process.

**Table 1**  
Representation, classification and organisation methods for ACC.

Reference	Classification Method	Rule Organisation	Representation Method	Domain	Country
Fenves (1966) [15]	NA	NA	Production rule (decision table)	Steel construction	USA
Fenves, et al. (1987) [17]	NA	Cross-reference	Production rule (decision table)	Steel construction	USA
Rasdorf and Lakmazaheri (1990) [46]	Based on content	Content-based	Logic-based (predicate logic)	Steel construction	USA
Garrett Jr. and Hakim (1992) [19]	NA	Object-oriented, Class hierarchy	Object-oriented	Steel construction	USA
Yabuki and Law (1993) [56]	NA	Object-oriented, class hierarchy	Object-oriented, Logic-based	Steel construction	USA
Kiliccote, et al. (1994) [25]	NA	Context-based	Object-oriented	Building codes	USA
Yurchyshyna and Zarli (2009) [57]	Based on IFC interpretability	Content-based	Semantic-based (semantic web)	Building codes	France
Tan, et al. (2010) [53]	NA	Cross-reference	Production rule (decision table)	Envelope design	USA, Canada
Lee (2010) [30]	NA	NA	Logic-based (parametric table)	Courthouse circulation	USA
Pauwels, et al. (2011) [42]	NA	NA	Semantic-based (semantic web)	Acoustic performance	Belgium (EU)
Lee et al. (2014); Lee (2011) [28,29]	NA	NA	Language-driven (domain-specific language)	Building circulation and spatial programme	USA
Hjelseth and Nisbet (2010) [23]	NA	Cross-reference	Semantic-based (RASE)	Structure rule	USA
Preidel and Borrmann (2015, 2016) [43,44]	NA	NA	Language-driven (visual programming language)	Fire code	Germany, Korea
Solihin and Eastman (2015) [49]	Based on rule complexity	NA	NA	Fire code, accessibility	USA
Solihin and Eastman (2016) [50]	NA	Cross-reference	Logic-based (conceptual graph)	Fire code, accessibility	USA
Park and Lee (2016) [41]	NA	NA	Language-driven (domain-specific language)	Building permit codes	Korea
Zhang and El-Gohary (2016) [59]	Based on semantics	NA	NA	Ventilation code	USA
Macit İlal and Günaydın (2017) [33]	Based on interdependency	Cross-reference	Semantic-based	Housing and zoning code	Turkey
Kim, et al. (2019) [26]	NA	NA	Language-driven (visual programming language)	Building permit codes	Korea
Nawari (2020) [39]	Based on semantic and interdependency	Cross-reference	Language-driven (LINQ)	General building heights and areas	USA
Soliman-Junior et al. (2020,2021) [51,52]	Based on semantic and logic	NA	NA	Healthcare facility codes	UK
Beach et al. (2015) [3]	NA	Cross-reference	Semantic-based	BREEAM and code for sustainable homes	UK
Zhang and El-Gohary Nora (2015) [60]	NA	NA	Semantic-based (natural language processing)	Building Codes	USA
Dimiyadi, et al. (2016) [12]	NA	NA	Language-driven (Regulatory Knowledge Query Language)	Fire codes	New Zealand
Zhang and El-Gohary (2021) [61]	Based on syntactic, semantic features and computability metrics	NA	Semantic-based (natural language processing)	Building Codes	USA

ambiguous) and the possibility of translation into logical rules [51]. Despite the conciseness and capability to represent very complex rules using parametric tables, this approach suffered from limited expressiveness and difficulty in maintenance due to the hard-coded nature [50]. They were also criticised for being “black boxes” to the user [43], which lacked transparency and may lead to credibility issues.

Some other logic-based methods adopted predicate logic to represent building rules. For instance, Rasdorf and Lakmazaheri [46] developed a logic-based SASE model by formally organising the building codes using a set of predicate logic statements. In addition, they adopted an organisational sub-model with a tree structure to show the categories of rules and their linkages. This approach highlighted the relationships between the classifiers and rule clauses. However, predicate logic statements include many mathematical symbols and are often lengthy, making them hard to use and understand by domain experts.

Solihin and Eastman [50] introduced the conceptual graph (CG) method to represent building codes to improve their readability and ease of use. It was an approach with a semantic foundation in predicate logic but with graphic notations. The CG approach considered the BIM model (in IFC format) to be checked when developing rule representation. It stressed the main object(s) in each rule provision and presented the properties and relationships that need to be checked and the geometrical, mathematical algorithms and simulations required when

executing rules. The formalisation of CG representation relied on the classification of building rules. The classifications include: 1) class 1: rules that require a single or a small number of detailed data; 2) class 2: rules that require simple derived attribute values; 3) class 3: rules that require extended data structure; 4) class 4: rules that require a “proof of solution” (i.e., meets the performance requirements) [49]. Some authors (e.g., [1]) have noted the trend in regulatory content away from prescriptive regulations and toward performance regulations, implying that performance-based regulations may be outside of the scope for automated applications. However, Solihin and Eastman [49] suggested that whether ACC can check the performance-based regulations depends on whether the evidence the applicants provide is acceptable. Such classification and representation recognised the influence of rule complexity on rule representation and provided a link between the target building model and rule representation. However, using this approach, the rule representation was restrained to the IFC model, often insufficient to represent all required information in the rules, primarily abstract geometrical and topological constraints [43]. Furthermore, these methods cannot deal with results other than “pass” or “fail”, such as “unknown” and “pending”.

A handful of studies employed object-oriented thinking; they recognised the importance of organising rules based on rule contents instead of themes. Garrett-Junior and Hakim [19] proposed an object-

oriented method that organises rules around objects related to rules. Yabuki and Law [56] adopted an object-logic hybrid approach. An object-oriented modelling approach is used for the organisation and data items of the building code, and first-order predicate logic is used for representing provisions. However, both the object-oriented and object-logic models have been criticised for having a too complex class hierarchy and being cumbersome to handle [25]. To address this issue, Kiliccote et al. [25] developed a context-oriented model that organised building code around “contexts”, which are essentially a set of subclasses of the applicability constructs in provisions.

More recently, some studies explored the semantic structure of rules. Hjelseth and Nisbet [22] explored general rule features and identified four general constructs in short phrases and longer rule sections: “requirement”, “applicability”, “selection”, and “exception” (RASE). They also recognised the influence of logical connectives among different semantic constructs on the checking results and used a tree-like method to demonstrate the logical calculus [40]. In addition, the RASE method developed a dictionary to maintain the consistency of terms and deal with algorithmic calculations and simulations. The later extension of RASE further recognised the need to capture the actions when the rules have outcomes other than pass/fail [2]. It incorporated the achieved “output” and target “total” constructs to represent the point-scoring “actions” in BREEAM (BRE Environmental Assessment Method) rules [7]. Macit İlal and Günaydın [33] integrated the SASE model with the RASE method and proposed a new hybrid model, including domain level, rule level, rule-set level and management level. This model reduced redundancies of repeating definitions of the same concept using a lower-level (i.e., domain-level) library. It also emphasised the logical relationships among rule objects at the rule-set level. They also adopted a classification method, where rules were classified into linked explanatory and self-contained categories. However, this classification is not comprehensive as it is only based on rule interdependency.

There have also been attempts to use natural language processing (NLP), which emphasises rules' various syntactic and semantic features [59]. The semantic web is another semantic-based representation method that gained popularity recently [4,42,57]. For example, Yurchyshyna and Zarli [57] formalised regulation texts using SPARQL queries and query annotations. Queries were used to represent conformance constraints using an IFC-based ontology. Semantic annotations were used on queries to include all information and knowledge related to compliance checking (e.g., application level, application conditions). As the queries adopted an IFC-based ontology, this method also suffers from limited expressiveness.

The last category of rule representations is language-based methods. A typical example is the Building Environment Rule and Analysis (BERA) [29] for building circulation and spatial rules. When using the BERA language, experts interpret rules into queries and use queries to check the building model data. A domain-specific query language incorporates specific syntax and functions for building rule representation and is also easier to learn for non-programmers. Some more recent studies have focused on visual programming languages [26,43,44] as a rule representation. They balance representing complex rule logic and not requiring computer programming [50]. Wires were used to link many pre-defined method nodes with input and output ports to form rules. They are “small white boxes” with known functions, making rule-checking transparent and easy to understand by rule experts [44]. Nevertheless, deficiencies are still exhibited in visual programming languages, especially when handling recursions.

In summary, although there has been significant progress in building rule representation (Table 1), several issues exist in the current methods:

1. In most studies, the representation developed can only represent part of building rules or certain features of rules while incapable of representing others. This issue is mainly reflected in two aspects:

- a. Existing studies rarely deal with both the high and the low levels of rules, including rule provisions and rule organisation. They only have the capabilities to address a single level. For example, many studies only focused on individual requirements but ignored the context.
  - b. Most methods can only represent a selection of rule features or have limited capabilities to represent rule organisation, limiting the expressiveness and affecting the quality of representations. For example, many studies focus only on requirements expressed numerically.
2. Most research failed to recognise the importance of rule representations to be independent of the rule engine and the building data model. The dependency on the rule engine and the building data model restricts the expressiveness of rule representation and makes it hard to maintain. Rules that relate to concepts that are not explicitly represented, such as escape routes or thermal performance in IFC models, or building height in GIS models, have been dismissed.
  3. Most methods only focus on the grammar and language of rules; they rarely cope with the intensity “underneath” the rules.
  4. Previous studies have an ambiguous scale of rules; rules can be better organised and represented with a more precise definition. As a result, those approaches are only applicable to a limited scope or apply only off-the-shelf rules found in existing toolkits.
  5. Existing methods have not fully recognised the requirements for rule representations to support implementation aspects, such as credibility and human-readability, thus restricting their practical applicability [43]. In addition, many implementations depend on multiple interpretation stages, reducing the linkage to the source material.
  6. Most of the current approaches failed to prepare the representation for future modifications. Further requirements for representations may be needed to enable efficient updates. For example, the latest version of HBN 00-02 was published in 2016 to replace the previous version published in 2013. Regulatory documents are typically modified every few years. The representation method needs to be prepared for the modification to avoid rework.

This research proposed a method to systematically analyse required capabilities for rule representation, using healthcare regulations as an example to address these issues.

### 3. Methods

The authors took the constructivist worldview. The constructivist worldview aims to generate generic patterns or theory inductively [8], and in this paper specifically, generating patterns of healthcare building regulations. This paper collected healthcare facility regulations from UK government and organisation websites. The detailed data collection process is presented in Fig. 2 and explained in Section 3.1. After data collection, to aid the identification of required capabilities for healthcare rule representation, the authors proposed a method capable of systematically and qualitatively analysing the selected regulations and extracting general patterns of rules. This method considers various rule-related aspects and requirements for implementation (Fig. 2), detailed in Section 3.2.

#### 3.1. Data collection

This paper collected the sample for analysis from England regulatory documents. It includes both healthcare-specific regulations and regulations applicable to building work in general. The type of convention used by the UK government has nine constraint sequences, among which six are regarding regulatory documents, as shown in Fig. 3. In contrast, the other three (i.e., sequences 5–7) were not presented as they focus on evidence (e.g., certificates). The six constraint sequences fall into three constraint levels: regulation, requirement and recommendation. Regulation is the highest constraint level, and recommendation is the lowest

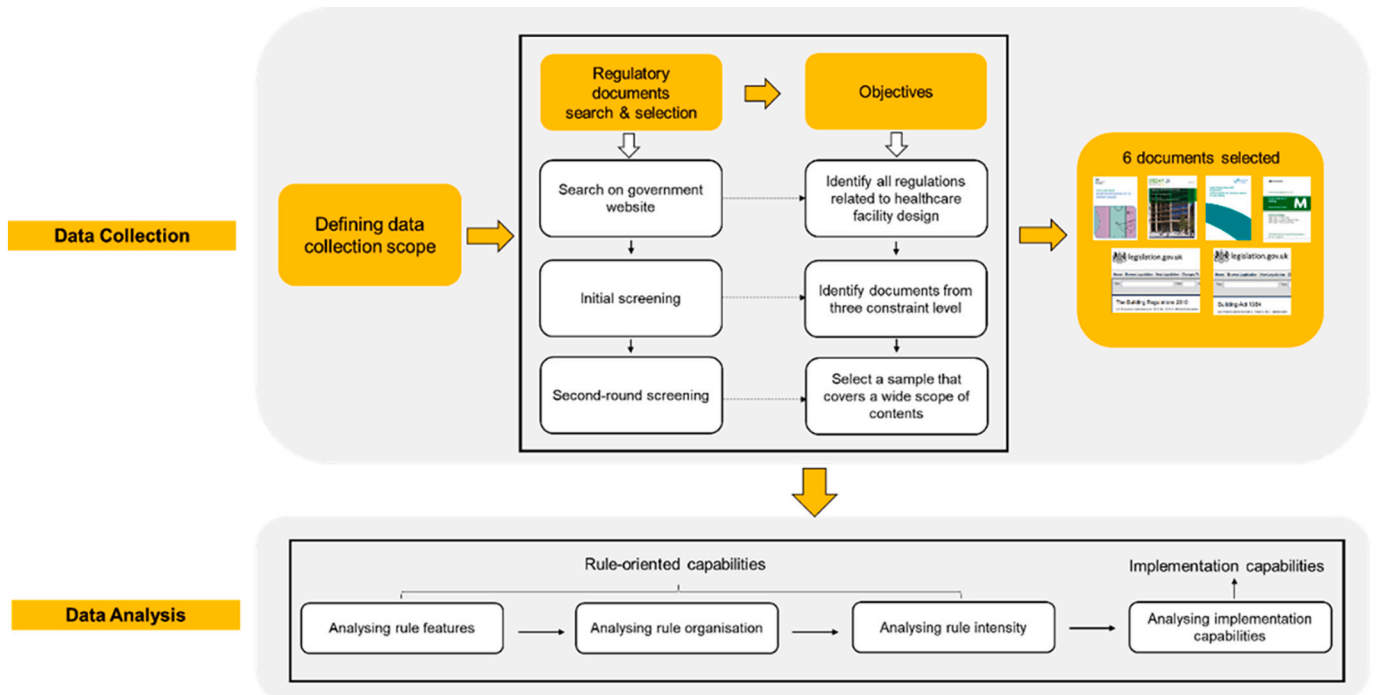


Fig. 2. Research workflow.

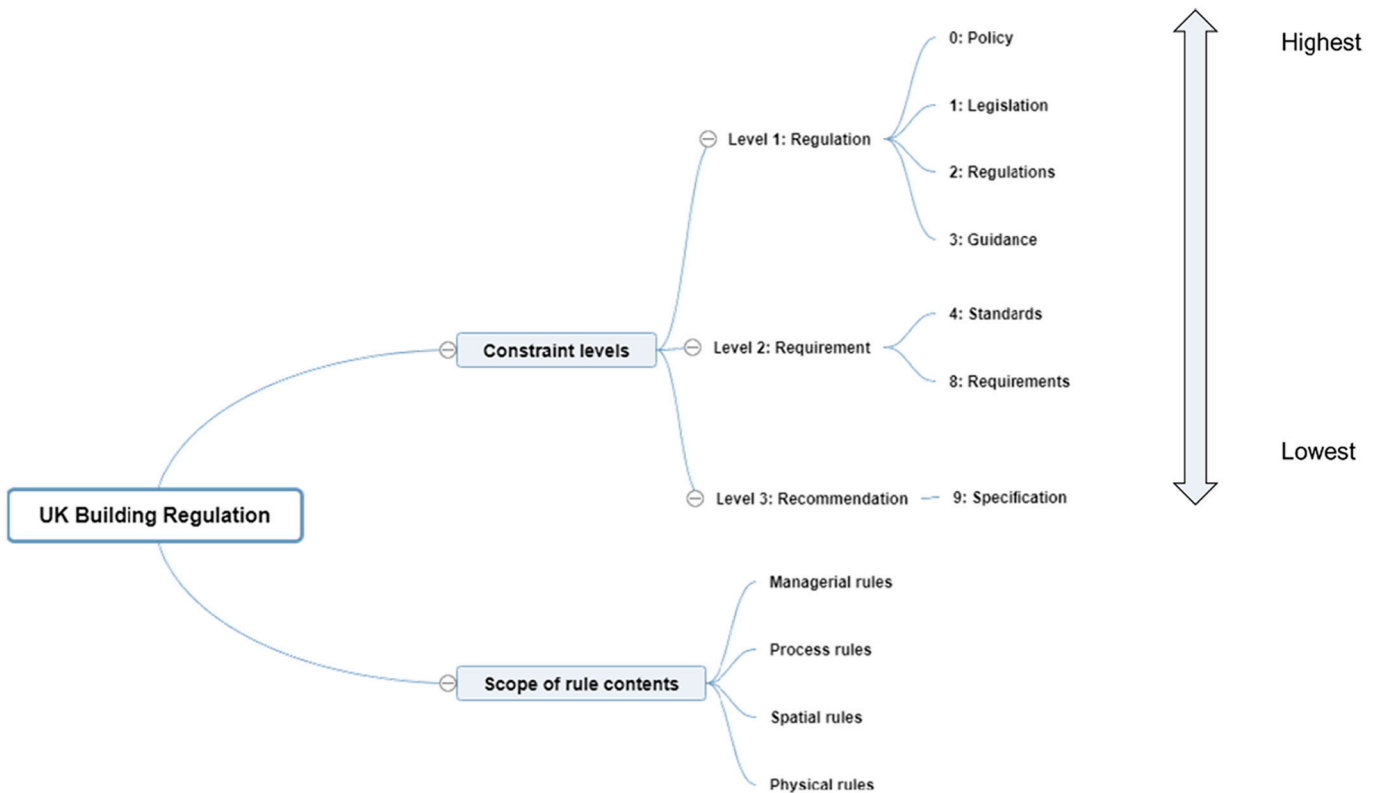


Fig. 3. Constraint levels, constraint sequence and scope of UK building regulation.

[5]. In terms of the scope of rule contents, there are mainly four types: namely managerial, physical, process, and spatial rules (Fig. 3).

The objectives of healthcare regulatory document selection are 1) to cover all constraint levels of healthcare regulatory documents; and 2) to include a sufficiently broad scope of rule contents in the healthcare domain, which helps ensure the findings are sufficiently generic. First,

the authors searched on the UK government websites to identify constraint levels of multiple regulatory documents applicable to healthcare facilities. According to [36], the Building Act 1984 [54] and Building Regulations 2010 [55] are mandatory legislations that need to be complied with when conducting building work in England. The Approved Documents [35] guide practitioners to meet these statutory



**Table 2**  
Details of the selected regulatory documents.

No.	Document Name	Constraint Sequence	Constraint Level	Main Scope
1	Building Act 1984 [54]	1: Legislation	Regulation	Process
2	Building Regulations 2010 [55]	2: Regulation	Regulation	Spatial, Physical, Process
3	Approved Document M Volume 2 [35]	3: Guidance	Regulation	Spatial
4	HBN 00-02 [10]	8: Requirement	Requirement	Managerial, Spatial
5	HBN 04-01 supplement 1 [11]	8: Requirement	Requirement	Physical, Spatial
6	BREEAM UK New Construction 2018 [6]	9: Specification	Recommendation	Managerial, Physical

rules. The Approved Document M Volume 2 [35] is for buildings other than dwellings. The Health Building Notes (HBNs) [9] are guidance for healthcare facilities in England and Wales. The BREEAM UK new construction 2018 [6] includes a set of criteria for assessing the sustainability of buildings, including healthcare buildings. These documents are all shortlisted to be further analysed. Next, the authors skimmed through their titles and introductions to identify their main scope and themes. After this step, six documents were selected as the sample for analysis, as shown in Table 2.

### 3.2. Data analysis

The authors developed a four-step analysis method to identify required capabilities for healthcare rule representation. As rules are the main subject to be captured and represented, this approach predominantly analyses rules. The first three analysis steps aim to understand rules from a consolidated list of aspects, including 1) rule features, 2) rule organisation, and 3) rule intensity. Step 4 focuses on implementation aspects because ACC system should also be equipped with implementation capabilities, which put forward requirements for the representation method. Previous studies have observed implementation capabilities [29,43], such as the transparency and user-friendliness of the visual programming language VCCL [44]. These four steps were detailed in Sections 3.2.1–3.2.4.

#### 3.2.1. Rule features

Previous studies have identified many characteristics of rules (e.g., dependency, conditionality) in different domains, yet they have not been thorough and generic. The need to identify general features of rules stems from the idea that the representation method needs to be capable of capturing all constructs of rules. Capturing all rule constructs is the baseline to “reproduce” the rules using a different representation and ensure minimum knowledge loss during rule interpretation, which is essential for the credibility and accuracy of ACC. Analysing rule features is the abstraction of the semantic meaning of and relationships among rule constructs. Such abstraction aims to address several questions:

1) How many semantic constructs are there in the rule, and what are they?

For example, is the phrase or word indicating what is to be checked? Does it denote what will happen if the rule is satisfied?

2) Are the semantic constructs isolated, or do they have interrelationships?

For example, does one word or phrase act as another's attributive, adverbial, or standalone? Is the scope of concept to be checked affected by other quantifiers?

#### 3.2.2. Rule organisation

Rule organisation involves the order and interdependencies among rule provisions and the hierarchies among different regulatory documents. It is necessary to understand rule organisation for several reasons. Firstly, the traditional way of categorising rules merely considers the theme of rules but neglects their contents [57]. Rule experts subjectively and arbitrarily decide the order of rule provisions. Consequently, it is difficult to collect all rules required for a specific concept, and some rules may be omitted when designing new construction. A better rule representation method should be able to address this issue. Secondly, an ACC system often includes many regulatory documents. These regulatory documents have different constraints (as shown in Fig. 3 and Table 2). Some contents in different documents are even contradictory (e.g., HBN and Approved Document M). The ACC system must address this issue to make the checking result credible. Thirdly, rule provisions frequently have interdependencies with other rules. These rules can only be correctly represented if the interdependencies are captured accurately. For these three reasons, it is crucial to rethink the capabilities required for rule organisation in the representation method to elicit interdependencies and relationships among rule provisions and hierarchies among regulations.

#### 3.2.3. Rule intensity

This paper also considered the rule intensity aspect. The term “rule intensity” is distinct from the “rule complexity” [49]. While rule complexity can result from the subjectivity, ambiguity, or referential relationships among rules, rule intensity refers mainly to the rule engine's intensity of operations used to execute rules. For example, “a touchdown base should be recessed sufficiently from any circulation routes so that a staff member, standing or perching on a stool, does not cause an obstruction”. This rule is very complex because the needed gap behind the touchdown base is not precisely given and requires further reference to the size of the human body. However, once these are specified, checking the dimension of the open space does not require very intense computer operations, so the rule intensity is low.

Previous research has dealt chiefly with the literal language representation of rules, such as syntax and grammar, while the hidden assumption and embedded knowledge in rules have been rarely accounted for [49]. Ideally, regardless of the intensity of rules, the embedded assumption and knowledge can be revealed by domain experts during rule representation. However, in the current research and practices, only a specific domain or small portion of rules with relatively low rule intensity have been considered, while more intensive rules remain unsolved. It may result in underestimating rule intensity [49] and may sacrifice the completeness of the number of rules the ACC system can check. On the other hand, when checking rules with high intensity, the advantages of ACC (e.g., saving time and improving accuracy) are more pronounced, as computers are more reliable and efficient at the repetitive execution of detailed algorithms. Thus, highly intensive rules must also be considered when developing rule interpretation and representation methods.

In addition, even if rules with higher intensity are considered during interpretation, scholars and practitioners tend to decide which rules need manual checking without sufficient evidence arbitrarily. The authors' experience in developing CORENET and RASE shows that what is deemed “uncheckable” by the ACC system frequently turns out to be checkable when analysed from the regulatory perspective. For example, Solihin and Eastman [50] found rules difficult by not giving sufficient weight to the original text and instead focusing primarily on the target representation. In one example relating to visibility (“All patient rooms shall be visible from the nurse station”), the grammar of the definite article “the” implies an existing relationship between the rooms and their station. In another case relating to contamination (“The discharge pipe shall not be located in places where...”), the context of the word

“places” suggests it is a space that fails. In both cases, carefully reading the text greatly simplifies the parsing and execution. Hence, whether specific rules are suitable for automatic checking must be thoroughly considered. The authors argue that this cannot be done by just identifying rule features; the intensity of each rule also needs to be understood. They are analysed in Section 4.1.4.

3.2.4. Implementation capabilities

Besides rule-oriented capabilities, ACC systems, as a computer-aided tool, typically put forward requirements for system efficiency, user-friendliness and transparency [45]. A typical ACC process involves manual interpretation and representation by domain experts. The representation is to be understood by people who consume the rules, and when needed, software engineers translate it into computer codes (see Fig. 1). Hence, the efficiency and quality of interpretation, representation and programming rely on these actors' productivity and quality of work. For the users of the ACC system, characteristics such as readability and complexity of the rule representation may greatly influence productivity and quality of manual work. Thus, these characteristics need to be carefully considered when developing ACC systems.

Capabilities related to system maintenance were also identified in this paper. Previous work showed a general lack of attention to the modification of regulations and its influence on the rule interpretation, representation and ACC system development. Two important facts have been frequently overlooked: 1) rule interpretation and representation processes are iterative. The initial representations may be changed multiple times as experts have different opinions [18]. 2) Rules are updated every few years. Fig. 4 presents the regulation improvement, design improvement and rule implementation cycles. The design submission and compliance checking processes generate feedback for the regulation development and improvement. Rule representations need to be updated and maintained accordingly when the regulations are re-introduced. Thus, considering capabilities regarding system maintenance could prepare the system for future updates and changes.

4. Analysing required and desired capabilities for representing healthcare regulations

4.1. From the rule perspective

4.1.1. Scale of rules

An individual rule (the phrase or the sentence predicate) can create a specific requirement. However, previous research often ignored the broader context to understand the rule's effect. The broader context may include phrases that control the applicability and selection, identifying the sentence subject. The subject may be qualified further by exceptions that may be found in the adjacent sentences or paragraphs and are sometimes not explicitly linked. Preceding headings and clauses may limit the subject matter. At the start of the document (or even in the first volume of a series), the beginning matter may contain overall subject statements and definitions (such as the classification of applicable building types). The regulations' applicability in terms of regions and commencement dates may be defined in the document title or enabling legislation.

To illustrate, Table 3 was presented with a typical example of predicates (the first row) and possible broader context (row 2–7) to

Table 3

Scale of rules.

Predicate and subject	Typical Examples
Predicate phrases alone	Escape distance to be less than 35 m
With subject phrases	Space occupied by more than 60 people
With references	As defined in volume 1
With immediate exceptions	Unless alternative means are available
With preceding headings and clauses	Classrooms and lecture theatres
With introductory matter and definitions	Planning Use class F1
With title, metadata and parent documentation	Fire guidance In England and Wales, Enacted 1st Nov 2010

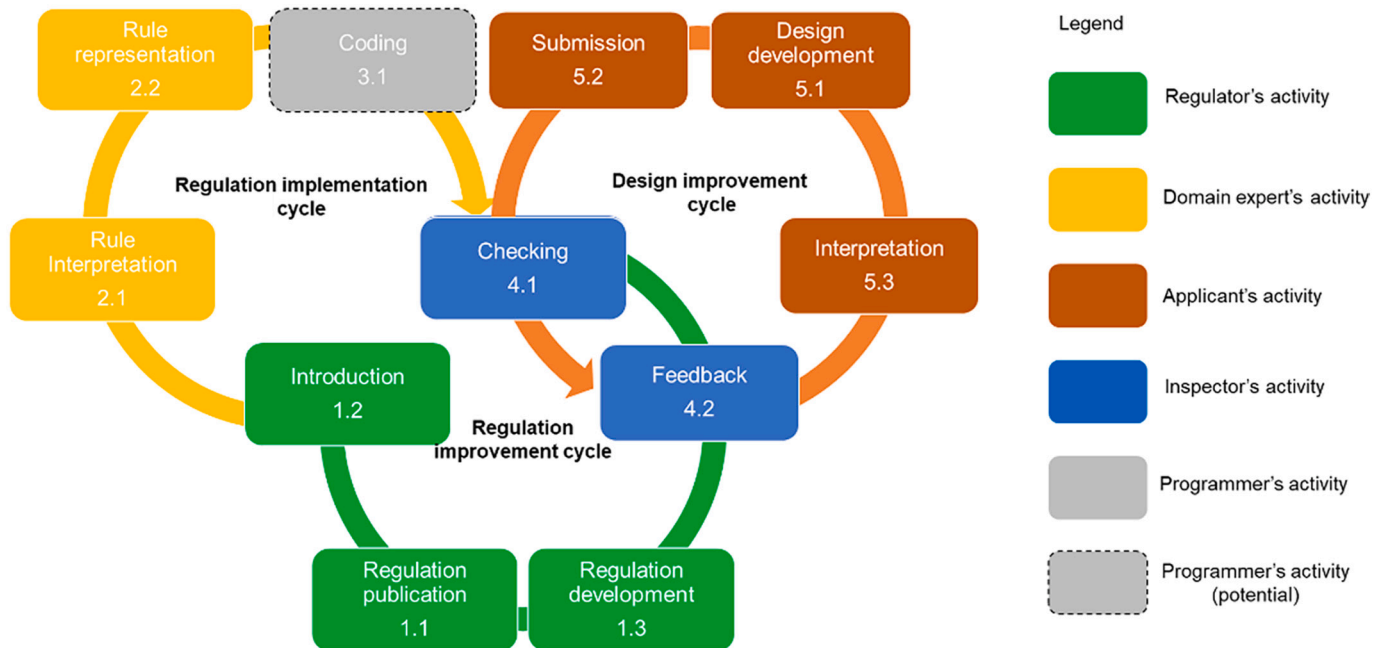


Fig. 4. Design improvement, regulation improvement and regulation implementation cycles.

explain the issue. It highlights that the semantics of a rule depends on the choice of scale.

It can be seen in Table 3 that the predicate phrase (i.e., requirement) is about the minimum escape distance. However, it is insufficient to focus only on the requirement for a “minimum escape distance” since it may need to be qualified at each of the subsequent scales. The sequence of checking also matters. It may be necessary to check the “space occupancy” either before or after checking the requirement. Some of the broader scale criteria are also very important; they may be equally intrinsically intense (Section 4.1.4) or extrinsically complex (Section 4.1.3) as the predicate phrase alone. In Table 3, the consideration of the “planning use classes” may depend on building access arrangements and other exceptions. Parent documents such as primary legislation may determine the relevance of the source document by place and time. Ignoring these broader scales may generate false negatives as fails or false positives as passes.

#### 4.1.2. Rule features

Rules and provisions are the basic elements of regulations. The wide variety of provisions and logical connectives pose significant challenges to computerising healthcare regulations. The rule constructs, as the most basic parts in rule sentences, were analysed first in this paper to capture rule knowledge. The representation method must be able to represent these constructs.

After reviewing a number of regulations as mentioned in Table 2, the authors found the following distinctions:

- 1) Expectations
  - a) requirement (compulsory)
  - b) definition (optional)
- 2) Conditionality
  - a) exception (optional)
  - b) applicability (optional)
  - c) selection (optional)
- 3) Outcomes
  - a) pass/ fail/ unknown (compulsory)
  - b) other actions and side-effects (optional)
- 4) Logical relationships
  - a) or/ and/ not (optional)

Each rule must have at least one of all compulsory components, plus any type or number of the optional components. These constructs are explained in Sections 4.1.2.1–4.1.2.4.

**4.1.2.1. Expectations.** Each rule has requirements stating what specifically needs to be checked. Usually, requirements are signified by future imperatives such as “should”, “shall”, or “must” [23]. At the core of requirements is a topic such as either attribute of entities or relationships among entities to be checked. Depending on the specific rule, fixed values (arithmetic or descriptive), value ranges (arithmetic), sets of enumerated values or relationships need to be checked. The attributes or relationships to be checked may be as simple as properties of the entities or simple calculations using several property values (e.g., height, coordinates). The example shows the dimension requirements for the dispensers mounted on the back of the hinged door. The requirements are in the form of value ranges (i.e., project no more than 50 mm) of the “depth” property of the dispensers.

**4.54** *“The dispensers mounted on the back of the hinged door should not project more than approximately 50 mm (depending on the door design) to ensure they do not conflict with the use of the hinged grabrail between the door and the toilet pan.”*

—HBN 00-02

#### Example Rule 1

In some cases, the checking to be performed is not reflected by any property directly available in the target data model, and high-level

methods such as algorithms and/or simulations are required. Energy performance rules in BREEAM, for example, fall into this category. This category of rules requires the rule engine to be equipped with algorithms and simulation capabilities. This will be further discussed in Section 4.1.3.

In regulations, terms are frequently used. To avoid ambiguity and misunderstanding, there are often definitions to describe and explain certain terms used in rules. For example, the following definition defines the term fire safety information:

**Regulation 38 (3)** *“In this regulation—*

*(a) “fire safety information” means information relating to the design and construction of the building or extension, and the services, fittings and equipment provided in or in connection with the building or extension which will assist the responsible person to operate and maintain the building or extension with reasonable safety;” —Part 8, The Building Regulations 2010 (No.2214)*

#### Example Rule 2

As shown in this example, definitions typically do not put forward any sorts of new requirements. Their sole purpose of existence is to define terms.

**4.1.2.2. Conditionality.** There may be some constraints or conditions to narrow down the applicability of rules, such as specifying the applicable area (e.g., patient room), intended use (e.g., for independent wheelchair use), occupancy type (e.g., more than ten occupants), or building type (e.g., hospital). During the ACC process, such applicability information may not always be available in the building information models. Thus, human input may be required to supplement relevant information via a user interface. In most cases, this information is used to decide whether or not the rule needs to be checked or which branch this check should go down. However, in rare cases, the conditions are rather complex. For example, the specific rule may depend on the risk assessment (Example Rule 3). Only experts are able to address these rules appropriately and confirm that a manual compliance checking method may be suitable in these cases.

**4.68** *“BS 8300 recommends a hinged grabrail at right-angles in front of shower seats for independent wheelchair transfer. This grabrail is to help prevent users falling forward. This rail is not considered necessary in healthcare premises. However, a risk assessment is recommended to confirm requirements.”*

—HBN 00-02

#### Example Rule 3

A similar but distinct construct to applicability is a selection [23]. Selections also concern the scope of the requirement, but instead of one single applicability, it offers a selection of alternative subjects, for example, “pedestrians or wheelchair users”.

Despite specifying the applicability of rules, some rules may have further exceptions. Exceptions can be written 1) in a separate sentence specifying the condition of exception; or, 2) within the same sentence of the rule, providing negation of the rest of the sentence or part of it. The following rules are examples of the first and second cases, respectively. In the first example, the height requirements can differ for switches, outlets, and controls set into the floor in open plan offices. While in the second example, the rule means the pressure in the room is lower than the corridor and other adjacent areas, but it is not necessarily lower than the pressure in its en-suite.

**4.25** *“The key factors that affect the use of switches, outlets and controls are ease of operation, visibility, height and freedom from obstruction. However, there will be exceptions to height requirements for some outlets, e.g. those set into the floor in open plan offices”*

—Approved Document M Volume 2

#### Example Rule 4



**2.11** “This room is at negative pressure to the corridor and other adjacent areas (except for its en-suite).”

—HBN 04-01 Supplement 1

#### Example Rule 5

**4.1.2.3. Outcome.** Rules may not always specify the outcome when requirements are met; this typically means “pass” by default. For example, the following managerial rule means: if the position of the shower seat is adjusted between different patients' uses, the rule is passed. By contrast, when the requirement in the rule is not satisfied, it will “fail”. However, the outcome could also be “unknown” when there is a lack of information to make the judgement. Example Rule 6 is expressed as an operational requirement, but its implication is that the seat should be specified as “adjustable” in the design. If this property is not given explicitly, the outcome of the rule-checking should be flagged as “unknown”.

**4.48** “...The position of the shower seat should be adjusted between uses as required.”

—HBN 00-02

#### Example Rule 6

Ideally, the ACC system should provide an interface for users to supplement information. When it is impossible to do so, the outcome depends on the views taken on unknown data. The closed-world assumption only sees something to be true when it is known to be true, while the open-world assumption indicates that unknown or implicit knowledge can also be true [24]. In the case of unknown information in ACC, many scholars adopted an open-world assumption (e.g., [13]). They believe that when the information is missing, the result is “unknown”. This paper also adopts an open-world assumption because the closed world assumption can result in many false negatives [58] (i.e., when there is unknown information, the check result is “fail”, but it may be “pass” if given enough information).

Apart from “pass” or “fail” or “unknown”, it is possible to have outcomes such as “warning”. Warnings may be triggered when the design does not satisfy the suggested requirements from guidance and recommendations. Although failing to meet these requirements does not result in non-compliance, reporting warnings provides further opportunities for design improvement.

In addition, with the increasing complexity of rules, there are some rules with other outcomes or side-effects. For example, BREEAM takes a balanced scorecard approach for sustainability performance assessment by awarding credits (points/scores) [7]. If the minimum requirements are met, the percentage of ‘credits’ earned in each section is then multiplied by the corresponding section weighting [7]. The final rating (i.e., outstanding, excellent, very good, good, pass, unclassified) reflects the cumulative score of each section. Accordingly, to accurately reproduce such BREEAM rules, the capability to represent outcomes other than “pass”, “fail”, or “unknown” is essential.

**4.1.2.4. Logical relationships.** Many studies have recognised that the logical relationships among rule constructs can affect the meaning of the rule provision (e.g., [23,53]). Hence, knowledge in rule provisions can only be accurately captured when the logical relationships are correctly identified. In rule sentences, logical relationships are typically affected by logical connectives, including “and”, “or” and “not”, meaning logical conjunction, disjunction and negation, respectively. These words can appear in rules either individually or jointly with the risk of ambiguity. Example Rule 7 is an example of using a single logical connective “and”, indicating that clauses (a) and (b) must be both satisfied to enable the public body to dispense or relax the requirement.

**Section 8 Relaxation of building regulations** “(4) If— (a)building regulations so provide as regards any requirement contained in the

regulations, and. (b)a public body considers that the operation of any such requirement would be unreasonable in relation to any particular work carried out or proposed to be carried out by or on behalf of the public body, the public body may give a direction dispensing with or relaxing that requirement.”

—Part I, c. 55, Building Act 1984

#### Example Rule 7

While the above example only used a single logical connective, actual rule provisions may include multiple logical connectives. The representation needs to be capable of presenting all logical connectives in rule provisions to replicate their meanings.

#### 4.1.3. Rule organisation

Rule organisation mainly include two aspects. The first aspect of rule organisation concerns the overall organisation—the framework of and the relationships among different statutory and guidance documents. Frequently, such organisation is expressed using a hierarchical approach to denote which document takes precedence, where documents with higher precedence have a higher constraint level. For example, Fig. 5 shows the hierarchies of regulatory documents for healthcare facilities in England. They are characterised by a complex and potentially confusing mix of statutory and guidance documents [52], including UK Public General Acts, statutory documents (e.g., CQC regulations) and best practice guidance. Thus, being able to capture superiority and inferiority is crucial for rule representation and ACC, as such information plays a key role when the contents of different documents are different or contradictory. In this sense, an equally important thing is whether a document is mandatory or suggestive. It helps with the decision of checking outcomes.

Another aspect of rule organisation is regarding the referential relationships among rule provisions, either in the same document or across different documents. For example, Example Rule 8 shows cross-references between paragraph (2) (a), (2) (b) and 16 (4) in Part 9 of the Building Regulations 2010. The requirements can only be correctly understood by looking at all these provisions. Thus, the representation of rules should present the referential relationships among rules clearly.

#### Sound insulation testing

**41. (3)** “The results of the testing referred to in paragraph (2)(a) shall be—

(a)recorded in a manner approved by the Secretary of State; and.

(b)given to the local authority in accordance with paragraph (2)(b) not later than the date on which the notice required by regulation 16(4) is given.”

—Part 9, The Building Regulations 2010 (No. 2214)

#### Example Rule 8

#### 4.1.4. Rule intensity

Rule intensity concerns the intensity related to the operations of executing rules. This may depend on the nature of the target building representation which may affect the nature of the algorithm needed to enrich or report results. For example, annual energy consumption may depend on an approved calculation, or it may be found explicitly in a model representing an asset in use. The authors analysed the requirements and entities to be checked and found three different capabilities are needed for representing various rule intensities, which are explained in Sections 4.1.4.1–4.1.4.3.

**4.1.4.1. No calculations or simple calculations.** In the simplest rules, the attributes of entities or relationships among entities are expected to be available in the building data model or user supplements. These rules may have fixed values as requirements for the design to comply with. For example, in Example Rule 9, the fixed property to be checked is “inward opening” of the door.

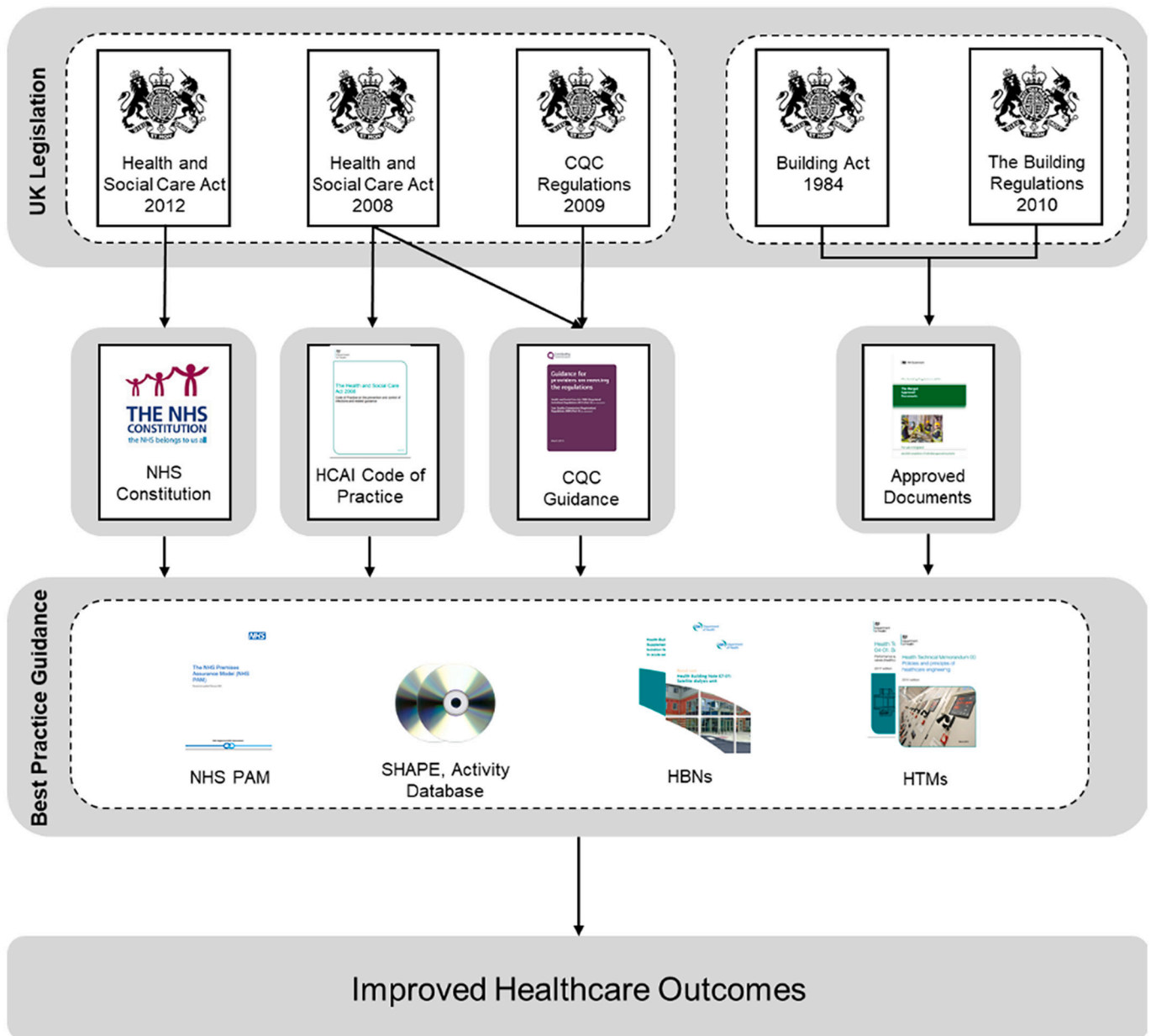


Fig. 5. Hierarchies of Healthcare Regulations in England (Adapted from [9,52]).

2.21 “.....The door between the corridor and the lobby should open into the lobby.....”

–HBN 04-01 Supplement 1

Example Rule 9

Sometimes the requirements are not fixed values. Some simple calculations of attribute value or comparison may be required. Such simple calculations are mostly arithmetic calculations. They typically do not involve calculations of complex spatial relationships or physical issues. For example, to check Example Rule 10, the rule engine compares the mounting height of the control with 1200 mm and 1400 mm.

4.30 “f. Controls that need close vision are located between 1200mm and 1400mm above the floor so that readings may be taken by a person sitting or standing (with thermostats at the top of the range);”

–Approved Document M Volume 2

Example Rule 10

4.1.4.2. *Functions or algorithms.* Rules that require functions or algorithms to be checked are typically spatially, geometrically or topologically complex. This type of rule often involves combinatorial issues that deal with multiple objects and possibilities to compliance [49]. An example rule falling into this category is accessibility rules that check the closest obstruction of an object. Such rules typically require the distance between the object and the nearest obstruction to be greater than a particular value, thus making sure that the specific space has acceptable accessibility. As the distances between the object and obstructions are determined by their locations, there is no fixed formula to calculate the distance. In this case, algorithms or functions are needed. For example, Fig. 6 demonstrates a checking that can automatically draw a circle (with the required distance as its radius) and detect whether any obstruction intersects this circle. In the example, since the WC intersects the circle, the rule is failed. If no obstruction intersects the circle, the rule is passed.

4.1.4.3. *Simulations.* The requirements for simulations in ACC often

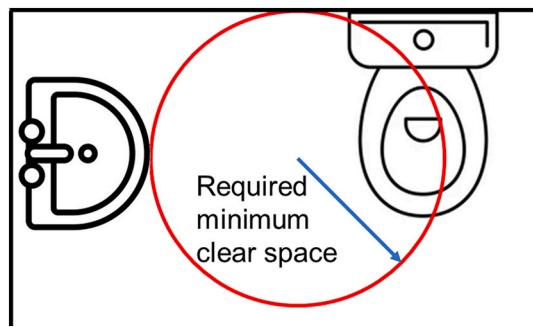


Fig. 6. Demonstration of the nearest obstruction rule

appear in regulations of building performance, such as energy consumption rules, fire codes, etc. This type of rule does not provide prescriptive requirements for compliance but ask designers to provide a feasible solution. As such, there are multiple possible ways to compliance. The checking is more focused on whether the proposed design meets the expectations, not the specific route to compliance. These rules are often referred to as “performance-based rules” [1,49] as opposed to “prescriptive rules”. These may be expressed as formulae or calculation methods. Due to high rule intensity and specialised requirements, some of these types of rules may not be calculated directly using the rule engine. Typically, domain-specific software is used to run simulations to test the proposed design. Then the results are reported to the rule engine or the building data model to generate results. For example, in Example Rule 11, approved building energy and performance modelling software is used to assist the checking. The representation needs to clearly present where the simulation is required and what results are required.

#### One credit - Thermal modelling

“1. Thermal modelling has been carried out using software in accordance with CIBSE AM11(79) Building Energy and Performance Modelling,.....  
3. The modelling demonstrates that: 3.a For air-conditioned buildings, summer and winter operative temperature ranges in occupied spaces are in accordance with the criteria set out in CIBSE Guide A Environmental design (80), Table 1.5; or other appropriate industry standard (where this sets a higher or more appropriate requirement or level for the building type); or the thermal environment in occupied spaces meet the Category B requirements for PPD, PMV and local discomfort set out in Table A.1 of Annex A of ISO7730:2005.”

–Hea 04 Thermal comfort, BREEAM UK New Construction 2018

Example Rule 11

#### 4.2. From the implementation perspective

ACC systems are developed to provide high-quality, timesaving and cost-saving compliance checking for building design. To this end, several capabilities are required, among which some are also reflected in the rule representation method, such as capabilities related to user experience, system efficiency and rule expressiveness aspects.

As an output of the interpretation process, rule representation is a machine-readable form of rules created by domain experts. Such representation often takes the form of logic-based expression, semantic web queries, domain-specific codes, etc. Domain experts typically do not know how to use these expressions or write queries. These representations are learned during interpretation. Therefore, whether a representation method is easy to learn and use by users becomes one of the main factors affecting interpretation efficiency. Previous work has shown that the steep learning curve of past methods has been a reason why they have not been widely accepted, such as semantic web [42] and hard codes [48]. As such, a suitable representation method should have the capability of being easy to use and learn by users.

In order to reduce knowledge loss, representations should be able to present all the information contained in the rules. Some existing representations have used IFC entities, properties and relationships to express rules. However, as IFC is not sufficient to represent all concepts found in rules [43], this may lead to missing information. In fact, representations not only need to be independent of IFC, but they should also be independent of building data in general. Broadly speaking, the design used for ACC inspection may be in any form, such as paper-based drawings, BIM models, 3D CAD etc. Some of these representations may be checked manually using checklists generated from the analysis of the regulations, some may be made accessible by artificial intelligence (for example, information embed in paper-based drawings can be extracted by computer-vision algorithms), others may be directly susceptible to automated checking. It would limit the expressiveness and versatility of representation if assuming a particular type of building data would be used. Therefore, the representation should be equivalent to original rule texts without any limitations by building data format.

While the expressiveness can be enhanced using representations independent of the building data model, it would make the checking process more problematic as it is hard to match the target data in the model with the rule to be checked. To address this issue, Nisbet [40] used a dictionary to assist the mapping between rule objects and IFC objects. It can also be used to store functions and definitions to ensure their consistency and reusability [40]. It is especially helpful because similar terms may be used for the same entity, such as “adjustable washbasin”, “adjustable-height wash hand basin”, and “adjustable wash hand basin”. These terms may bring confusion to the process of translating rule representation into computer codes. Therefore, a dictionary is essential to maintain the unity of terms.

In addition, regulations and target representations are frequently updated. Representation needs to be independent of the rule engine to ensure that it is easy to maintain by domain experts [33]. In this way, domain experts only need to interpret the rules into the representation without dealing with complex computer programming. To make the representation adaptable to future modification, it is also desirable that the representation has translatability. Translatability refers to the ability to translate a certain representation into another representation. An example of this is the translatability between conceptual graphs and predicate logic [50]. This ability improves the efficiency when the existing representation is not suitable for the modified rules or if the existing representation is being updated because a more suitable representation has been found. It avoids re-creating representations from scratch.

In addition, it is also desirable for a representation to be concise. Conciseness can be reflected in two aspects, namely, the length of representation and the level of redundancy of representation. As for the length of the representation, ideally, the shortest possible expression represents the rule without losing any information. Such expressions may not always be possible, but they are crucial to avoid lengthy representation. The level of redundancy deals with the repetitiveness of rules. Previous studies such as Tan et al. [53] and Lee [30] presented table-based representations to reduce repetition. The main idea is that some rules (e.g., envelope design rules) have a set of similar headings such as “location” and “primary heating source”. The only differences are the required values. A desirable representation could avoid showing these parameters many times, thus enhancing the efficiency of interpretation.

#### 4.3. Summary of capabilities for healthcare building rule representation

In summary, 18 capabilities for a representation method were identified (Table 4), among which 16 are required while two are desired. These required capabilities mean that the representation method should be able to: 1) represent all seven rule features; 2) capture all three rule intensity levels and represent them explicitly; 3) show the hierarchy and cross-references among regulations and rules; and 4) incorporate four

**Table 4**  
Required and desired capabilities identified in this paper.

No.	Aspects	Required capabilities	Desired capabilities
1	Rule features	①Requirement ②Applicability ③Selection ④Exception ⑤Definition ⑥Outcome ⑦Logical relationship	NA
2	Rule intensity	①No calculation or simple calculation ②Function and algorithm ③Simulation	NA
3	Rule organisation	①Hierarchy ②Cross-reference	NA
4	Implementation	①Easy to use and understand ②Dictionary ③Independent of the rule engine ④Independent of the data model	①Conciseness ②Translatability

implementation qualities. Conciseness and translatability are nice-to-have capabilities but not essential.

## 5. Discussion

Previous research has proposed many rule representation methods for ACC. However, none of them has been developed based on a thorough analysis of the rules to be represented; no studies have explicitly summarised the required and desirable capabilities for a rule representation method. As the first step toward a well-rounded representation for building regulations, this research distinguishes from previous studies and make contributions in the following aspects:

### 5.1. Analysing a wide scope and multiple constraint levels of building regulations

Existing representation methods have been criticised for only focussing on the representation of certain domains or types of rules [33,49]. This can be attributed to scholars typically selecting one or several documents “randomly”, normally without paying special attention to the breadth of scope or constraint level. To alleviate this issue, in this research, the authors systematically explored regulations related to healthcare facilities in the UK, including their scope, constraint level, constraint sequence, and hierarchies. The selected samples cover all three constraint levels (i.e., regulation, requirement, recommendation) and four main scopes (i.e., managerial rules, process rules, spatial rules, physical rules) as identified by the authors to improve data representativeness. It also includes healthcare-specific rules and general regulations that apply to healthcare rules. The enrichment of datasets provides a larger sample for analysis and generates more reliable and generalised results.

### 5.2. Providing a clear definition of rule scale

The scale of the rule representation is defined by choosing the bounding limitations. Any representation limitations that create a boundary should be explicitly stated, even if obvious or implicit, as in the use of conceptual graphs in [50] or the rule level hierarchy in [33]. Ideally, a rule representation needs to be capable of representing the scale independent of how the narrowly or broadly boundary is drawn.

### 5.3. Understanding healthcare building rules from a consolidated list of aspects

The current representation development has not been based on a thorough understanding of rules. It can be reflected by the rule classifications lacking explicit criteria [38] and considering only a few aspects such as semantics [51] and complexity [49]. The rule features, rule intensity and rule organisation aspects considered in this paper are more

comprehensive to understand rules and analyse the required capabilities. It can address lower-level rule provisions, higher-level relationships among provisions and hierarchies of regulation documents. Using the three aspects, both the explicit rule constructs and implicit knowledge embedded in rules can be revealed. Such an understanding help to depict a whole picture of required and desired capabilities, thereby helping future representation development. The findings discovered 12 rule-oriented required capabilities, many of which have been neglected in previous studies, such as hierarchy and definition.

### 5.4. Considerations on implementation aspects

Previous ACC studies have considered limited implementation aspects, mostly restricted to user-friendliness or reliability [43]. Consequently, many other implementation aspects have been rarely accounted for. In this paper, the authors considered multiple implementation aspects, including the efficiency of using the system, user-friendliness, consistency and reusability of terms, and mapping between rules and data model. A total of six capabilities were identified related to implementation, among which four are required. These capabilities could contribute to developing a more all-around representation method and the whole ACC system.

Notably, this research recognised the capabilities to prepare the system for future modification. Previously, most ACC systems have been developed for proof-of-concept, where scholars typically test the system once and stop maintaining the system thereafter. As a result, some potential problems in the practical implementation stage have been ignored, such as the difficulty and resources required for maintaining and modifying rules. In this paper, the authors addressed this issue by considering the difficulty for domain experts to re-interpret rules and the possibility of directly translating one representation into another. Considerations on maintenance could help the representation method to be practically implemented and stand the test of time.

## 6. Conclusion

This paper presented the required and desired capabilities for healthcare rule representation. Being equipped with these capabilities, such a representation is envisaged to represent virtually all kinds of healthcare building rules and enable efficient interpretation and maintenance simultaneously. To the best of the authors' knowledge, this work is the first attempt to explore the capabilities needed for rule representation in the healthcare building domain. It provides a fresh perspective and a solid foundation for developing a new representation method suitable for healthcare-building rules.

A large data sample has been collected and analysed in this paper. It includes six England healthcare regulatory documents covering four scopes and three constraint levels. This dataset provides a better generalisation of healthcare building rules than previous studies only using simple rules or rules of constrained scope. The novel 4-step analysis method helped to analyse building rules in both rule-oriented and implementation aspects systematically. As a result, 16 capabilities were identified as essential for the efficient use and maintenance of a representation method, whilst two capabilities were desirable. The proposed capabilities could serve as a checklist when developing or modifying rule representation methods. It also provides a set of criteria for assessing different rule representation methods to help make improvements to automated code compliance for the built environment.

This study also has some limitations. Although this research strives to systematically consider the required capabilities for healthcare rule representation, some capabilities may still be omitted because this process heavily relies on inductive reasoning. Future research could interview some domain experts to refine this list. In addition, although the authors reviewed many healthcare regulations, other regulations not included in the sample may still have other features that require additional capabilities, which could be considered in future work.



## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

No data was used for the research described in the article.

## References

- R. Amor, J. Dimiyadi, The promise of automated compliance checking, *Develop. Built Environ.* 5 (2021), 100039, <https://doi.org/10.1016/j.dibe.2020.100039>.
- T.H. Beach, T. Kasim, H. Li, N. Nisbet, Y. Rezgui, *Towards Automated Compliance Checking in the Construction Industry*, Springer, 2013, pp. 366–380, [https://doi.org/10.1007/978-3-642-40285-2\\_32](https://doi.org/10.1007/978-3-642-40285-2_32).
- T.H. Beach, Y. Rezgui, H. Li, T. Kasim, A rule-based semantic approach for automated regulatory compliance in the construction sector, *Expert Syst. Appl.* 42 (12) (2015) 5219–5231, <https://doi.org/10.1016/j.eswa.2015.02.029>.
- K.R. Bouzidi, B. Fies, C. Faron-Zucker, N. Le Than, O. Corby, Towards a semantic-based approach for modeling regulatory documents in building industry, in: *European Conference on Product and Process Modelling, ECPPM 2012*, CRC Press, Reykjavik, 2012, pp. 347–353, <https://doi.org/10.1201/b12516-55>.
- bSI Regulatory Room Working Group, *Open Standards for Regulations, Requirements and Recommendations Content*, <https://www.buildingsmart.org/wp-content/uploads/2017/11/17-11-08-Open-Standards-for-Regulation.pdf>, 2017 (accessed May 3, 2022).
- Building Research Establishment, *BREEAM New Construction 2018 (UK)*, <https://files.bregroup.com/breeam/technicalmanuals/NC2018/>, 2018 (accessed May 3, 2022).
- Building Research Establishment, *Building Research Establishment Environmental Assessment Method*, <https://www.breeam.com/>, 2021 (accessed May 3, 2022).
- J.W. Creswell, J.D. Creswell, *Research design: Qualitative, Quantitative, and Mixed Methods Approaches*, Sage Publications, 2017 (1506386717).
- Department of Health and Social Care, in: D.o.H.a.S. Care (Ed.), *DH Health Building Notes, 2017*, <https://www.england.nhs.uk/estates/health-building-notes/> (accessed May 3, 2022).
- Department of Health and Social Care, in: D.o.H.a.S. Care (Ed.), *Health Building Note 00–02: Sanitary Spaces, 2016*, [https://www.england.nhs.uk/wp-content/uploads/2021/05/DH\\_HBN\\_0002.pdf](https://www.england.nhs.uk/wp-content/uploads/2021/05/DH_HBN_0002.pdf) (accessed May 3, 2022).
- Department of Health and Social Care, in: D.o.H.a.S. Care (Ed.), *Health Building Note 04–01 Supplement 1: Isolation Facilities for Infectious Patients in Acute Settings, 2013*, <https://hberm.com/wp-content/uploads/2015/10/HBN-04-01-Supplement-1-Isolation-facilities-for-infectious-patients-in-acute-settings-20131.pdf> (accessed May 3, 2022).
- J. Dimiyadi, P. Pauwels, R. Amor, Modelling and accessing regulatory knowledge for computer-assisted compliance audit, *J. Inform. Technol. Construct.* 21 (2016) 317–336, [https://www.itcon.org/papers/2016\\_21.content.02683.pdf](https://www.itcon.org/papers/2016_21.content.02683.pdf) (accessed May 3, 2022).
- C. Eastman, J.-M. Lee, Y.-S. Jeong, J.-K. Lee, Automatic rule-based checking of building designs, *Autom. Constr.* 18 (8) (2009) 1011–1033, <https://doi.org/10.1016/j.autcon.2009.07.002>.
- E. Enache-Pommer, M.J. Horman, J.I. Messner, D. Riley, A unified process approach to healthcare project delivery: synergies between greening strategies, lean principles, and BIM, *Construct. Res. Congr.* 2010 (2010) 1376–1405, [https://doi.org/10.1061/41109\(373\)138](https://doi.org/10.1061/41109(373)138).
- S.J. Fenves, Tabular decision logic for structural design, *J. Struct. Div.* 92 (6) (1966) 473–490, <https://doi.org/10.1061/JSDEAG.0001567>.
- S.J. Fenves, E.H. Gaylord, S.K. Goel, Decision Table Formulation of the 1969 AISC Specification, University of Illinois Engineering Experiment Station, College of Engineering, University of Illinois at Urbana-Champaign, 1969, <https://www.id.eals.illinois.edu/bitstream/handle/2142/14275/SRS-347.pdf?sequence=2&isAllowed=y> (accessed May 3, 2022).
- S.J. Fenves, R.N. Wright, F.I. Stahl, K.A. Reed, Introduction to Sase: Standards Analysis, Synthesis, and Expression, Editorial NBSIR (National Bureau of Standards: Washington, DC), <https://www.govinfo.gov/content/pkg/GOVPUB-C13-2ee11e8ce4de1e0fb96ae6e2375b7f03/pdf/GOVPUB-C13-2ee11e8ce4de1e0fb96ae6e2375b7f03.pdf>, 1987. (Accessed 3 May 2022).
- Fiatch, AutoCodes Project: Phase 1, Proof-of-Concept Final Report, <https://documents.pub/document/autocodes-project-phase-1-proof-of-concept-final-report-autocodes-project.html>, 2012 (accessed April, 21, 2022).
- J.H. Garrett Jr., M.M. Hakim, Object-oriented model of engineering design standards, *J. Comput. Civ. Eng.* 6 (3) (1992) 323–347, [https://doi.org/10.1061/\(ASCE\)0887-3801\(1992\)6:3\(323\)](https://doi.org/10.1061/(ASCE)0887-3801(1992)6:3(323)).
- C.S. Han, J.C. Kunz, K.H. Law, Client/server framework for on-line building code checking, *J. Comput. Civ. Eng.* 12 (4) (1998) 181–194, [https://doi.org/10.1061/\(asce\)0887-3801\(1998\)12:4\(181\)](https://doi.org/10.1061/(asce)0887-3801(1998)12:4(181)).
- S. Hignett, J. Lu, An investigation of the use of health building notes by UK healthcare building designers, *Appl. Ergon.* 40 (4) (2009) 608–616, <https://doi.org/10.1016/j.apergo.2008.04.018>.
- E. Hjelseth, N. Nisbet, Capturing Normative Constraints by use of the Semantic mark-up RASE Methodology, 2011, pp. 1–10, <https://itc.scix.net/pdfs/w78-2011-Paper-45.pdf> (accessed May 3, 2022).
- E. Hjelseth, N. Nisbet, Exploring Semantic Based Model Checking vol. 54, Citeseer, 2010, <https://www.irbnet.de/daten/iconda/CIB21890.pdf> (accessed May 3, 2022).
- U. Hustadt, Do we need the closed world assumption in knowledge representation?, in: *Knowledge Representation Meets Databases vol. 1*, 1994, <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.9005&rep=rep1&type=pdf> (accessed May 3, 2022).
- H. Kilicote, J.H. Garrett, T.J. Chmielenski, K.A. Reed, The Context-Oriented Model: An Improved Modeling Approach for Representing and Processing Design Standards, ASCE, 1994, pp. 145–152, <https://cedb.asce.org/CEDBsearch/record.jsp?dockey=0088475> (accessed May 3, 2022).
- H. Kim, J.-K. Lee, J. Shin, J. Choi, Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking, *J. Comput. Design Eng.* 6 (2) (2019) 143–148, <https://doi.org/10.1016/j.jcde.2018.08.002>.
- H. Lee, J.-K. Lee, S. Park, I. Kim, Translating building legislation into a computer-executable format for evaluating building permit requirements, *Autom. Constr.* 71 (2016) 49–61, <https://doi.org/10.1016/j.autcon.2016.04.008>.
- J.-K. Lee, C.M. Eastman, Y.C. Lee, Implementation of a BIM domain-specific language for the building environment rule and analysis, *J. Intell. Robot. Syst.* 79 (3–4) (2014) 507–522, <https://doi.org/10.1007/s10846-014-0117-7>.
- J.K. Lee, Building Environment Rule and Analysis (BERA) Language and its Application for Evaluating Building Circulation and Spatial Program, Georgia Institute of Technology, 2011, [https://smartech.gatech.edu/bitstream/handle/1853/39482/Lee\\_Jin-Kook\\_201105\\_PhD.pdf?sequence=1&isAllowed=y](https://smartech.gatech.edu/bitstream/handle/1853/39482/Lee_Jin-Kook_201105_PhD.pdf?sequence=1&isAllowed=y) (accessed May 3, 2022).
- J.M. Lee, Automated Checking of Building Requirements on Circulation over a Range of Design Phases, Georgia Institute of Technology, 2010, [https://smartech.gatech.edu/bitstream/handle/1853/34802/lee\\_jae\\_m\\_201008\\_phd.pdf?sequence=1&isAllowed=y](https://smartech.gatech.edu/bitstream/handle/1853/34802/lee_jae_m_201008_phd.pdf?sequence=1&isAllowed=y) (accessed May 3, 2022).
- L. Ma, R. Sacks, U. Kattel, T. Bloch, 3D object classification using geometric features and pairwise relationships, *Comp.-Aid. Civ. Infrastruct. Eng.* 33 (2) (2018) 152–164, <https://doi.org/10.1111/mice.12336>.
- L. Ma, R. Sacks, K. Uri, Building Model Object Classification for Semantic Enrichment Using Geometric Features and Pairwise Spatial Relationships 2017, 2017, LC3, pp. 373–380.
- S. Macit İlal, H.M. Günaydin, Computer representation of building codes for automated compliance checking, *Autom. Constr.* 82 (2017) 43–58, <https://doi.org/10.1016/j.autcon.2017.06.018>.
- S. Macit, Computer Representation of Building Codes for Automated Compliance Checking, Izmir Institute of Technology, 2014, <https://gcris.iyte.edu.tr/bitstream/11147/4296/1/T001336.pdf> (accessed May 3, 2022).
- C.L.G. Ministry of Housing, Approved Document M: Access to and Use of Buildings, Volume 2: Buildings Other than Dwellings, 2015 ed., 2015 (9781859466094).
- C.L.G. Ministry of Housing, Manual to the Building Regulations: A Code of Practice for Use in England, 2020 (9798710651216).
- National Audit Office, A Safer Place for Patients: Learning to Improve Patient Safety, The Stationery Office, London, 2005, <https://www.nao.org.uk/report/a-safer-place-for-patients-learning-to-improve-patient-safety/> (accessed May 3, 2022).
- A. Nawari, Generalized adaptive framework (GAF) for automating code compliance checking, *Buildings* 9 (4) (2019), <https://doi.org/10.3390/buildings9040086>.
- N.O. Nawari, Generalized adaptive framework for computerizing the building design review process, *J. Archit. Eng.* 26 (1) (2020), [https://doi.org/10.1061/\(asce\)jae.1943-5568.0000382](https://doi.org/10.1061/(asce)jae.1943-5568.0000382).
- N. Nisbet, J. Wix, D. Conover, The future of virtual Construction and regulation checking, virtual futures for design, *Constr. PRO* (2009) 241–250, 9781444302349.
- S. Park, J.K. Lee, Definition of a domain-specific language to represent Korea building act sentences as an explicit computable form, in: M.A. Schnabel, W. Nakapan, S. Roudavski, S.F. Chien, M.J. Kim, S. Choo (Eds.), 21st International Conference on Computer-Aided Architectural Design Research in Asia: Living Systems and Micro-Utopias: Towards Continuous Designing, CAADRIA 2016, The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), 2016, pp. 73–82, <https://www.itcon.org/paper/2016/26> (accessed May 3, 2022).
- P. Pauwels, D. Van Deursen, R. Verstraeten, J. De Roo, R. De Meyer, R. Van de Walle, J. Van Campenhout, A semantic rule checking environment for building performance checking, *Autom. Constr.* 20 (5) (2011) 506–518, <https://doi.org/10.1016/j.autcon.2010.11.017>.
- C. Preidel, A. Borrmann, Automated code compliance checking based on a visual language and building information modeling, in: 32nd International Symposium on Automation and Robotics in Construction and Mining: Connected to the Future, ISARC 2015, International Association for Automation and Robotics in Construction I.A.A.R.C., 2015, <https://doi.org/10.22260/isarc2015/0033>.
- C. Preidel, A. Borrmann, Towards code compliance checking on the basis of a visual programming language, *J. Inform. Technol. Construct.* 21 (2016) 402–421, <https://www.itcon.org/paper/2016/25> (accessed May 3, 2022).
- F. Puppe, *Systematic Introduction to Expert Systems: Knowledge Representations and Problem-Solving Methods*, Springer Science & Business Media, 2012 (3642779719).



- [46] W.J. Rasdorf, S. Lakmazaheri, Logic-based approach for modeling organization of design standards, *J. Comput. Civ. Eng.* 4 (2) (1990) 102–123, [https://doi.org/10.1061/\(ASCE\)0887-3801\(1990\)4:2\(102\)](https://doi.org/10.1061/(ASCE)0887-3801(1990)4:2(102)).
- [47] Solibri, Solibri Model Checker, Vol. 2021. <https://www.solibri.com/>, 2022.
- [48] W. Solihin, Lessons learned from experience of code-checking implementation in Singapore, in: BuildingSMART Conference, BuildingSMART, Singapore, 2004. [https://www.researchgate.net/publication/280599027\\_Lessons\\_learned\\_from\\_experience\\_of\\_code-checking\\_implementation\\_in\\_Singapore](https://www.researchgate.net/publication/280599027_Lessons_learned_from_experience_of_code-checking_implementation_in_Singapore) (accessed May 3, 2022).
- [49] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, *Autom. Constr.* 53 (2015) 69–82, <https://doi.org/10.1016/j.autcon.2015.03.003>.
- [50] W. Solihin, C. Eastman, A knowledge representation approach in BIM rule requirement analysis using the conceptual graph, *J. Inform. Technol. Construct.* 21 (2016) 370–402. <https://www.itcon.org/paper/2016/24> (accessed May 3, 2022).
- [51] J. Soliman-Junior, C.T. Formoso, P. Tzortzopoulos, A semantic-based framework for automated rule checking in healthcare construction projects, *Can. J. Civ. Eng.* 47 (2) (2020) 202–214, <https://doi.org/10.1139/cjce-2018-0460>.
- [52] J. Soliman-Junior, P. Tzortzopoulos, J.P. Baldauf, B. Pedo, M. Kagioglou, C. T. Formoso, J. Humphreys, Automated compliance checking in healthcare building design, *Autom. Constr.* 129 (2021), <https://doi.org/10.1016/j.autcon.2021.103822>.
- [53] X. Tan, A. Hammad, P. Fazio, Automated code compliance checking for building envelope design, *J. Comput. Civ. Eng.* 24 (2) (2010) 203–211, [https://doi.org/10.1061/\(asce\)0887-3801\(2010\)24:2\(203\)](https://doi.org/10.1061/(asce)0887-3801(2010)24:2(203)).
- [54] UK Government, *The Building Act 1984, 1984* (9781717088154).
- [55] UK Government, *The Building Regulations 2010, 2010* (9781859467152).
- [56] N. Yabuki, K.H. Law, An object-logic model for the representation and processing of design standards, *Eng. Comput.* 9 (3) (1993) 133–159, <https://doi.org/10.1007/BF01206345>.
- [57] A. Yurchyshyna, A. Zarli, An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction, *Autom. Constr.* 18 (8) (2009) 1084–1098, <https://doi.org/10.1016/j.autcon.2009.07.008>.
- [58] J. Zhang, N.M. El-Gohary, Semantic-based logic representation and reasoning for automated regulatory compliance checking, *J. Comput. Civ. Eng.* 31 (1) (2017), [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000583](https://doi.org/10.1061/(asce)cp.1943-5487.0000583).
- [59] J. Zhang, N.M. El-Gohary, Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking, *J. Comput. Civ. Eng.* 30 (2) (2016), [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000346](https://doi.org/10.1061/(asce)cp.1943-5487.0000346).
- [60] J. Zhang, M. El-Gohary Nora, Automated information transformation for automated regulatory compliance checking in construction, *J. Comput. Civ. Eng.* 29 (4) (2015), B4015001, [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000427](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000427).
- [61] R. Zhang, N. El-Gohary, Clustering-based approach for building code computability analysis, *J. Comput. Civ. Eng.* 35 (6) (2021), [https://doi.org/10.1061/\(asce\)cp.1943-5487.0000967](https://doi.org/10.1061/(asce)cp.1943-5487.0000967).
- [62] R. Zhang, N.M. El-Gohary, A clustering approach for analyzing the computability of building code requirements, in: C. Wang, C. Berryman, R. Harris, C. Harper, Y. Lee (Eds.), *Construction Research Congress 2018: Construction Information Technology*, CRC 2018, Vol. 2018-April, American Society of Civil Engineers (ASCE), 2018, pp. 86–95, <https://doi.org/10.1061/9780784481264.009>.
- [63] Z. Zhang, L. Ma, T. Broyd, Towards fully-automated code compliance checking of building regulations: challenges for rule interpretation and representation, in: EC<sup>3</sup> (European Conference on Computing in Construction), 2022, <https://doi.org/10.35490/EC3.2022.148>.
- [64] B. Zhong, H. Wu, L. Ding, H. Luo, Y. Luo, X. Pan, Hyperledger fabric-based consortium blockchain for construction quality information management, *Front. Eng. Manage.* 7 (4) (2020) 512–527, <https://doi.org/10.1007/s42524-020-0128-y>.