2022 European Conference on Computing in Construction
Ixia, Rhodes, Greece
July 24-26, 2022

TOWARDS FULLY-AUTOMATED CODE COMPLIANCE CHECKING OF BUILDING REGULATIONS: CHALLENGES FOR RULE INTERPRETATION AND REPRESENTATION

Zijing Zhang[1], Ling Ma[1], and Tim Broyd[1]
[1]University College London, London, United Kingdom

Abstract

Before the building design is finalised, it needs to be checked against regulations. Traditionally, manual compliance checking is error-prone and time-consuming. As a solution, automatic compliance checking (ACC) was proposed. Many studies have focused on the crucial ACC rule interpretation process, yet no research has synthesised the themes and identified future research opportunities. This paper thus aims to fill this gap by conducting a systematic literature review and identifying challenges facing this field. Findings revealed that the representation development process lacks a methodological backdrop. Understandings of rules, representations, and relationships between them are insufficient. Potential solutions were proposed to address these challenges.

Introduction

In the AEC industry, ensuring compliance is mandatory before moving on to the construction stage (Soliman-Junior et al., 2021). However, the current manual compliance checking process is often characterised by error-prone, time-consuming and costly (Han et al., 1998;Macit İlal and Günaydın, 2017). Therefore, automatic code compliance checking has been actively researched as a promising solution. It generally includes four steps: rule interpretation, target model preparation for checking, rule checking and reporting (Eastman et al., 2009).

There have been two approaching perspectives to achieving automatic compliance checking (ACC): the target model perspective and the rule perspective. The data model perspective primarily aims to find suitable methods for easy and efficient data retrieval and query (Solihin et al., 2020). While these studies typically develop new methods for easier retrieval and query data, they tend to pay less attention to the accuracy and expressiveness of rule representation (i.e., rule representations are sometimes restricted by target data model structure). By contrast, rule perspective studies aim to develop a computer-readable representation for building rules with minimum knowledge loss (Solihin and Eastman, 2016). These studies develop methods to interpret rules written in natural language to a suitable computer-readable form, which typically requires a deep understanding of rule structure, semantics and complexity (Hjelseth and Nisbet, 2011). However, ease of retrieval and query of data are less accounted for, which may compromise the efficiency of the ACC system.

The existing literature review on ACC mainly focuses on the ACC system (Eastman et al., 2009) or the target data model essential for ACC (Hu et al., 2021). Despite significant research efforts, the rule perspective research of ACC has yet to be synthesised. As the rule interpretation process is still a time-consuming bottleneck that impedes the efficiency and accuracy of ACC, it is essential to understand the current progress and existing issues to help the research community work towards problem-solving. Specifically, the research questions of this study are:

1)What are the themes of ACC research in terms of the rule interpretation process, and how are they related to each other? 2)What are the challenges facing ACC from the rule perspective?

Methodology

This research adopted a systematic literature review method. A systematic review is a widely adopted method to synthesise and evaluate the state of knowledge, identify research gaps and create research agendas in a research domain (Snyder, 2019). It has been used in multiple previous studies in the Architectural, Engineering and Construction (AEC) domain.

A keyword search was conducted using the Scopus database. First, peer-reviewed journal articles and conference proceedings that were written in English were selected. The authors then screened out papers that do not focus on the rule perspective of ACC, such as papers that only concern the data model structure or how the rule engine is built.

Results and Findings

Three main themes were found in the rule perspective: rule classification, rule organisation, and rule representation. Rule representation aims to use a computer-readable method to represent building rules in a structured way without losing or changing the meaning of the original regulation texts. It can be further divided into rule representation in general as the output of rule interpretation or as individual rule representation specifically. The interpretation of rules typically includes the analysis of individual rules and organisation among rules. Individual rule interpretation and organisation among rules can be overlapping as scholars have a different definition of rule scale. While primarily dealing with categorising individual provisions, rule classification sometimes includes criteria regarding interrelationships among rules, thus touching on the organisation of rules. Their relationships are shown in Figure 1. Primary studies are summarised in Table 1.

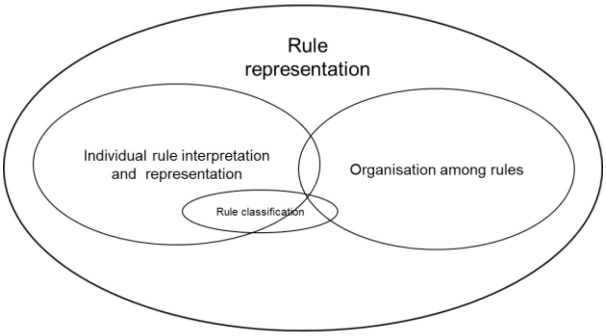

Figure 1: Relationships among research themes in the rule interpretation process

Theme 1: Rule classification

To assist the interpretation of rules, scholars have developed many classification methods. Many studies explored rule semantic structure. Macit İlal and Günaydın (2017) considered rule dependency, where rules were classified into self-contained and linked explanatory categories. The former refers to rules that are complete in themselves and do not depend on other rules, while the latter links to other rules as clarifications, exceptions, exemptions, or modifications. Nawari (2020) provided a more detailed classification with conditional, content, ambiguous, and dependent clauses. Conditional clauses can be directly interpreted into computable rules. Content clauses are definitions or clarifications that cannot be translated into true or false. Ambiguous clauses refer to clauses that have subjective wordings such as "big enough", "relatively". Moreover, dependent clauses mean whether the clauses need to be checked depending on other clauses. Dependent clauses are suggested to be checked manually due to the dependency. This classification method recognised the ambiguity and dependency of building rules.

Soliman-Junior et al. (2020) proposed a semantic-based framework for hospital rule-checking, where they classify rules against the nature of rule (i.e., qualitative, quantitative and ambiguous) and the possibility of translation into logical rule (i.e., logical, non-logical). This method attempted to explore the relationships between the type of rule and whether it can be translated into a machine-readable form but failed to reveal the characteristics that make the rules possible or impossible to be interpreted into logical rules. "Non-logical" rules are suggested to be checked manually.

Yurchyshyna and Zarli (2009) introduced a classification method based on interpretability related to the Industry Foundation Classes (IFC) model. The four types of requirements include: 1) completely IFC-interpretable (CI); 2) reformulated in IFC (RI) (requires human interpretation); 3) partially interpretable (PI) (the IFC model is not sufficient for interpretation); 4) non-interpretable (NI) (rules with ambiguous information or describing common knowledge). A major deficiency of this approach is that it restrained the rule representation to the IFC model that is often not sufficient to represent all required information in the rules, especially abstract geometrical and topological constraints (Preidel and Borrmann, 2016). As a result, too many rules are left to be checked manually.

The above classification methods are typically used to distinguish which rules are suitable for ACC and which are suitable for manual checking. For this reason, these studies tend to have a selective bias: only rule features that are not conducive to ACC become the main focus, such as dependency and conditionality, but the general features of rule provisions (e.g., semantic constructs and logic connectives) are unrevealed.

Some studies attempted to explore general rule features. Hjelseth and Nisbet (2011) identified four general constructs in each rule clause: 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 (Nisbet et al., 2009). In addition, the RASE method developed a dictionary to: 1) link the terms in the target model and representation; 2) maintain the consistency of terms; 3) 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 (Beach et al., 2015). It incorporated "output" and "total" constructs to represent the "actions" in BREEAM rules. Solihin and Eastman (2016) also generalised rule structure logic, including concepts, requirements, constraints, functions, derived data, exceptions and rule dependency. Notably, functions and derived data are implicit in rule provisions. Rule analysts capture them when turning the "black-boxes" into operable white-boxes. This highlights the importance to go beyond the syntax, language and grammar of rules to understand the complexity of the rules.

Considering the rule complexity about data structure, Solihin and Eastman (2015) developed a four-class classification method, including 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". This classification took a holistic view of the ACC system, where the rule representation is later mapped to the model view definition of the target model. This method provided insights into the objects, properties and relationships that need to be checked and the geometrical, mathematical algorithms and simulations required when executing rules. However, such a representation suffers from a similar problem with the classification proposed by Yurchyshyna and Zarli (2009), where the representation can be limited due to the limitation of IFC entities.

Zhang and El-Gohary (2021) developed a more comprehensive set of metrics considering syntactic,

Table 1: Major studies on rule classification, rule organisation and rule representation

Reference	Classification Criteria	Classification Objective	Rule Organisation Criteria	Representation Method
Fenves (1966)	NA	NA	NA	Production rule (decision table)
Garrett Jr and Hakim (1992)	NA	NA	Object-oriented, Class hierarchy	Object-oriented
Yabuki and Law (1993)	NA	NA	Object-oriented, class hierarchy	Object-oriented, Logic-based
Kiliccote et al. (1994)	NA	NA	Context-based	Object-oriented
Yurchyshyna and Zarli (2009)	Based on IFC interpretability	Ability for manual/automatic checking	Content-based	Semantic-based (semantic web)
Tan et al. (2010)	NA	NA	Cross-reference	Production rule (decision table)
Pauwels et al. (2011)	NA	NA	NA	Semantic-based (semantic web)
Lee et al. (2015)	NA	NA	NA	Language-driven (domain-specific language)
Beach et al. (2015);Hjelseth and Nisbet (2011)	Based on rule semantic constructs and logical connectives	Mark-ups for automatic checking	Cross-reference	Semantic-based (RASE)
Preidel and Borrmann (2016)	NA	NA	NA	Language-driven (visual programming language)
Solihin and Eastman (2015)	Based on rule complexity	Mapping to MVD for automatic checking	NA	NA
Solihin and Eastman (2016)	Based on general rule structure logic and interdependency	Identify constructs for rule representation	Cross-reference	Logic-based (conceptual graph)
Zhang and El-Gohary (2016)	Based on semantics	Simplifies the representation of patterns and numbers of patterns	NA	NA
Zhang and El-Gohary (2021)	Based on syntactic, semantic features and computability metrics	Computability by computer	NA	NA
Macit İlal and Günaydın (2017)	Based on interdependency	Ability for manual/automatic checking	Cross-reference	Semantic-based
Kim et al. (2019)	NA	NA	NA	Language-driven (visual programming language)
Nawari (2020)	Based on semantics and interdependency	Ability for manual/automatic checking	Cross-reference	Language-driven (LINQ)
Soliman-Junior et al. (2020);Soliman-Junior et al. (2021)	Based on semantics and logic	Ability for manual/automatic checking	NA	NA

semantic features and computability metrics. A total of 12 types have been identified using a clustering-based approach based on these metrics. However, this work only analysed IBC codes, while analysis of other regulation documents could find more syntactic, semantic and computability metrics. Furthermore, these metrics were tested only in the information extraction task but not in other tasks such as rule representation (Zhang and El-Gohary, 2021).

In summary, despite the many existing classifications, most of them are categorised using a single dimension or a few casually selected criteria or metrics. A consolidated list of metrics could better classify rules, thereby facilitating the rule interpretation process.

Theme 2: Rule representation methods

Several studies focused on rule representation. It provides a method for knowledge capture, which ease future modification and update of rules.

One of the earliest types of rule representation is production rules. It takes the form of "if <conditions> then <actions>". An example in this category is the decision table (Fenves, 1966), an unambiguous representation of applicable conditions and corresponding actions in tables. However, the main defect of this approach lies in its inability to show relationships among rules. Tan et al. (2010) proposed a new decision table approach with better expressiveness. Parameters (e.g., location, type) extracted from rules were used as sub-headings for the decision table. The reference index is used to denote the original link or cross-reference. However, this decision table can only deal with rules with similar conditions and actions. What is more, it still failed to show the logical relationships among rules.

Another category of representations is a logic-based method. Parametric tables in some commercial software such as Solibri Model Checker can be found in this category, although they are not technically a separate representation due to rules embedded in the rule engine. This method often suffers from a lack of transparency and is difficult to maintain (Solihin and Eastman, 2016). Some other work (Kerrigan and Law, 2003) adopted predicate logic to represent building rules. However, predicate logic can become lengthy and hard to read when dealing with complex rules. Solihin and Eastman (2016) proposed a conceptual graph (CG) approach to represent building rules to address this issue. It has the foundation in predicate logic but is tailored to the requirements of building rules using IFC properties and objects. Its graphic notations improve the readability. However, as it is based on predicate logic, it cannot represent actions other than true/false. In addition, the use of IFC in representing rules restrained this method to the target data model. As a result, only a limited number of rules can be represented using this approach.

Some more recent studies developed semantic-based representation methods. Hjelseth and Nisbet (2011) proposed a Requirement, Applicability, Selection and Exception (RASE) approach concerning the semantic constructs of rules. This approach kept the regulation text as is but added mark-ups during the interpretation process to assist automatic checking. A tree-like method was employed to represent the logical connectives among different constructs. One of the distinguishing features of RASE is that instead of using IFC expressions in rule representation, it resorted to an intermediate method (a dictionary) to link building model and rule representation.

Other attempts have been observed using natural language processing (NLP), which also concerned syntactic and semantic features of rules (Zhang and El-Gohary, 2016). Another semantic-based representation method is the semantic web (Pauwels et al., 2011;Yurchyshyna and Zarli, 2009). For example, SPARQL queries and query annotations formalise building rules (Yurchyshyna and Zarli, 2009). However, as this approach adopted an IFC-based ontology, it also suffered limited expressiveness. Pauwels et al. (2011) addressed this issue by creating a set of vocabulary extracted from rules using OWL. Nevertheless, the semantic web is criticised for having a steep learning curve and is thus not easy for domain experts to use.

The last category is language-based methods. It mainly includes domain-specific languages and visual programming languages. The BERA language (Lee et al., 2015) for building circulation and spatial rules falls into the former group. It is a domain-specific query language where the syntax and functions embedded in building rules can be represented. Therefore, it is easier to learn for non-programmers compared with general-purpose languages. Some more recent research used visual programming languages (Kim et al., 2019;Preidel and Borrmann, 2016) to represent rules. These languages can represent complex rule logic without computer programming (Solihin and Eastman, 2016). In this approach, small "white boxes" with known functions are linked to input and output ports by wires, making the rule-checking process transparent and easy to understand by rule experts (Preidel and Borrmann, 2016). Notwithstanding, it still has deficiencies when handling recursions in rules.

Theme 3: Rule organisation

The above-mentioned rule representation methods mainly focused on single rule provision, whilst few studies have explicitly considered rule organisation, that is, the relationships among rule provisions. However, it is essential to consider and represent rule dependencies because rules can be better organised, but the relationships and interdependencies affect the checking results.

One of the earliest rule organisation methods is the SASE (Standards Analysis, Synthesis and Evaluation) model. It includes two networks on four levels. The network on the top level is the organisational network (organising building codes). The information network (dependency relationships among provisions) includes an individual provision level (decision tables), derived data items level and basic data items level. However, this method does not apply to data items and has complex precedence relationships (Macit İlal and Günaydın, 2017).

There have been some attempts using object-oriented thinking. For example, Garrett Jr and Hakim (1992) organised rules around objects in rules. Yabuki and Law (1993) also employed an object-oriented modelling method to organise rules, but their object-logic hybrid method includes using data items and predicate logic to represent rule provisions. However, both models have been criticised for having too complex a class hierarchy and being cumbersome to handle (Kiliccote et al., 1994). To alleviate this issue, Kiliccote et al. (1994) developed a context-oriented model that organised building code around “contexts”, which are essentially a set of subclasses of the applicability constructs in provisions.

Yurchyshyna and Zarli (2009) recognised the issues of traditional regulation text organisation. Instead of organising by themes, they suggested that the rules be organised based on their contents. They developed three main classifications to organise rules, including classification 1) by construction, 2) by key concepts, and 3) by application condition. However, this approach failed to address the interdependencies among rule provisions.

Integrating SASE with RASE, Macit İlal and Günaydın (2017) proposed a method of organising rules on four levels, namely: domain level, rule level, rule-set level and management level. The domain level is a library of concepts, attributes, and definitions to avoid repeating definitions when representing rules. The rule-set level stresses the logical connectives among objects that appeared in rules. Finally, the management level recognises the importance of the overall organisation of building rules; it does so by grouping closely related rule sets (i.e., those with the same concepts) and connecting them. Nevertheless, this model is programming-intensive and requires the involvement of a software engineer.

Discussion

Many classifications, organisation and representation methods have been proposed but with limited practical implementation. The rule interpretation process has yet to be fully automated.

Transforming regulation texts to computer-readable rule representation can be divided into two steps, rule interpretation and rule representation. The rule interpretation step requires extensive experience and expertise in building rules. For this reason, currently, it is primarily a manual process where domain experts analyse rules and take notes about the constructs, logic and implicit assumptions of rules. This is a time-consuming step that can take up to 30% of the total time of implementing a rule (Solihin and Eastman, 2016). The importance of automating this step has been recognised, while the attempts using machine learning methods (e.g., NLP) have not led to very satisfactory results due to the complexity and diversity of regulation texts.

By contrast, the rule representation step may be easier to automate given all the rule semantic constructs, and logical connectives are already known. The RASE method (Nisbet et al., 2009) is an example of automating this step. It provides a range of different representations (e.g., IFC constraints) (Nisbet et al., 2009), plain language (Hjelseth and Nisbet, 2011), SWRL (Beach et al., 2015), all generated automatically from the RASE mark-upped regulation text. However, this method failed to connect different types of rules and different representations. This is important as not all representations are suitable for representing a specific type of building regulations.

It is crucial to tackling the following issues to achieve full automation of the rule interpretation and representation process (Figure 2).

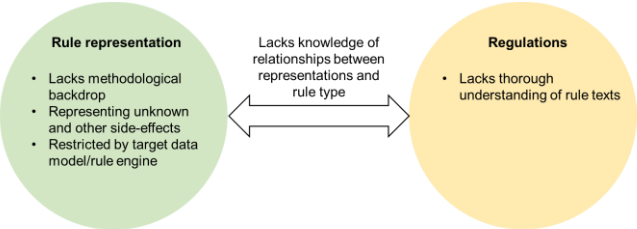

Figure 2: Issues in rule interpretation and representation of ACC

The current rule representation development process lacks a solid methodological backdrop.

The design of a better rule representation method falls into the scope of design science research (Hevner and Chatterjee, 2010), where the rule representation method is the primary artefact being developed. Artefacts are typically developed based on specific objectives to be achieved (Peffers et al., 2006). The artefact development process requires assessment and evaluation of design options (i.e., different representation methods) before final design decisions are made (Pries-Heje et al., 2008). Typically, the process also includes a few iterations.

However, in current literature, frequently new representations are arbitrarily borrowed from the available knowledge representation methods in other domains, without a thorough analysis of the objectives and required functionalities for representation. As a result, none of the representation methods can represent all types of rules (Macit İlal and Günaydın, 2017). Furthermore, little research has adopted evaluation methods to assess design options before the artefact is built. In addition, the current ex-post evaluation of the artefact is mainly based on prototype development and validation by researchers using the prototype without input and feedback from actual users. While the demonstrations showed the usability of the artefact, the performance tests (e.g., efficiency) were often not sufficient.

The artefact should be developed with more rigorous building and evaluation processes to address these issues. It is recommended to have both ex-ante evaluations for design options and ex-post evaluations to test artefact performance (Pries-Heje et al., 2008). In this way, the proposed artefact's usability, efficiency, and accuracy could be better assessed. The proposed artefact has a better chance of success before putting many resources into its development.

Reporting the "unknown" and representing other side effects

Many of the existing representation methods (e.g., predicate logic, conceptual graph (Solihin and Eastman, 2016) for building rules are drawn from the knowledge representation domain, a branch of artificial intelligence (AI). These methods are typically not tailored for building regulations and mostly only deal with binary results (i.e., true or false). However, in automatic compliance checking of building regulations, as data are not always available in the model submitted to the ACC system, it is crucial to have "unknown" when the corresponding data cannot be found in the model provided.

The binary results are underpinned by a closed world assumption, which regards what is not known as false (Hustadt, 1994). Considering that not all required information is available in the building model, this can easily result in many "false-negatives", which compromises the accuracy and reliability of ACC.

Two methods have been considered to alleviate this issue. The first one is to ask the modeller to follow a modelling guideline for the convenience of checking. However, modellers typically find this a huge burden (Amor and Dimyadi, 2021). It would also affect the efficiency of model preparation and, eventually, the whole ACC process. In addition, much of the extra work incurred from this process may not be of use during checking later on.

The second method makes an open-world assumption. During the checking, the required information may not be available. Using an open-world assumption, when this happens, instead of an immediate "false", it would be regarded as an "unknown". Further questions may be asked, and the decision will be made based on the supplemented information. The second method saves resources, time and cost as it only requires information when needed. However, currently only few ACC systems support "unknowns".

In addition, some rule clauses have side effects. For example, in BREEAM, the final score is calculated by adding credits from satisfying individual clauses (Building Research Establishment, 2018). Current methods drawn from other domains are not equipped with this function. For this reason, a domain-specific representation that is tailored for building regulations is needed. There have been attempts to adapt existing knowledge representation methods for building regulations. For example, Solihin and Eastman (2016) adapted the conceptual graph (Sowa, 1976) and proposed a new conceptual graph with features in line with BIM data. However, care must be taken when making such an adaptation. It could easily result in a lack of rigour because it may not be valid mathematically and logically.

Many representations are restricted by target model structure and rule engine

Many existing representations are developed using the properties, objects, and relationships from the target data model to facilitate the rule checking process. For example, many studies structured the regulation representation based on IFC's objects, properties, and relationships. However, although IFC has been updated many times and is fairly mature, its expressiveness is still limited. Some rules cannot be represented using the properties, objects, and relationships from IFC, or it would be very cumbersome to use IFC to represent (e.g., circulation rules). This may ultimately limit the usability and efficiency of the ACC system.

Apart from the restrictions from the target data model, some rule representations also suffer from the restriction by the rule engine. For example, to adapt to a specific rule engine, the objects to be checked need to be translated from natural language to the form acceptable in the rule engine to enable the matching and checking process. Examples of such methods include query-based (Solihin et al., 2020) and domain-specific-language-based (e.g., Lee et al., 2015) representations with specific syntax and grammars customised to the specific rule engine.

The expressiveness issue calls for an intermediate way to link different ontologies, thus achieving the connection of the target data model and the rule representation. A viable way for this seems to be a dictionary-like approach (e.g., Nisbet et al., 2009) using a customised and expandable dictionary to link the objects, properties and relationships in the data model (typically BIM model) to the rule representation. Such a method keeps the representation free from being restricted by the data model and rule engine structure.

Lacks thorough understanding of rule texts

The classification methods in literature can be categorised into the following types based on the main criteria used for classification: semantics, complexity, interpretability with IFC, logic, syntactic, interdependency or a mixture of some of these. Although these classifications shed some light on building rules, there are still some deficiencies of these methods:

1) many classifications only focus on one aspect of building rules (e.g., semantics). However, as rules have different semantic constructs, logic features, the complexity of checking, implicit assumptions, this could easily lead to a partial understanding of rules. Furthermore, when a comprehensive understanding of rules is lacking, it is easy to regard them as manual-checkable. This limits the scope of using automatic checking and ultimately affects checking efficiency.

2) many classifications are proposed solely to analyse whether a given rule can be checked automatically or not. Although this is an important goal of analysing rules, the classification could provide more benefits, such as identifying suitable representation methods based on the features of a specific type of rule.

Existing classification methods mainly focused on individual provisions. As a result, many of them failed to recognise the importance of requirement titles when interpreting these rule provisions, regarding them as irrelevant or even "nonsense" (Zhang and El-Gohary, 2021). However, titles could provide an essential context or applicability for the rule provisions. In addition, while focusing on individual rule provisions, the linkages and interdependencies among sentences have mainly been ignored.

To address this issue, a consolidated list of aspects should be used to conduct the rule analysis to obtain a thorough understanding of rules. Specifically, the consolidated list should consider two levels. On the lower level, it concerns the constructs of individual rules. This may include the complexity of checking the rules, the level of ambiguity, the semantic constructs (e.g., conditions, requirements), etc. The definition and scale of an individual rule also need to be made clear. On the higher level, the relationships among rule constructs and different rule clauses also need to be analysed.

Lacks knowledge of relationships between representations and rule type

The relationships between representations and rules are reflected by the representation method's capabilities and the capabilities needed to represent specific rules (Figure 3). Unfortunately, existing research largely ignores the exploration of such relationships. As a result, the representation development process typically improved certain aspects of representation (e.g., conciseness) at the cost of the others (e.g., expressiveness). This could be one of the reasons why there is still no representation capable of representing all building rules. Therefore, a change is needed in the representation development process to address this issue. It is envisaged that the new process should ideally include four steps:

1) identify what rules are being represented;

2) analyse and extract rule features;

3)make comparisons of several potential representations, analysing their capabilities and capabilities that some valid adaptations can acquire;

4) select and match rules and representations (make some adaptations if needed).

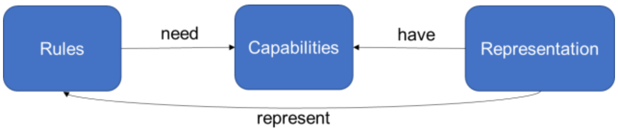

Figure 3: Relationships among rules, capabilities and representations

Using the proposed steps, developers will have a good idea of what capabilities they are looking for when they develop the representation method. The developed representation method can better fulfil the proposed goals by analysing the rule features and the relationships between rules and representations.

Conclusion

Automatic code compliance checking is a promising method to accelerate the design review and approval process, help achieve better quality, productivity gain and cost savings in the AEC industry. However, one of the keys to ensuring a favourable ACC result lies in accurate and efficient rule interpretation and representation.

Despite intensive research in this domain, the rule interpretation and representation processes are yet to be fully automated. Reasons for that lie across the rules, representations and the relationships between them. In this paper, the authors identified and explained five issues in the rule classification, rule representation and rule organistaion aspects, including: 1) the lack of solid methodological backdrop in current rule representation development processes; 2) reporting the "unknown" and representing other side effects; 3) representations are restricted by the target data model and/or rule engine; 4) the understanding of rule texts is not thorough; 5) lacks knowledge of relationships between representations and rule type. Potential solutions were also proposed. This research is significant to the research community and practitioners by providing research directions and potential ideas for representation development.

References

Amor, R. & Dimyadi, J. (2021). The promise of automated compliance checking. Developments in the Built Environment, 5, 100039.

Beach, T.H., Rezgui, Y., Li, H. & Kasim, T. (2015). A rule-based semantic approach for automated regulatory compliance in the construction sector. Expert Systems with Applications, 42, pp. 5219-5231.

Building Research Establishment. (2018). BREEAM New Construction 2018 (UK).

Eastman, C., Lee, J.M., Jeong, Y.S. & Lee, J.K. (2009). Automatic rule-based checking of building designs. Automation in Construction, 18, pp.1011-1033.

Fenves, S. J. (1966). Tabular decision logic for structural design. Journal of the Structural Division, 92, pp.473-490.

Garrett Jr, J.H. & Hakim, M.M. (1992). Object-oriented model of engineering design standards. Journal of computing in civil engineering, 6, pp.323-347.

Han, C.S., Kunz, J.C. & Law, K.H. (1998). Client/Server Framework for On-Line Building Code Checking. Journal of Computing in Civil Engineering, 12, pp.181-194.

Hevner, A. & Chatterjee, S. (2010). Design science research in information systems. Design research in information systems. Springer.

Hjelseth, E. & Nisbet, N. (2011). Capturing normative constraints by use of the semantic mark-up RASE methodology, pp. 1-10.

Hu, Z.Z., Leng, S., Lin, J.R., Li, S.W. & Xiao, Y.Q. (2021). Knowledge Extraction and Discovery Based on BIM: A Critical Review and Future Directions. Archives of Computational Methods in Engineering, pp.1-22

Hustadt, U. (1994). Do we need the closed world assumption in knowledge representation?. Citeseer.

Kerrigan, S. & Law, K.H. (2003). Logic-based regulation compliance-assistance, pp.126-135.

Kiliccote, H., Garrett, J.H., Chmielenski, T. J. & Reed, K. A. (1994). The Context–Oriented Model: An improved Modeling Approach for Representing and Processing Design Standards. ASCE, pp. 145-152.

Kim, H., Lee, J.K., Shin, J. & Choi, J. (2019). Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking. Journal of Computational Design and Engineering, 6, pp.143-148.

Lee, J.K., Eastman, C.M. & Lee, Y.C. (2015). Implementation of a BIM Domain-specific Language for the Building Environment Rule and Analysis. Journal of Intelligent and Robotic Systems: Theory and Applications, 79, pp. 507-522.

Macit İlal, S. & Günaydın, H.M. (2017). Computer representation of building codes for automated compliance checking. Automation in Construction, 82, pp. 43-58.

Nawari, N.O. (2020). Generalised Adaptive Framework for Computerising the Building Design Review Process. Journal of Architectural Engineering, 26.

Nisbet, N., Wix, J. & Conover, D. (2009). The Future of Virtual Construction and Regulation Checking, in Brandon, P., Kocaturk, T. (Eds),Virtual Futures for Design, Construction and Procurement, Blackwell, Oxfordshire. doi: 10.1002/9781444302349.ch17

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R. & Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. Automation in Construction, 20, pp. 506-518.

Peffers, K., Tuunanen, T., Gengler, C. E., Rossi, M., Hui, W., Virtanen, V. & Bragge, J. (2006). The design science research process: A model for producing and presenting information systems research, pp.83-106.

Preidel, C. & Borrmann, A. (2016). Towards code compliance checking on the basis of a visual programming language. Journal of Information Technology in Construction, 21, pp. 402-421.

Pries-Heje, J., Baskerville, R. & Venable, J.R. (2008). Strategies for design science research evaluation.

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. Journal of Business Research, 104, pp. 333-339.

Solihin, W., Dimyadi, J., Lee, Y.C., Eastman, C. & Amor, R. (2020). Simplified schema queries for supporting BIM-based rule-checking applications. Automation in Construction, 117.

Solihin, W. & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. Automation in Construction, 53, pp. 69-82.

Solihin, W. & Eastman, C. (2016). A knowledge representation approach in BIM rule requirement analysis using the conceptual graph. Journal of Information Technology in Construction, 21, pp. 370-402.

Soliman-Junior, J., Formoso, C.T. & Tzortzopoulos, P. (2020). A semantic-based framework for automated rule checking in healthcare construction projects. Canadian Journal of Civil Engineering, 47, pp. 202-214.

Soliman-Junior, J., Tzortzopoulos, P., Baldauf, J.P., Pedo, B., Kagioglou, M., Formoso, C. T. & Humphreys, J. (2021). Automated compliance checking in healthcare building design. Automation in Construction, 129.

Sowa, J.F. (1976). Conceptual Graphs for a Data Base Interface. IBM Journal of Research and Development, 20, pp. 336-357.

Tan, X., Hammad, A. & Fazio, P. (2010). Automated code compliance checking for building envelope design. Journal of Computing in Civil Engineering, 24, pp. 203-211.

Yabuki, N. & Law, K.H. (1993). An object-logic model for the representation and processing of design standards. Engineering with Computers, 9, pp. 133-159.

Yurchyshyna, A. & Zarli, A. (2009). An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction. Automation in Construction, 18, pp.1084-1098.

Zhang, J. & El-Gohary, N.M. (2016). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. Journal of Computing in Civil Engineering, 30.

Zhang, R. & El-Gohary, N. (2021). Clustering-Based Approach for Building Code Computability Analysis. Journal of Computing in Civil Engineering, 35, 04021021.