The Beginning, Not the End: Revisiting Automated Compliance Checking for BIM-based Design Adaptation

Jiabin Wu,[1] Stefan Fuchs,[2] Tanya Bloch,[3] and André Borrmann[4]

[1]Chair of Computing in Civil and Building Engineering & TUM Georg Nemetschek Institute,
Technical University of Munich, 80333, Munich, Germany; e-mail: j.wu@tum.de
[2]Chair of Computing in Civil and Building Engineering & TUM Georg Nemetschek Institute,
Technical University of Munich, 80333, Munich, Germany; e-mail: stefan.m.fuchs@tum.de
[3]Faculty of Civil and Environmental Engineering, Technion – Israel Institute of Technology,
3200003, Haifa, Israel; e-mail: bloch@technion.ac.il
[4]Chair of Computing in Civil and Building Engineering & TUM Georg Nemetschek Institute,
Technical University of Munich, 80333, Munich, Germany; e-mail: andre.borrmann@tum.de

ABSTRACT

Automated Compliance Checking (ACC) evaluates Building Information Modeling (BIM) models against regulations, assisting designers in ensuring building design compliance. While existing research primarily focuses on enabling automatic checks, the automation of post-checking design modifications remains largely unexplored. This study aims to establish a theoretical foundation for a framework to evaluate the applicability of automated design adaptation in cases of non-compliant designs. By analyzing regulatory requirements and integrating the parametric modeling principles, we refine the key ACC feature types that enable automated modifications. Using the RASE markup and an ACC-centered rule extraction method, we investigate the International Building Code (IBC) accessibility requirements to identify property-specific features requiring adaptation in cases of rule violations. This approach demonstrates the feasibility of extracting modification-related objects and identifying parametric characteristics for post-checking design adaptations, supporting the integration of compliance checking with parametric design workflows to enhance efficiency.

INTRODUCTION

Automated Compliance Checking (ACC) helps designers evaluate BIM models against building regulations (Amor & Dimyadi, 2021). A few studies address code-compliant design generation, but ACC's broader potential for post-checking design adaptation is underexplored. Code violations often necessitate design changes of varying complexity, ranging from object properties to single-object, multi-object, and design topology levels (Sacks et al., 2018). Existing ACC-related research primarily addresses rule interpretation, model preparation, and checking processes, with limited relevance to design corrections to achieve automated code compliance (Wu et al., 2025). Varying building regulations present challenges to developing a generalized approach to code-compliant design adaptation across different requirements.

The ACC results are typically presented in textual formats, at best represented by instances of the BIM Collaboration Format (BCF) that integrates textual descriptions with screenshots of the identified issues and directly references objects within the model. Designers must manually interpret and implement these results to achieve code-compliant designs, a process that is inherently laborious. Consequently, reporting ACC results in a way that supports the effective correction of identified issues is crucial (Solihin & Eastman, 2015). Moreover, addressing one design issue can easily cause violations of previously satisfied requirements or invalidate integrity relations in other parts of the design (Eastman et al., 1997). Thus, there is a need to automate ACC-related design corrections and reduce manual work for designers.

This study revisits the potential of ACC in supporting the adaptation of BIM-based designs to achieve code-compliant solutions. This study aims to develop a theoretical foundation for identifying the regulatory scenarios eligible for automated adaptation approaches and those unsuitable for automated correction. Specifically, it analyzes the regulatory features and integrates them with parametric modeling principles. It focuses on analyzing how these features can be linked to parametric characteristics—i.e., design parameters and interrelationships—that significantly influence automated design adaptations. The proposed method is illustrated through accessibility rules defined in the International Building Code (IBC) (International Code Council (ICC), 2021). This study lays the foundation for establishing ACC-based automatic design adaptations, showing that ACC marks the beginning, not the end, of achieving more efficient design workflows.

The paper is organized as follows: It begins with the research background, followed by the research gap and objectives. The proposed approach is then detailed, evaluating selected building requirements. Finally, key findings, limitations, and future research directions are discussed.

BACKGROUND

ACC-related Design Adaptations. It is noted that the community has reached a point where ACC systems provide checks of BIM models against building regulations (Amor & Dimyadi, 2021). Despite the relatively mature development of ACC approaches, there remains a need to revisit their potential to provide efficient guidance for post-checking design modification processes, enabling suitable decision-making and streamlined design adaptation (Soliman-Junior et al., 2022). Several studies investigated how ACC can be leveraged to enhance design processes in different ways. For instance, a rule-checking-based design recommendation system has been developed to address specific door types of fire code violations (Lee et al., 2019). This system emphasizes semantic issues related to fire objects, with less focus on more challenging geometric and topological design issues. On the other hand, regulatory requirements have been directly integrated as generation rules to find code-compliant design alternatives, such as interior space design configurations (Sydora & Stroulia, 2020), and quantity take-off performance (Liu et al., 2022). By integrating domain-specific regulatory knowledge as constraints for design generation, these studies support correct model generation. However, compliance checking, often mandated by building authorities (Pfeiffer & Urban, 2024), can be bypassed in such rule-based generation methods, and streamlined ACC workflows with their generated checking results remain underutilized for correcting and refining design models. Recently, a Large Language Model (LLM) approach has been proposed to enable automatic code compliance checking and autonomous BIM design revision (Ying & Sacks, 2024). Nevertheless, the design revisions investigated are limited to single-object properties, such as the thickness of concrete floor slabs. Further experiments involving diverse and complex design requirements are needed to comprehensively evaluate the effectiveness of this approach. With the aim of having a generic adaptation framework, a design healing approach was introduced to directly link ACC results with the post-checking design adaptation process (Wu et al., 2025). While this framework establishes a foundation for formal design knowledge representation and effectively addresses code violations in architectural spatial designs, its ability to handle the broader complexities of building regulations remains a significant challenge. Therefore, further research is required to identify feasible design adaptation types for varying aspects of compliance checking.

Leveraging compliance checks to automated model adaptation requires revisiting the ACC outcomes, as they serve as one of the essential input data for post-checking design adaptation. While the textual checking results can be interpreted using the rapidly evolving Natural Language Processing (NLP) techniques (Yang & Zhang, 2024), the description detail levels in the generated BCF files heavily depend on how the predefined checking rules are structured within the compliance checker. Given the variability in the quality and consistency of textual descriptions, analyzing the textual checking results is out of the scope of this research. The Globally Unique Identifiers (GUIDs) in the BCF are usually utilized to link to directly related building elements within the BIM model (Schulz et al., 2021), which is typically the minimum required data for compliance checking outcomes. However, the ACC results usually do not incorporate the interrelations among building elements or the complexity of the evaluated requirements, particularly the specific dependencies relevant to targeted compliance checks. Other parts that are indirectly related to the requirements are excluded from the ACC results. This limitation reduces the value of ACC approaches in supporting effective design issue resolution. For example, a code violation related to fire safety door requirements might be addressed by directly adjusting the related fire-rating properties or globally replacing all relevant fire safety doors with code-compliant alternatives. Similarly, resolving excessive evacuation time issues of a building design necessitates accounting for specific design parameters and topological dependencies among all circulation-related components, such as stairways, accessible ramps, interior doors (e.g., space doors and storey exits), and exterior doors (e.g., building exits). Identifying the building components directly associated with the design issues is primary for ensuring code compliance. However, insufficient consideration of both direct and indirect design dependencies among components and spaces limits the effectiveness of applying ACC for automated design adaptation. Systematic analysis of design dependencies for code compliance checks necessitates consideration of two perspectives: the compliance checking aspect and the BIM-based building design aspect.

ACC-related Classification Systems. Given the diversity of regulatory requirements across building codes, as interpreted in ACC studies, the framework proposed by Solihin and Eastman (2015) offers a classification of building checking rules. Building rules are classified into four classes according to their computational complexity and requirements imposed on the rule execution environment. This classification provides a foundation for developing automated rule-checking systems that target one or more rule classes. Similarly, BIM-based model checking concepts are classified into four major types: validation checking, model content checking, smart object checking, and design option checking (Hjelseth, 2016). However, the established rule classification and concept category have limited relevance for achieving code-compliant designs. This limitation arises because the ACC process is focused on evaluating design data, whereas data modification is required in design adaptation. The application of ACC systems for code-compliant design adaptation remains unexplored, necessitating a suitable classification framework to group regulatory requirements into categories where specific adaptation types are required.

Developing a generic framework to ensure ACC-based automated code-compliant designs is challenging. This difficulty arises not only from the diverse types of regulatory requirements but also from the quality and richness of information represented in the design. Building rules involve different concepts and properties (Bloch & Sacks, 2020), necessitating semantic enrichment (SE) of BIM models as a prerequisite for ACC. Furthermore, the objects or parameters being checked in ACC may either be absent from BIM models or not directly available within existing data structures (Solihin & Eastman, 2015). In parametric building design, automated modifications primarily rely on the parameters within BIM models. Parametric modeling uses dependencies and constraints to create flexible models that adjust efficiently to changing conditions (Borrmann & Berkhahn, 2018). Modern parametric modeling systems consist of different parametric levels: solid modeling, assembly modeling, topology, and script-based modeling (Sacks et al., 2018). Basic dependencies and constraints are incorporated in most BIM authoring tools, while advanced designer-driven parametric techniques are mainly utilized in conceptual design tools such as Rhino[1]. Consequently, it is essential to account for basic parametric principles across different levels to analyze the parameters involved in different ACC-related modification tasks.

RESEARCH GAP AND OBJECTIVES

While ACC enables designers to evaluate building designs against codes, post-checking adaptations remain largely manual and labor-intensive. Identifying non-compliance issues alone does not ensure resolution, as effective adaptation requires specific modifications. Existing ACC research has limited exploration and relevance in supporting automated design modifications.

To address this gap, automating compliance checking and design adaptation first requires a theoretical basis to differentiate scenarios feasible for (semi-)automated adaptations from those not eligible for automation. Our objective is to develop a foundation for a framework to evaluate the applicability of automated design adaptation in cases of non-compliant designs by bridging

---

[1] https://www.rhino3d.com

between regulatory requirements, design characteristics, and parametric modeling techniques. As an initial effort, this study focuses on identifying key features from regulatory requirements that influence BIM-based adaptations. Specifically, it investigates the characteristics of textual rules in relation to parametric modifications based on existing ACC-related classification systems.

PROPOSED APPROACH

Based on the main ACC stages outlined by Eastman et al. (2009), Solihin and Eastman (2015) categorize rules into four classes to reflect their complexity in terms of implementability. The first and second classes mainly check explicit design entities, values, and simple derived attribute values, while complex cases require extended data structures (i.e., class 3) and may necessitate proof-of-solutions with multiple acceptable answers (i.e., class 4). Since streamlined design adaptation relies on automated checking as a prerequisite, we focus exclusively on checking tasks that can be implemented and verified within BIM models (i.e., classes 1, 2, and 3). Rules from IBC accessibility clauses are selected to demonstrate the feasibility of the proposed framework.

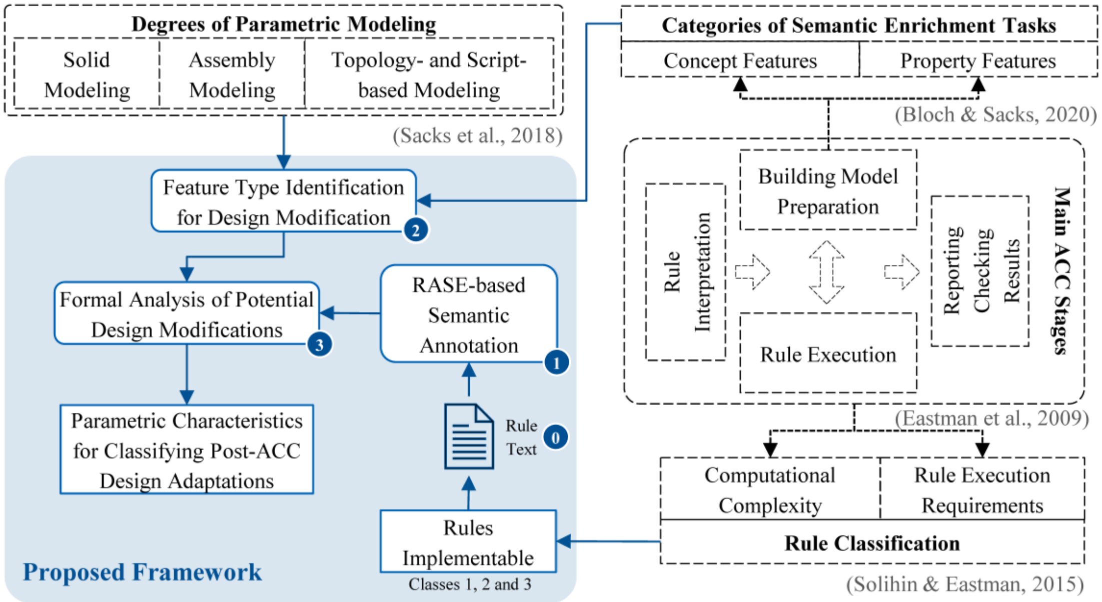


Figure 1. Proposed framework for linking ACC features to design modification processes (highlighted in blue), based on existing ACC classification and clustering methods.

The proposed framework is grounded in the main stages of ACC and varying degrees of parametric modeling (Sacks et al., 2018). We consider the characteristics of the individual ACC components: building regulations, denoted as 0 and 1; building design, expressed through SE tasks to support ACC, denoted as 2 (see **Figure 1**). For the regulation component, RASE-based analysis of building rules provides a robust foundation for identifying features to be investigated and modified for code compliance (Hjelseth & Nisbet, 2011). For the design component, the SE concept and property features identified by Bloch and Sacks (2020) are adapted to conceptualize parametric features for different design modifications. The key steps include: 1) use a RASE-based semantic annotation to extract checking objects and properties; 2) refine ACC feature types to align with the design modification perspective; and 3) perform a formal analysis of building rules to investigate required modifications and categorize the corresponding parametric characteristics.

RASE-based Semantic Annotation. RASE stands for Requirement, Applicability, Selection, and Exception and describes a formal way for making regulations computer-readable (Hjelseth & Nisbet, 2011). Requirement specifies the constraints to be met for the building design to be compliant, typically indicated by "shall" or "must". Applicability refers to the building element the requirement applies to. Therefore, it restricts the scope of the requirement. Selection widens the scope of the requirement. Exception marks building elements or specific cases in which the requirement does not apply, or an alternative requirement applies.

In practice, textual regulations are annotated with the RASE tags in the first step, and in the second step, the annotated phrases and sentences are enriched with information on objects, properties, comparators, and target values – which are required for the execution of the rules in an ACC system. For instance, RASE was utilized by Mendonça et al. (2020) to annotate accessibility regulations and implement the rules in the Solibri Model Checker[2]. In this paper, RASE was employed to analyze a set of accessibility requirements for their potential to support automated design adaptations. The RASE annotations decompose the requirements in their primary semantic units essential for compliance checking. Furthermore, it is hypothesized that the different RASE tags have varying influences on design adaptations. The RASE analysis is expected to identify the requirements that can be violated and their corresponding applicable building elements.

Feature Type Identification for Design Modification. This step serves as a bridge between the regulatory requirements and the actual building elements to which they apply. Identifying regulatory features reveals comparable aspects between ACC-oriented SE tasks and post-checking modification tasks, as they both focus on requirement-relevant target objects. The previously developed SE framework supporting ACC identifies two categories of SE tasks (Bloch & Sacks, 2020): concept-centered and property-centered tasks (see **Table 1**). Specifically, a concept refers to a building object or a group of interrelated objects subject to modification during design adaptations, while a property refers to a parameter defined for one or multiple objects that governs the geometric, semantic, or topological aspects of the design. A key difference between the SE and design modification perspective is the independent treatment of properties and concepts in the SE framework. In many cases, the concept already exists, and the SE task focuses on adding properties. For example, adding a missing room tag to a created room in a BIM is considered a SE task. In the same manner, design modification tasks adjust properties according to the scopes or relationships defined within the concepts. SE tasks generating an explicit representation of different concepts cannot be considered within design adaptation. An example of such SE tasks is generating a geometry to represent an egress path by aggregating the associated spaces. Thus, we focus on the property features identified in the SE framework, specifically those that drive various modifications through parameters associated with the objects (see **Table 1**).

---

[2] https://www.solibri.com

Table 1. Adjusted feature types for BIM-based design modification processes.

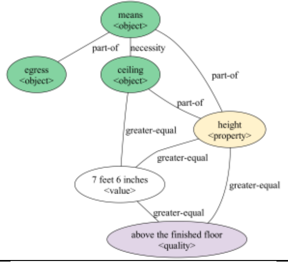
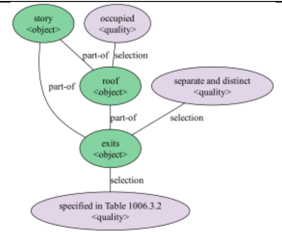
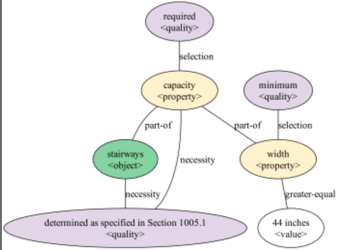
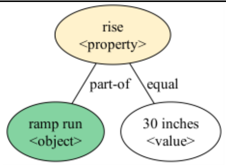
Semantic Enrichment Tasks (Bloch & Sacks, 2020)			**Design Modification Tasks**
Category	Feature	Enumerated Feature Values	(Adjusted) Enumerated Feature Values
Concepts	Element type	Physical (0), Abstract (1), Objectified relationship (2)	Not relevant
	Dependency type	Independent (0), Spatial (1), Connectivity (2), Access (3), Enclosure (4)	
	Composition type	Discrete (0), Aggregation (1), Assembly (2)	
	Association type	None (0), Abstract only (1), Physical only (2), Abstract or physical (3)	
Properties	Property type	Geometric (0), Functional (1), Relationship pointer (2), Value (3)	Geometric (0), Semantic (1), Design topological (2)
	Parent object type	Physical (0), Abstract (1)	Physical (0), Abstract (1)
	Composition type (of parent object)	Discrete (0), Aggregation (1), Assembly (2)	Discrete (0), Aggregation (1), Assembly (2)
	Dependency type (beyond the parent object)	No (0), Yes (1)	Independent (0), Spatial (1), Connectivity (2), Access (3), Enclosure (4)
	Value type	Numerical (0), Text (1), Enumerated value (2)	Numerical (0), Text (1), Enumerated value (2)

It should be noted that, in the SE context, tasks grouped under the property category may involve not only the properties of building elements but also aspects such as spatial placement, distances between objects, etc. Typically, these parameters ultimately represent properties assigned to the elements and the quantitative or qualitative relationships between these elements. Thus, the previously defined property features are adjusted based on 1) parametric modeling principles and 2) the distinction between SE and design modification tasks. First, the property types are regrouped into geometric, semantic, and topological classes, reflecting the fundamental aspects of parametric building design. Secondly, composition types are linked to those of the parent object, such as a discrete object (e.g., a door), an aggregation group (e.g., an apartment), or an assembly object (e.g., a multi-layer wall). Additionally, varying dependency types with other elements primarily involve design parameters at the multi-object or design topology levels (Sacks et al., 2018), such as spatial, connectivity, access, and enclosure aspects.

Formal Analysis of Potentially Required Design Modifications. A formal analysis is conducted to identify the parametric characteristics of required modifications to achieve code compliance. As shown in **Table 2**, building elements requiring changes are identified as Application and Selection through manual RASE annotation. For simple requirements, RASE-based annotations can help find associated properties through modification-related objects. However, for more complex rules, related objects and properties can constitute any of the RASE elements. For such cases, RoBERTa models, fine-tuned by ACCORD-NLP (ACCORD, 2024), were utilized to automatically extract entities (i.e., object, property, quality, and value) and classify their relations (e.g., comparators, such as "greater-equal"). These entities include the modifiable building elements (<object>) and their controlling parameters with corresponding values (<property>, <quality>, and <value>) and were represented using knowledge graphs. We manually investigated the required design modifications by explicitly focusing on the adjusted property feature types outlined in **Table 1**, with reference to building design domain knowledge. The required modifications are presented in a structured textual format. Furthermore, the corresponding parameter characteristics and their relationships, derived from the interpreted modification solutions, are summarized accordingly.

Taking IBC rule 1011.2 as an example, the generated knowledge graph indicates relevant building objects (stairway) and properties (width). Based on parametric characteristics, object-level modifications (e.g., stairway replacement or adjustment of related walls) or property-level modifications (e.g., geometric parameters of the stairway object) can be considered.

Table 2. Formal analysis of selected IBC accessibility rules and design modifications (on related objects and their properties) to determine the parametric characteristics.

IBC Rules with RASE Annotations	Automatically generated knowledge graph (ACCORD, 2024)	Required design modifications (manually interpreted)	Parametric characteristics (parameters & relationships)
1003.2 Ceiling height. The <a>means of egress shall have a <r>ceiling height not less than 7 feet 6 inches (2286 mm) above the finished floor</r>.		- Increase the ***ceiling height*** in the ***means of egress*** areas.	- Geometric parameters - Physical and abstract composition types
		- Reroute the ***means of egress*** through areas that meet the ceiling height requirement.	- Parameters in all property types - Physical and abstract relationships for both composition and dependency
1006.3.2 Egress based on occupant load. <s>Each story</s> and <a>occupied <s>roof</s> <r>shall have the minimum number of separate and distinct exits, or access to exits, as specified in Table 1006.3.2 </r>.		- Incorporate additional ***exits*** into the building.	- Geometric parameters - Physical dependency types
		- Relocate ***exits*** and their ***access routes***.	- Parameters in all property types - Physical and abstract relationships for both composition and dependency
1011.2 Width and capacity. The required <r>capacity</r> of <a>stairways shall <r> be determined as specified in Section 1005.1</r>, but the <r>minimum width</r> shall <r>be not less than 44 inches (1118mm) </r>.		- Increase the ***width*** of the ***stairways***.	- Geometric parameters - Physical dependency types
		- Replace the ***stairway*** family.	- Geometric and semantic parameters - Physical composition types
		- Add extra ***stairways*** or ***egress routes*** (and adjust related walls).	- Parameters in all property types - Physical and abstract relationships for both composition and dependency
1012.4 Vertical rise. The <r>rise>/r> for any <a>ramp run shall <r>be 30 inches (762 mm) maximum</r>.		- Replace the ***ramp*** family.	- Geometric and semantic parameters - Physical composition types
		- Reorganize or add extra ***ramps*** (and adjust related walls).	- Parameters in all property types - Physical dependency types

The summarized parametric characteristics (as shown in the last column of **Table 2**) form a structured basis for assessing and classifying parametric design adaptations by presenting both parameter types and their corresponding relationships. These characteristics are categorized into three parameter types (geometric, semantic, and topological) and different interrelationship levels (independent, composition, and dependency types). Additionally, incorporating both physical and abstract objects enhances their applicability to a wide range of design modification scenarios, with the latter (abstract objects) often involving more dependent relationships that extend beyond the parent object. Together, these overarching characteristics establish a foundation for evaluating necessary adaptations to achieve code compliance, often requiring multi-category changes to address diverse and complex design challenges effectively.

CONCLUSION

This paper explores the gap between compliance checking and design modifications to advance ACC-driven automated design adaptations. It proposes a structured framework for analyzing regulatory requirements from a new perspective: automated design adaptation. With RASE-based semantic annotation, the key modification-related features are identified from textual rules for parametric characteristic analysis. The framework bridges regulatory text, building design, and parametric modeling, all targeting a more intelligent workflow for design review that extends beyond a report of compliance issues. It connects rule characteristics across all three stages, enabling the identification of related objects and parameters requiring modifications in cases of non-compliance. The proposed approach examines regulatory requirements in terms of the applicability of needed automated modifications. At this stage, it focuses on defining the key property-specific features common to both compliance checks and design adaptations. The summarized parametric characteristics capture rule features, and the parametric nature of designs represented in BIM models. This provides a foundation for classifying building requirements based on the complexity and applicability of the implementable parametric design adaptations.

A future key focus will be a detailed examination of regulatory rules where ACC has been successfully implemented, aiming to develop an inclusive classification of regulatory requirements that are practically implementable, based on the parametric characteristics summarized in this work. Expanding beyond the RASE-based semantic annotations, further exploration of alternative methods for accurately extracting characteristics from complex building regulations is essential. This includes exploring and testing the feasibility of various natural language processing (NLP) techniques to manage the diverse structures and complexities of accessibility and other building regulations. Additionally, incorporating features associated with advanced generative designs, particularly those related to topology-based parametric modeling, has significant potential for automating complex design modifications. Finally, a limitation of the current approach is the determination of “required design modifications,” which relies on domain knowledge and manual interpretation. Addressing this limitation will involve developing a systematic framework to conceptualize and categorize fundamental change scenarios.

REFERENCES

ACCORD. (2024). *ACCORD-NLP: NLP framework to facilitate Automated Compliance Checking (ACC) within the Architecture, Engineering, and Construction (AEC) sector*. https://github.com/Accord-

Project/accord-nlp

Amor, R., & Dimyadi, J. (2021). The promise of automated compliance checking. *Developments in the Built Environment*, *5*(October 2020), 100039. https://doi.org/10.1016/j.dibe.2020.100039

Bloch, T., & Sacks, R. (2020). Clustering Information Types for Semantic Enrichment of Building Information Models to Support Automated Code Compliance Checking. *Journal of Computing in Civil Engineering*, *34*(6), 04020040. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000922

Borrmann, A., & Berkhahn, V. (2018). Principles of geometric modeling. In *Building Information Modeling: Technology Foundations and Industry Practice* (pp. 27–41). Springer International Publishing. https://doi.org/10.1007/978-3-319-92862-3_2

Eastman, C., Lee, J. min, Jeong, Y. suk, & Lee, J. kook. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Eastman, C., Parker, D. S., & Jeng, T. (1997). Managing the integrity of design data generated by multiple applications: The principle of patching. *Research in Engineering Design*, *9*(3), 125–145. https://doi.org/10.1007/BF01596599

Hjelseth, E. (2016). Classification of BIM-based Model checking concepts. *Journal of Information Technology in Construction*, *21*(November), 354–370.

Hjelseth, E., & Nisbet, N. (2011). Capturing Normative Constraints By Use of the Semantic Mark-Up Rase Methodology. *Proceedings of CIB*, *March*, 26–28.

International Code Council (ICC). (2021). *International Building Code 2018*. https://codes.iccsafe.org/content/IBC2018P6

Lee, P. C., Lo, T., Tian, M., & Long, D. (2019). An Efficient Design Support System based on Automatic Rule Checking and Case-based Reasoning. *Construction Management*, *23*, 1952–1962. https://doi.org/10.1007/s12205-019-1750-2

Liu, H., Cheng, J. C. P., Gan, V. J. L., & Zhou, S. (2022). A novel Data-Driven framework based on BIM and knowledge graph for automatic model auditing and Quantity Take-off. *Advanced Engineering Informatics*, *54*(September), 101757. https://doi.org/10.1016/j.aei.2022.101757

Mendonça, E. A., Manzione, L., & Hjelseth, E. (2020). Converting Brazilian Accessibility Standard for BIM-Based Code Checking using RASE and SMC. *CIB W78 Information Technology for Construction Conference*, *October 2021*, 291–307.

Pfeiffer, D., & Urban, H. (2024). BIM checking software requirements in the scope of the Vienna building authority. *Proc. of the CIB W78 Conference 2024*.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). Core Technologies and Software. In *BIM Handbook* (pp. 32–84). John Wiley & Sons, Ltd. https://doi.org/10.1002/9781119287568.CH2

Schulz, O., Oraskari, J., & Beetz, J. (2021). BcfOWL: A BIM collaboration ontology. *Proceedings of the 9th Linked Data in Architecture and Construction Workshop - LDAC2021*, *3081*, 142–153.

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*, 69–82. https://doi.org/10.1016/j.autcon.2015.03.003

Soliman-Junior, J., Tzortzopoulos, P., & Kagioglou, M. (2022). Designers' perspective on the use of automation to support regulatory compliance in healthcare building projects. *Construction Management and Economics*, *40*(2), 123–141. https://doi.org/10.1080/01446193.2021.2022176

Sydora, C., & Stroulia, E. (2020). Rule-based compliance checking and generative design for building interiors using BIM. *Automation in Construction*, *120*(July), 103368. https://doi.org/10.1016/j.autcon.2020.103368

Wu, J., Nousias, S., & Borrmann, A. (2025). Design Healing framework for automated code compliance. *Automation in Construction*, *171*. https://doi.org/10.1016/j.autcon.2025.106004

Yang, F., & Zhang, J. (2024). Prompt-based automation of building code information transformation for compliance checking. *Automation in Construction*, *168*(PA), 105817. https://doi.org/10.1016/j.autcon.2024.105817

Ying, H., & Sacks, R. (2024). From Automatic to Autonomous : A Large Language Model- driven Approach for Generic Building Compliance Checking. *Proc. of the CIB W78 Conference 2024*.