2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

INFORMATION REPRESENTATION AND AUTOMATIC MATCHING IN BUILDING REGULATIONS AND BIM

Sihao Li, Yangze Liang, Guangyao Chen, and Zhao Xu
Southeast University, Nanjing, Jiangsu, China

Abstract

Building design requires various disciplines and spatial relationships, which can lead to errors. This study introduces a comprehensive conceptual framework for automated BIM compliance checking. It involves the creation of knowledge graphs by formulating ontologies for building regulations and developing models for semantic role annotation. Data extraction pipelines are established using the Dynamo module within Revit to gather pertinent information from BIM models. Compliance checking logic is articulated using graphs to match the extracted knowledge from building standards with the information in BIM models. The practicality of this automated compliance-checking framework was tested using BIM models from two actual projects.

Introduction

The evolution of the design review process in building construction has transitioned from hand-drawn blueprints to Computer-Aided Design (CAD) (Balachandran et al., 1991) and Building Information Modeling (BIM) (Liu et al., 2022; Solihin et al., 2020), which has become a cornerstone for project approvals in numerous cities. In China, the exponential growth of BIM data is propelling digital transformation and smart city development (Dimyadi and Amor, 2013). While BIM compliance checks are essential for ensuring safety, regulatory adherence, and public interest, they are hindered by the complexity of BIM data. Manual reviews are not only inefficient and expertise-intensive but also prone to biases (Li et al., 2024).

Automated Compliance Checking (ACC) presents a viable solution to these challenges (Noardo et al., 2022). Research efforts have concentrated on standardizing regulations (Ismail et al., 2017) and integrating model data (Zhang and El-Gohary, 2017), resulting in tools such as Solibri Model Checker (Solihin et al., 2020) and Autodesk Navisworks (Hjelseth, 2015). These systems excel in tasks like clash detection and stairway inspections but face limitations in handling complex logic and interrelated data. Enriching model information remains a significant hurdle, and current systems depend on hard-coded rules, necessitating extensive programming and frequent updates to align with evolving standards (Zhang and El-Gohary, 2017). Although semantic inspection methods offer greater flexibility (Beach et al., 2015; Liebich et al., 2004), they lack the capability to automate rule construction and updates effectively.

Knowledge graphs (KG) (Singhal, 2012), introduced by Google in 2012, are renowned for their semantic reasoning and flexible knowledge representation capabilities (Li et al., 2021). KGs can structure and formalize building standards, converting unstructured clauses into computable data for machine interpretation. By leveraging soft-coding techniques and Information Extraction (IE), automated rule library creation becomes feasible, enabling domain experts to define rules without programming expertise, thereby enhancing BIM compliance efficiency.

However, existing KGs are constrained by their limited scope, poor transferability, and a primary focus on representation rather than reasoning. To address these limitations, this study proposes an ACC framework that integrates ontology, Natural Language Processing (NLP), and deep learning to automate KG construction for Chinese building standards. Logical relationships within KGs are utilized to interpret standards, while processes in Dynamo and Java establish connections between BIM models and standards, enabling automated compliance checks. An intelligent platform further augments the efficiency and accuracy of BIM reviews.

Experiment

This study introduces an integrated conceptual framework for building engineering that leverages Natural Language Processing (NLP), machine learning, and knowledge graph techniques to automate the compliance checking of BIM models. The primary objective is to enhance the efficiency and accuracy of this process through automation. The methodologies employed in this research are designed to be easily updatable and transferable, offering a foundation for the future advancement of knowledge graphs across diverse building engineering disciplines. An overview of the proposed methodology is illustrated in Fig. 1, accompanied by explanatory visuals for key concepts and steps. The framework comprises three interconnected components: knowledge graph

construction, BIM model information extraction, and the matching of BIM models to the knowledge graph for compliance checking.

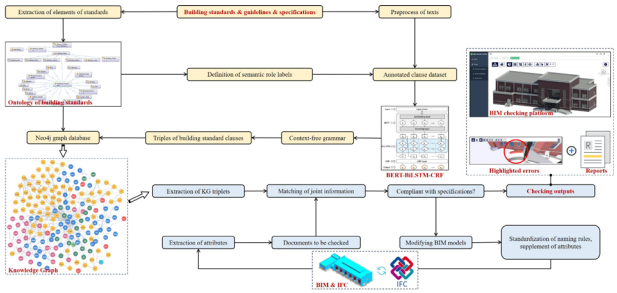


Figure 1: Overview of the proposed methodology framework

The initial step involves the construction of the building standard knowledge graph. This begins with a thorough analysis of standards and a review of relevant texts in the field. Based on this analysis, an ontology encompassing architectural and fire protection elements is developed. The building standard documents undergo preprocessing, with semantic role labels applied to them. A corpus of building standards is subsequently compiled and annotated with these labels. A deep learning model, specifically BERT-BiLSTM-CRF, is trained to automatically assign labels to the building standard texts. Syntactic analysis is conducted using Context-Free Grammar (CFG), and knowledge graph triples are extracted from the annotated texts. Finally, Neo4j is utilized to construct and store the knowledge graph, where nodes and attributes are designed to maintain the logical relationships derived from the building standards, thereby enhancing the graph's reasoning capabilities.

Following this, the BIM model checking data is predefined, and a data extraction method based on Dynamo is developed to retrieve the necessary files for checking. A matching algorithm is then formulated to represent the logical rules of the knowledge graph triples, culminating in an automated compliance checking method for BIM models. An intelligent BIM model checking platform is established and validated through two engineering case studies. The platform effectively conducts automated compliance checking for both architectural and fire protection aspects of BIM models, producing detailed checking reports. Given that the building standards utilized in this study are based on Chinese regulations, the research primarily focuses on Chinese texts, with comparative English translations provided for clarity.

Building design and construction standards serve as the cornerstone for compliance checks. This study focuses on several widely adopted Chinese standards in architecture and fire protection, chosen for their broad coverage, comprehensive content, and extensive applicability. These standards include the "Unified Standard for Civil Building Design (GB 50352-2019)", "Residential Building Code (GB 50368-2005)", "Design Code for Residential Buildings (GB 50096-2011)", "Fire Protection Design Code for Buildings (GB 50016-2014)", "Technical Standard for Fire Emergency Lighting and Evacuation Indication Systems (GB 51309-2018)", "Fire Protection Design Code for Automobile Garages, Repair Garages, and Parking Lots (GB 50067-2014)", and "Fire Protection Design Code for Interior Decoration of Buildings (GB 50222-2017)". These standards provide the data foundation for developing the ontology of building standards, which is subsequently utilized to construct the knowledge graph (KG) through information extraction and syntactic analysis.

The process of building the KG in this study follows a structured approach: First, the ontology of building standards is developed. Next, the building standard texts undergo preprocessing. Based on this ontology, custom semantic role labels are designed, and the preprocessed texts are annotated with these labels to create a labeled dataset. A semantic role labeling model is then trained using this dataset to automatically annotate the building standard texts. Following this, syntactic analysis is performed on the annotated texts, which are parsed to extract data triples for the knowledge graph. Finally, the knowledge graph is constructed by integrating the pattern layer and data layer.

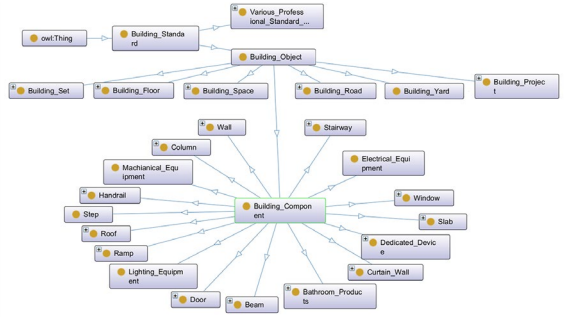
Ontology Construction

The clauses in building standards establish specific constraints for engineering design and construction, serving as the foundational knowledge for constructing the ontology and knowledge graph in this study. These clauses are classified into mandatory and non-mandatory types. Mandatory clauses are those that must be strictly followed, while non-mandatory clauses represent guidelines that are generally recommended under typical conditions. Mandatory clauses take precedence over non-mandatory ones. In this study, a hierarchical analysis of building standard knowledge and the logical relationships among various construction elements was conducted. A seven-step method was employed to develop the building standard ontology, which serves as the foundation for semantic role labeling and knowledge graph construction.

The research focuses on the compliance checking of BIM models, with the ontology's scope defined to cover architecture and fire control. By integrating concepts, terminology, and attributes from existing ontologies such as Ontolingua and DAML, the development of the building standard ontology in this study is enhanced to improve its efficiency and completeness. Key terms from the selected criteria are counted to generate a comprehensive list of all relevant terms.

The ontology development in this study adopts a top-down approach. Initially, key concepts are extracted from the selected standard texts to form the classes within the ontology. A hierarchical class structure is then created through a stepwise subdivision from top to bottom. The building code is designated as the parent class, which is divided into two main categories: professional standards and building objects. These categories are further

subdivided into subclasses, each containing various underlying objects. Some of these objects can be classified into different types of instances, while others are directly treated as instances that do not require further subdivision. The professional standards category includes building standards and fire control standards, while the building objects category is divided into subclasses such as project, building, space, site, storey, and component. A partial view of the constructed ontology for building standard texts is illustrated in Fig. 2.



Recognition of Semantic Role Labels

Based on the constructed ontology and Chinese word segmentation method, 6 semantic role labels have been customized, as shown in Table 1.

Table 1: Labels of semantic roles

Label	Definition	Interpretation
JD	Building object	Including building project, building site, building, building floor, building component, building space, building road, etc., referring to the semantic entities in the triples.
DS	Object property	Elements lower than JD, which are object properties of JD, including containment relationships and spatial relationships, referring to the semantic relationships between entities in the triples.
SS	Data property	Elements lower than JD, which are data properties of JD, including numerical properties, non-numerical properties, and measures, referring to the semantic properties in the triples.
SXZ	Property value	Requirements and conditions connected to property SS, with data formats such as Float, String, Boolean, referring to the semantic relationships between entities and properties in the triples.
BJ	Comparative term	Comparative relationship between property SS and requirement SXZ, including "greater than", "equal to", "less than", "should not be less than", and other comparative terms, referring to the semantic relationships between properties and property values in the triples.
LJ	Logical term	Expressing logical semantic relationships such as "parallel", "optional" including "and", "both", "except for", etc.

The selected building code provisions are annotated using the BIO (Begin-Inside-Outside) sequence labeling method, where "B" signifies the beginning of a label, "I" represents the middle position, and "O" indicates that the token does not belong to any label type. For instance, the sentence "Residential buildings with a height greater than 33m should be equipped with fire elevators" is annotated, as illustrated in Fig. 3. In this study, a total of 11,174 lines of data were annotated, and the datasets for training, development, and testing the deep learning model were constructed in an 8:1:1 ratio.

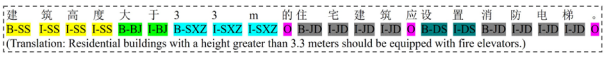


Figure 3: BIO sequence labeling data

After constructing the dataset, the BERT-BiLSTM-CRF model is employed to automatically label the semantic roles of building standard texts. The model architecture comprises three levels. First, the pre-trained BERT layer generates embedding vectors and outputs encoded sequence vectors $T=(t_1,t_2,\cdots,t_n)$. Next, the Bi-LSTM layer extracts semantic features from the text data by integrating contextual information, producing a bidirectional output sequence vector h_i, which indicates the probability of each position being assigned a BIO label. Finally, the CRF layer adjusts and optimizes the sequence vectors, outputting the most probable label sequence $Y=(y_1,y_2,\cdots,y_n)$.

Extraction of the Triplets

In the triples, entities and attributes correspond to JD, DS, and SS labels, which can be extracted by identifying entities under each label based on the output of the statistical model. Additionally, Context-Free Grammar (CFG) is employed to perform syntactic analysis on the labeled results, aiding in the identification and refinement of more complex relationships within the triples.

CFG is a formal language grammar that describes all possible string combinations generated by a probabilistic process, with sentence generation adhering to predefined rules. By analyzing common semantic representations in building standard texts, four types of rules are classified after CFG parsing and adjustment:

<JD-SS-BJ-SXZ>: Constraints on certain numerical attributes of building objects. For example, "The fire resistance grade of a high-rise factory should not be lower than Grade II."

<JD-DS-SXZ>: Constraints on measures or non-numeric attributes of data. For example, "The floor of the toilet should adopt waterproof construction measures."

<JD-DS-JD>: Constraints on the existence of certain building objects, equipment, or systems, or constraints on category matching for building objects. For example, "Residential buildings should have a lighting power supply system."

<JD-JD-DS>: Constraints on the spatial relationship between certain building objects. For example, "Electrical wiring pipes should not be set next to the toilet."

Taking the first rule as an example, the automatic semantic labeling result is illustrated in Fig. 4. By directly applying CFG, the syntactic tree is generated. According to the part-of-speech labeling rules, "high-rise factory" is identified as a noun phrase (NP) terminal, where "high-rise factory" is the entity to be extracted. Both "high-rise factory" and "fire resistance grade" are labeled as nouns (N), making it challenging to distinguish between the entity and attribute. Simultaneously, "should not be lower than" is split into "should not" and "be lower than," corresponding to the auxiliary verb (Aux) and verb (V). For the knowledge graph triple, "should not be lower than" should be extracted as a whole, representing the relationship between the attribute and its value.

高	层	厂	房	的	耐	火	等	级	不	应	低	于	二	级	。
B-JD	I-JD	I-JD	I-JD	O	B-SS	I-SS	I-SS	I-SS	B-BJ	I-BJ	I-BJ	I-BJ	B-SXZ	I-SXZ	O

(Translation: The fire resistance rating of high-rise buildings should not be lower than level 2.)

Figure 4: Automatic annotation results of semantic labels

At this stage, the triples cannot be directly derived from the syntactic tree alone. By integrating the semantic role labels with the syntactic tree, the semantic labels were modified. Specifically, the concept of JD corresponds to the entity in the triple, SS corresponds to the data attribute, and SXZ corresponds to the attribute value linked to the SS attribute. The knowledge graph triple for this rule, along with its minimal unit triple, is illustrated in Fig. 5. Following a similar approach, adjustments are applied to the labeled results, enabling the extraction of triples from the standard text.

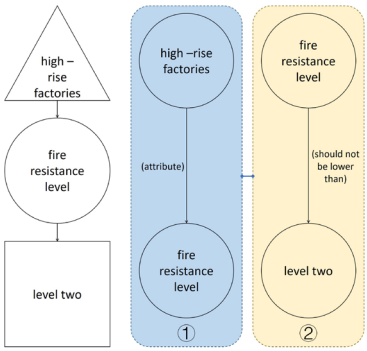


Figure 5: Knowledge graph triplet

Generation of Knowledge Graph

Once the triples are obtained, the knowledge graph is constructed and stored using the Neo4j database. The ontology triples developed in Protégé software are parsed and converted into ".turtle files," which are then imported into Neo4j to generate the knowledge graph network. To maintain logical relationships and fulfill the requirements for the subsequent automated review of BIM models, the knowledge triples of the clauses are systematically organized. Additional relationships are established, such as (standard) -[clause]-> (clause number), (clause number) -[clause type]-> (mandatory/non-mandatory), and (clause number) -[clause constraint]-> (building object). Entity nodes corresponding to each clause are created individually, as depicted in Fig. 6.

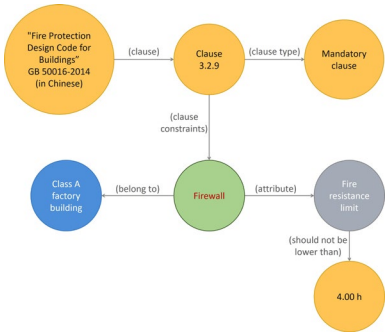


Figure 6: Example of a relational triplet

Data extraction

In this study, the automated compliance checking of BIM models involves two key steps: data extraction and matching. Dynamo is utilized to extract specific data from BIM models, encompassing tasks such as adding, extracting, and completing model data, as illustrated in Fig. 7 and Fig. 8. Following this, the extracted data is matched with the knowledge stored in the knowledge graph (KG) using triples, enabling the compliance check of the model. While Dynamo's visual nodes can extract certain parameters from model elements, their functionality is limited, as some parameters cannot be directly accessed using the built-in nodes. To address this limitation, this study integrates Dynamo's visual nodes with PythonScript to call the Revit API for data extraction and compliance checking.

Certain parameters required for compliance checks are not included in the default element parameters. Therefore, it is necessary to define and add custom attribute parameters based on the specific checking requirements to ensure comprehensive extraction of model information. In Revit, project parameters serve as a valuable tool, acting as containers for element data. These parameters can be customized to include various attributes, such as fire resistance level, combustibility, and mitigation measures. Additionally, project parameters allow users to flexibly define element attributes based on the specific requirements of the compliance checks. Revit elements

are primarily categorized into Categories, Families, Family Types, and Family Instances. Family parameters are divided into type parameters and instance parameters. Type parameters are shared across all instances of a family type, meaning that modifying them will alter all instance parameters within that family. Conversely, modifying the instance parameters of a specific family instance will only affect that instance. In this study, custom parameters are added as instance parameters to facilitate the assignment and extraction of parameters for various components. The batch addition of these parameters follows a programming workflow in Dynamo. When creating a new family instance, the instance parameters will automatically include the custom parameters, allowing the required data to be input into the model.

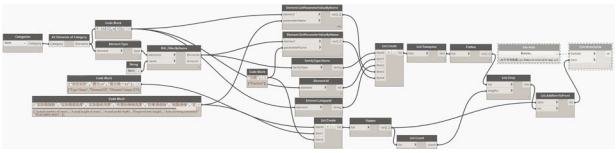


Figure 7: Extraction of step parameters

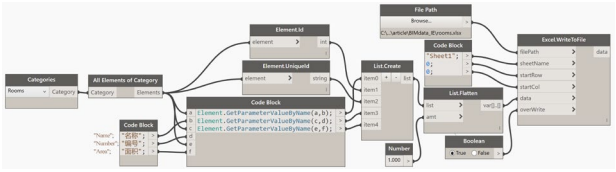


Figure 8: Process of writing Excel file

Discussion and result analysis

The experimental setup for the BERT-BiLSTM-CRF model, as described in Section 2.2 for automatic semantic role labeling, includes a DELL R730 server, a 64-bit Linux operating system, Visual Studio Code (Version 1.71), the TensorFlow 1.13.2 framework, and an NVIDIA T4 GPU. The model's performance is evaluated based on accuracy, precision, recall, and F1 score, derived from the confusion matrix. The formulas for these metrics are provided in Formula (1), (2), and (3). The performance of the semantic role annotation model in extracting various semantic labels is detailed in Table 2.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 \times \frac{precision * recall}{precision + recall} \tag{3}$$

Table 2: FID score statistics for different models

Label	Precision (%)	Recall (%)	F1
BJ	100.00	100.00	1.000
DS	77.70	87.50	0.824
JD	96.55	95.45	0.960
LJ	100.00	95.24	0.976
SS	95.12	97.50	0.963
SXZ	95.65	95.65	0.957

A web platform integrating the knowledge graph (KG) and the described algorithms has been developed to streamline BIM model compliance checking. The platform employs a Browser/Server (B/S) architecture and is structured into three layers: the Data Access Layer (DAL), the Business Logic Layer (BLL), and the User Interface (UI) Layer. The DAL manages the storage of standard ontologies, the KG, and BIM model databases, facilitating access through web service interfaces, drivers, and open standards. The BLL integrates both business functions and logical entities. Business functions include view, system, model, and project management, while logical entities consist of web servers, database servers, engines, and scripts. The platform is built on HTML5 and WebGL, with functionality implemented through CSS and JavaScript (JS) scripts. Users interact with BIM data on the visualization platform via the UI, performing operations such as adding, deleting, editing, querying, and generating compliance reports.

To validate the feasibility of this method, a factory engineering project was utilized, as illustrated in Fig. 9. For the building project, elements such as "steps," "stairs," "ramps," "railings," "doors," "firewalls," "roof access points," and "rooms" were selected for automated architectural and fire safety checks. The relevant data from the models was extracted and subsequently analyzed and assessed using matching algorithms.

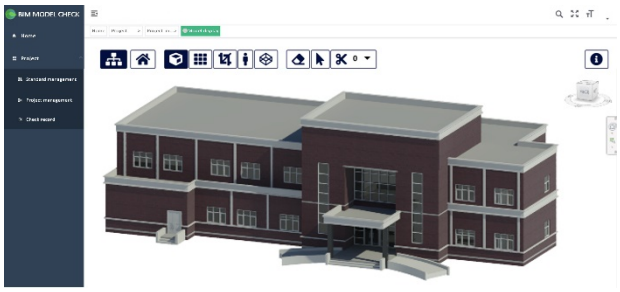


Figure 9: Compliance check platform for BIM models

Finally, an automated compliance report is generated for each BIM model. Non-compliant components are identified in the visualized model using their unique IDs and highlighted in distinct colors. For example, as depicted in Fig. 10, a ramp in the industrial building is flagged as non-compliant. According to Article 6.7.1-4 of the "Uniform standard for design of civil buildings (Chinese)", When the total step height is greater than 0.7m, protective measures must be installed at open edges. In this model, the step height is 0.75m, but the "protective measures" field is a "null".

Figure 10: Highlighted ramp in the building model

Conclusions

An integrated framework for the automated compliance checking of BIM models, utilizing Natural Language Processing (NLP) and knowledge graphs, has been developed based on the analysis of standards and BIM model data. The feasibility and effectiveness of this approach were demonstrated through two case studies. The theoretical value of this research lies in the proposal of a method to automate BIM model compliance checking, which also enhances the automation and transferability of domain-specific knowledge graph construction. Its practical significance is evident in the development and successful implementation of a BIM model checking system, providing both a theoretical foundation and practical guidance for related research and applications. The main contributions of the research are as follows:

1. An ontology for the building standards domain was developed using the "seven-step method," and a semantic role labeling model for building standard texts was created based on BERT-BiLSTM-CRF.

2. The semantic role labeling model was integrated with CFG-based analysis techniques, and a knowledge graph was constructed and stored in the Neo4j database. Logical markers were incorporated to enhance the graph's reasoning capabilities.

3. Script tools were developed using Dynamo and the Revit API to facilitate the addition, extraction, and supplementation of parameters for BIM model data.

4. Logical rule expressions for information matching were formulated, and a method for matching knowledge graph triples with BIM model data was proposed, enabling automated compliance checking.

Compared to traditional manual checking methods, the automated process in this study reduces human involvement, improves efficiency, and minimizes omissions caused by oversight. The advantages of automation are particularly evident in large-scale projects.

Due to time and knowledge limitations, the standards selected in this study cannot cover all regions and disciplines. In other fields, the generalization ability of this model still has room for improvement, and compatibility testing has not been conducted on software versions or file formats beyond those mentioned in this study.

However, this framework is extensible and generalizable. Entity labels can be redefined and retrained based on different language structures. Models like BERT have proven effective in various languages or fields (Devlin et al., 2018; Li et al., 2024). Future research will expand engineering categories and regulations to build structured graph databases. Developing tools for better data interoperability, including encoding and decoding rules, will be crucial for automated data checking.

References

Balachandran, M., Rosenman, M. A., and Gero, J. S. (1991). A knowledge-based approach to the automatic verification of designs from CAD databases. In J. S. Gero (Ed.), Artificial Intelligence in Design '91 (pp. 757-781). Butterworth-Heinemann.

Beach, T. H., Rezgui, Y., Li, H., et al. (2015). A rule-based semantic approach for automated regulatory compliance in the construction sector. Expert Systems with Applications, 42(12), 5219-5231.

Devlin, J., Chang, M.-W., Lee, K., et al. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv: 1810.04805.

Dimyadi, J., and Amor, R. (2013). Automated building code compliance checking: Where is it at? Proceedings of CIB WBC 2013, 172-185.

Hjelseth, E. (2015). BIM-based model checking (BMC). Building Information Modeling–Applications and Practices, 33-61.

Ismail, A. S., Ali, K. N., & Iahad, N. A. (2017). A review on BIM-based automated code compliance checking system. 2017 international conference on research and innovation in information systems (icriis) (pp. 1-6). IEEE.

Li, S., Wang, J., and Xu, Z. (2024). Automated compliance checking for BIM models based on Chinese-NLP and knowledge graph: An integrative conceptual framework. Engineering, Construction and Architectural Management.

Li, S., Jiang, Z., & Xu, Z. (2024). BIM-Based Model Checking: A Scientometric Analysis and Critical Review. Applied Sciences, 15(1), 49.

Liebich, T., Wix, J., and Qi, Z. (2004). Speeding-up the submission process: The Singapore e-plan checking project offers automatic plan checking based on IFC. International Conference on Construction Information Technology (INCITE 2004), 245-252.

Liu, H., Cheng, J. C. P., Gan, V. J. L., et al. (2022). A novel data-driven framework based on BIM and knowledge graph for automatic model auditing and quantity take-off. Advanced Engineering Informatics, 54, 101757.

Li, X., Lyu, M., Wang, Z., et al. (2021). Exploiting knowledge graphs in industrial products and services: A survey of key aspects, challenges, and future perspectives. Computers in Industry, 129, 103449.

Ministry of Housing and Urban-Rural Development of China. Uniform Standard for Design of Civil Buildings (Chinese), GB50352-2019. Beijing: China Architecture & Building Press, 2019.

Noardo, F., Wu, T., Arroyo Ohori, K., et al. (2022). IFC models for semi-automating common planning checks for building permits. Automation in Construction, 134, 104097.

Singhal, A. (2012). Introducing the knowledge graph: things, not strings. Official google blog, 5(16), 3.