Model Healing: Toward a framework for building designs to achieve code compliance

Jiabin Wu, Rohit K. Dubey, Jimmy Abualdenien, André Borrmann
Chair of Computational Modeling and Simulation
Technical University of Munich, Munich, Germany

ABSTRACT: When non-compliance against building regulations occurs in construction projects, architects and engineers analyze the design and iteratively revise the model to overcome those violations. With the help of state-of-the-art code compliance checkers, practitioners interpret the results and adjust the design manually based on their expertise. Currently, no methodology is available to automatically connect the compliance violations to solving the related design issues. To address this gap, a generalized framework, `Model Healing`, is introduced for adjusting BIM models to overcome non-compliance. Violation-related parameters are selected to create comparable variants. We define metric indices representing the checking conformance and the difference to the initial design to support the search toward `Model Healing`. Finally, valid design variants close to the initial design are elected. The paper concludes with a case study indicating the applicability and limitations of the proposed workflow. This framework will facilitate building design adjustments and error correction with higher efficiency.

1 INTRODUCTION

With the broader adoption of information technologies in almost all the industrial sectors over the past several decades, various digital methods have also been remarkably explored to improve efficiency in the Architecture, Engineering, and Construction (AEC) industry. As one of the most complex tasks, building design processes are usually iterative and repetitive. The design needs to meet diverse requirements from owners and adapt to numerous uncertainties derived from the previous design decisions, which could cause inefficient and repetitive manual participation.

The emergence of Building Information Modeling (BIM) has provided a promising method for practitioners to cooperate efficiently and significantly pushed innovation in building design (Borrmann et al. 2018). BIM is defined as a modeling technology and associated set of processes to produce, communicate, and analyze building models (Eastman et al. 2011). Considering different design stages of building projects, one of the most crucial benefits of utilizing BIM technology is its accurate modelization. BIM technology modelizes fully parametric objects containing geometrical, semantic variants, and topological connectivity.

To guarantee building safety and environmental sustainability, building designers need to ensure that every aspect of the design meets the requirements of building regulations (Dimyadi and Amor 2013). The conventional checking procedure for building designs is time-consuming and error-prone (Eastman et al. 2009, Hjelseth and Nisbet 2010). Extensive research has been conducted in Automatic Code Compliance Checking (ACCC) (Ismail et al. 2017, Solihin et al. 2019, Zhang and El-Gohary 2017, Cornelius Preidel; and André Borrmann; 2017, Amor and Dimyadi 2021, Wu and Zhang 2022), searching for suitable means to evaluate the compliance of building models according to regulatory demands and additional requirements. The studies until today have led the community to a point where designers can employ commercial systems to verify building compliance against specified codes and standards.

A major limitation still existing is that the following design optimization is time-consuming since it requires manual involvement for iterative model adjustments, notwithstanding the instruction of failures from compliance checking. Feeding the compliance checking results back to the building design process to achieve compliant models has not been adequately researched so far.

We present a generalized framework, called `Model Healing`, to adjust non-compliant building designs such that they fulfill corresponding requirements. At the core of the concept, we make use of the adaptive nature of parametric building design. The initial design deficiencies are first identified by rule checking.

Then related building components are investigated, and adaptable parameters are selected to vary the initial design. Metric indices representing the conformance on rule checking and the dissimilarity to the initial design are created to support searching toward `Model Healing` within the adapted variants while keeping the deviation to the original design minimal.

In this paper, we first provide an overview of related work (Section 2), which includes concise reviews of model compliance checking, parametric BIM modeling, and solution space formation. We then introduce the methods for adapting the BIM models toward compliant designs (Section 3). In Section 4, we describe a case study based on a simple BIM model with which the proposed generalized methodology is evaluated, providing comparable and valid variants toward `Model Healing`. Finally, Section 5 presents concluding remarks and discusses future works.

2 RELATED WORK

2.1 *Model compliance checking*

Building codes are typically national legislation, where prescriptive requirements associated with every aspect of building design have been specified as applicable standards. Some ACCC studies employ building rules that correspond to fire safety requirements and other geometrical and spatial constraints (Preidel and Borrmann 2016).

Several systematic literature reviews focusing on ACCC were recently conducted (Ismail et al. 2017, Zhang and El-Gohary 2017, Solihin et al. 2019, Amor and Dimyadi 2021, Wu and Zhang 2022). The overall checking process comprises three parallel fields of research: rule interpretation, BIM data processing, and compliance checking execution (Eastman et al. 2009).

The most laborious part is transferring the regulatory information into computer-readable rules, which interprets the original building regulations. Diverse methods have been employed to rule interpretation. Here are some selected approaches: RASE semantic approach (Hjelseth and Nisbet 2010), logic programming (Etalle 1996), BERA domain-specific languages (Lee 2011), natural language processing (Zhang and El-Gohary 2016, Zhang and El-Gohary 2021). Besides, visual programming languages (VPLs) have been employed to improve the translation automation from standards and codes to machine- and human-readable languages (Preidel and Borrmann 2016). On the other hand, query languages are typically utilized to extract necessary design information for BIM data processing (Nawari 2012). Further rule-based and BIM-based compliance checking execution developments have been invested into commercial software, where the checking process is entirely accomplished (Eastman et al. 2009, Dimyadi et al. 2016, Ismail et al. 2017, Solihin et al. 2020).

Many research focuses on the automated processes of compliance checking (Preidel and Borrmann 2015, Nawari 2019). Nevertheless, their checking process lacks transparency for the users (Preidel and Borrmann 2015) and the checking results are not effectively re-utilized to automate the entire design process. For example, although Solibri Office demonstrated the feasibility of BIM-based compliance checking, it remains a “black-box” process without employing rule representation languages. Even with a mere pass or fail result and the failure reasons obtained from the existing compliance checkers, practitioners manually interpret checking results and search for feasible solutions based on their domain expertise and experience. This process is time-consuming and demands an iterative checking for compliance of design changes with the corresponding regulations and requirements.

Existing checking systems provide evaluation against specific codes and standards, while ACCC studies eventually aim to enhance the efficiency of the whole design process. Therefore, further investigations on connecting checking processes, such as standardizing the checking processes and supporting correct model generation, are necessary to leverage the full potential of ACCC (Amor and Dimyadi 2021).

2.2 *Parametric BIM modeling*

As the digital representation of a project over the building’s life cycle, BIM can also be described as “the technology of generating and managing a parametric model of a building” (Lee et al. 2006). Thus, parametric objects are central to understanding BIM components and differentiation from traditional 3D objects (Eastman et al. 2011).

Parametric design, a process between design intent and design response that utilizes internal parameters and rules to modify models, contributes significantly to advances in the AEC industries. A “parameter” is an input value for building design, such as geometric dimensions, characterizes a component, and helps establish interactive dependencies. The parametric design brings engineers dynamic control on BIM modeling, delivering flexible and adaptive models that can adjust to meet varying design requirements.

Parametric BIM models are typically defined by using dependencies and constraints (Borrmann et al. 2018). By creating parametrized object types (such as families), all the information such as locations, dimensions, and relationships of building elements become much more easily controlled. Thus, modifying the parameters embedded in a specific component family promotes the modification of the related family constituents, enabling architects and engineers to automate and optimize the building design. The essential benefit of employing parametric modeling, particularly for early building design, is to facilitate the creation of comparable solutions.

Advanced parametric modeling approaches provide software users and developers with an effective means to embed domain knowledge in BIM modeling (Lee et al. 2006). Many research efforts endeavored to optimize building design performance for given criteria using parametric BIM (Gerber et al. 2012, Brown et al. 2020), such as energy consumption. One of the most promising approaches is Generative Design (Ma et al. 2021), which helps designers create diverse designs efficiently regarding special design conditions and criteria (Granadeiro et al. 2013). Compared with the generative methodology that employs a customized programmatic approach, a variational methodology provides a more concurrent approach that varies the given model and is easier to apply (Shahin 2008). For example, the modification of geometry parameters is accomplished by variation in two processes: selecting the parameters to be changed and entering the new values (Lin et al. 1981).

2.3 *Solution space formation*

The solution space is a conceptual space that represents all potential solutions as vectors concerning a specific design problem. Solution space analytically interprets the uncertainties raised in engineering problems. Through analogical thinking from product design, this space can be initiated on engineering problems by considering the underlying problem (Gassmann and Zeschky 2008).

Regarding the application of solution space in building design, the potential solutions are expressed by vectors representing the combinations of the design parameters within the BIM model. The space considered contains multi-dimensional regions that can be expressed as the product of parameter intervals. The compliance of a building design normally depends on multiple interacting parameters, thus leading to a multiple-dimensional system. For high-dimensional problems with many relevant input parameters, a considerable number of samples is demanded to identify the solution space (Graff et al. 2016).

For each input parameter in the solution space, its interval should be engaged to represent sufficient possibilities of design variants. Qualified input parameters lay the foundation for a complete multi-dimensional region that comprises all perspective solutions regarding specified building requirements (Markus and Johannes 2012). Those parameters should be decoupled from each other to make the value of one parameter in the solution space independent of the value of the others (Graff et al. 2016), contributing to a vast solution space with high flexibility for building design. Hence, it is necessary to analyze constraints and dependencies within the BIM model to transform the building design information into consistent data for forming the solution spaces.

A good space comprises feasible designs that satisfy the problem's constraints. On the other hand, a bad space is formed by infeasible designs (Markus and Johannes 2012). The boundaries between the good and bad regions are probed by sampling within the whole space (Graff et al. 2016). In the case of disconnected feasible regions, the solution selected by further optimization might only be locally optimal.

Moreover, providing applicable optimization methods for target problems within the space requires an accurate description of the solution space based on systematic knowledge of BIM models' underlying constraints and dependencies. For instance, as the design goes on, similarity analysis based on the priority of building design parameters (Brown and Mueller 2019) becomes necessary to quantify and qualify the variation between various solutions among the whole design set. The raw "distance" between those designs indicates the similarity and the difference between variants (Anandan et al. 2006). In the early design stages, the solution space with low dimension metrics enables designers to interact with variables directly affecting building performance (Gassmann and Zeschky 2008).

3 THE MODEL ADAPTATION PROCESS

This research follows the principles of Design Science Research (Peffers et al. 2006) to develop solutions for automating building design. The proposed research investigates computational methods for parametric building design and BIM model representation in solution spaces to adapt non-compliant building designs automatically. The framework for automatic adaptation of building designs is illustrated in Figure 1. This framework is a preliminary step toward developing a solution that connects the workflow from the compliance checking results back to improving the building design to solve the issues identified by model checkers.

We define this adaptation process as `Model Healing`. `Model Healing` refers to the automatic adaptation of building models toward similar designs while fulfilling the selected requirements. Based on available model checkers and BIM parametrization, the essential goal of `Model Healing` is to effectively overcome the non-compliance regarding specific checking rules by adjusting the related parameters (e.g., the width of a corridor, the fire rating of a wall material) of the BIM model. The adaptation is achieved by representing the building design in suitable solution spaces where multicriterial searching is applied. There are three main steps in this proposed adaptation process, as described in the following sections.

Step 1: Compliance checking

The research builds on results provided by state-of-the-art model checkers. To avoid unconcerned infor-

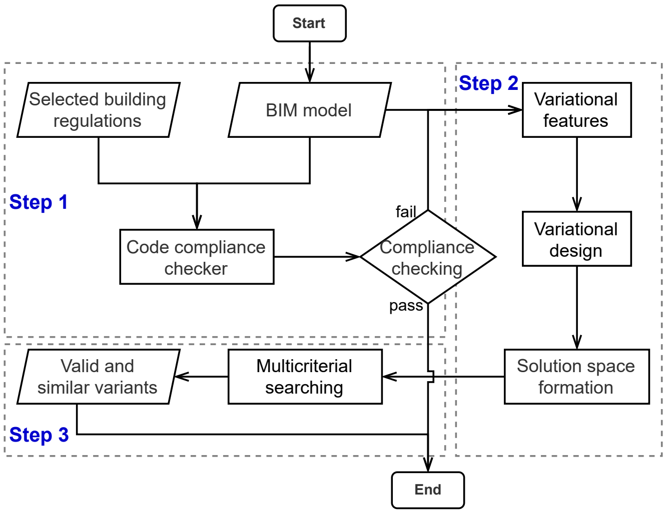


Figure 1: Illustration of the generalized `Model Healing` framework. Input: a BIM model and specific building regulations to comply with. Step 1: Compliance checking. Step 2: Solution space formation. Step 3: Searching toward healing. Output: A compliant BIM model that deviates the least from the input model (i.e., adhering to the original design).

mation and enhance the checking efficiency, we propose organizing the checking process into two steps: checking execution and failure localization.

The building design is evaluated by existing model checkers based on pre-implemented building rules. The model checkers investigate the model's compliance with specific building rules and provide the failure reasons if the checking fails. For example, in Solibri Office, rules-based checking results are presented, and the reasons for failure are provided in human-readable texts.

Uncomplicated descriptions of the failure reasons can be easily converted from explanatory texts (e.g., the horizontal distance between door and wall side $d_{example}$ is smaller than the minimum distance required d_{min}) into computer-readable formats ($d_{example} < d_{min}$). However, automatic conversion for various failure reasons is challenging to perform. On the other hand, the failure description in existing model checkers usually contains unconcerned information that hinders the extraction of the exact failure reasons. After identifying the failures within the Solibri Office API, we localize the "failed" components and related characteristics in the BIM model (e.g., via object identifiers IFC GUIDs), connecting the checking results to design adaptation.

Step 2: Solution space formation

Building designs regularly evolve through the design stages. Subsequent design decisions are significantly dominated by those made formerly. Most of the demands on systematical variations in BIM modeling are satisfied by utilizing dependencies and constraints. However, It is unnecessary to involve all building elements and their characteristics in creating a solution space regarding specific design problems. We collectively describe the design variables involved in the adaptation process as variational features. The term variational feature refers to design uncertainties such as objects' parameters, design constraints, and relationships in building design. We divide the variational features into semantic, geometrical, and topological features.

When violations occur, we identify related variational features and consider them as input parameters of the solution space. Given specific checking rules, only the features associated with the failure are considered the scope of the adaptation. It is vital to examine their priorities regarding the checking rules because the failure might involve multiple features in the BIM model. For example, checking the evacuation time from one room to the exit door of a building needs to consider multiple features: the overall story dimension, the relativeness with the adjacent rooms, and the location of the door.

Inspired by the definition of Building Development Level (BDL) (Abualdenien and Borrmann 2019), we propose to describe the variational features that represent the design uncertainties by hierarchical levels: story level, space level, and object level. Within this hierarchy, the model's degree of freedom continuously decreases while the design process is evolving. Feature changes influence other features on the same or lower levels without disturbing the features on the higher levels. For example, changing the position of an interior door (geometrical feature on object level) does not influence the functionality of the related space (semantic feature on space level). However, it might result in additional property-related failures because of different fire protection requirements (semantic feature on object level).

The solution space is formed by gathering alternative design variants in a conceptual space. Based on the concept of variational design, we vary the values of input parameters to create design variants. The values variation follows a two-sided truncated normal distribution (Robert 1995). The variation is doubly truncated by corresponding limit values captured by designers' expertise for each adaptable parameter. The original parameter values are employed as the location values for the variation. Afterward, varied designs are reevaluated by applying the compliance checking process. Not all varied input values will result in compliant design solutions, dividing the space into feasible and infeasible regions regarding specific checking rules.

Step 3: Searching toward healing

Notwithstanding that building designs are more complex than other products, the solution space can ease the problems in the early building design stages. The solution space supports investigating multicriterial searching around the initial design.

We propose a healing metric $\mathcal{H}(\mathcal{C},\mathcal{D})$ as an evaluation indicator during the adaptation process to select compliant variants that deviate the least from the initial design. The healing metric $\mathcal{H}$ consists of two components to quantify the variants' conformance regarding the checking rules and measure the variation from the initial design:

- Index of the distance $\mathcal{D}$: this index conveys the dissimilarity to the initially proposed design. We define a quasi-distance $\mathcal{D}$ as a weighted distance considering relevant parameters to ease quantifying variants' dissimilarity.

  The input parameters link to the variational features that violate the rules. First, the dissimilarity of geometrical features is easily understandable, such as the difference between the door height of variants. Secondly, the semantic features usually concern conditional or Boolean features (e.g., whether the door is an evacuation door). The topological features mainly influence the dependencies between features in the two former groups and the constraints in design. Afterward, the quasi-distance $\mathcal{D}$ is calculated by combining all the sub-distances as a "raw" distance to investigate the dissimilarity of variants within the solution space. Based on the end-users expertise in practice, weighting factors are assigned to each sub-attribute of $\mathcal{D}$ representing the difference between each input parameter.

- Index of the incompliance $\mathcal{C}$: $\mathcal{C}$ measures the design's compliance with the checked rules. Although $\mathcal{C}$ can probably be simplified to a pass/fail index, we propose quantifying the degree of violation instead of solely categorizing the variants as compliant and non-compliant groups. Therefore, $\mathcal{C}$ supports quantitatively describing variants' relative placements regarding the compliance criteria that need to be fulfilled. Especially when the adapted variants do not contain any compliant design, adjusting the $\mathcal{C}$ toward the compliance criteria guides us to improve the adaptation strategy. The analysis of $\mathcal{C}$ might facilitate the frameworks' application to intricate checking rules and design constraints.

Those two indices ($\mathcal{C}$,$\mathcal{D}$) serve respectively as a performance threshold and performance function during the multicriterial searching. For a given solution space Ω_{ss} containing every design variant $\mathbf{x}$, the proposed framework seeks such that

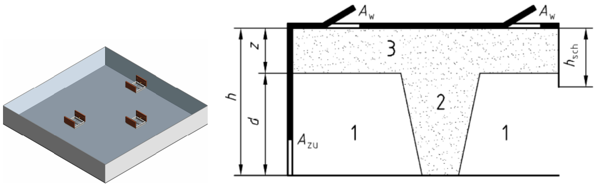
$$\mathcal{C}(\mathbf{x}) >= 0 \text{ for all } \mathbf{x} \in \Omega_{ss}, \ \mathcal{D}(\mathbf{x}) \to \min \qquad (1)$$

However, utilizing raw index values can lead to deviation in searching, especially for sub-attributes of $\mathcal{D}$ on different scales. Moreover, if the search for feasible solutions results in an empty set, the users should consider including input parameters with a more global impact and improve the adaptation methods, which might also involve the analysis of $\mathcal{C}$. Thus, transformation approaches are anticipated to conduce to metrics ($\mathcal{C}$, $\mathcal{D}$) on comparable scales. Finally, the search evolves toward the global maximum point symbolizing the closest valid variant during the adaptation, providing a compliant BIM model that deviates the least from the input model.

4 PROOF OF CONCEPT

4.1 *Scenario description*

In order to demonstrate the applicability of the `Model Healing` framework, a prototype of adaptive building design for the German standard DIN 18232-2 (DIN, 2007) is realized. This rule addresses security on room level in terms of smoke and fire protection, which ensures smoke is vented from escape routes. An academic one-space BIM model is created, which can be parametrically varied using the design tool Dynamo. The scenario is depicted in Figure 2.



(a) BIM model (b) Rule illustration

Figure 2: (a) An academic one-space BIM model, (b) Rule illustration: the minimal ventilation area for smoke extraction (German standard DIN 18232-11:2007 (DIN, 2007))

The minimal smoke ventilation area A_{min} in a room depends on room height h, the height of the fire smoke layer z, and the fire strength (Table 1).

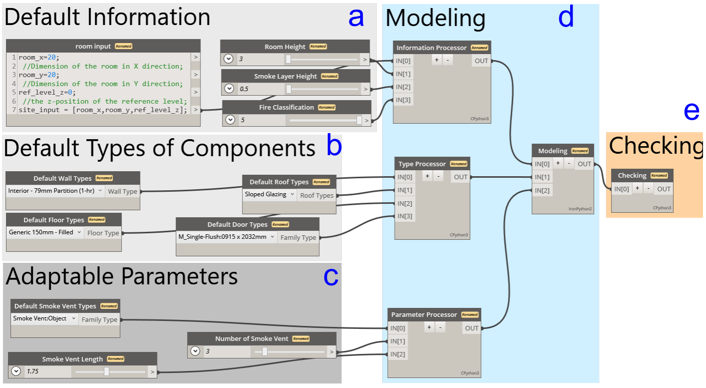
Table 1: Required smoke ventilation area (German standard DIN 18232-11:2007 (DIN, 2007))

Room Height [m]	**Height of the Smoke Layer [m]**	**Required Ventilation Area [m²]**				
		Fire Classification				
		1	**2**	**3**	**4**	**5**
3	0,5	4,8	6,2	8,2	11	15,4
3,5	1	3,4	4,4	5,8	7,8	10,9
	0,5	3,0	8,7	11,3	15	20,4
4	1,5	2,5	3,6	4,7	6,4	8,9
	1	3,0	6,2	8	10,6	14,4
4,5	2	2,5	3,1	4,1	5,5	7,7
	1,5	3,0	5	6,5	8,7	11,8
	1	3,5	8,4	10,7	13,9	18,6

4.2 *Proposal Application*

4.2.1 *Compliance checking*

The compliance checking examines the minimal ventilation area and solely considers this evaluation criterion. Due to the simplicity of this one-space model, we developed a checking algorithm to evaluate the compliance instead of taking existing model checkers for model evaluation. After the initialization of the design, compliance checking is executed following the overall workflow (Figure 3).



```
room_x=20;
 //Dimension of the room in X direction;
room_y=20;
 //Dimension of the room in Y direction;
ref_level_z=0;
 //the z-position of the reference level;
site_input = [room_x,room_y,ref_level_z];
```

Figure 3: Fragment of Dynamo script for `Model Healing`: a,b) primary input, c) adaptable input regarding the checking rule, d) initial modelization, and e) rule checking on the initial design

4.2.2 *Solution space formation*

Compared with complex BIM models from practice, this one-space model contains evident variational features related to the checking rule: objects (number and location of smoke vent) and their characteristics (type and dimensions of smoke vent).

In the proposed framework, the end-users determine the adaptable parameters from relevant variational features associated with the checking failures based on their understanding of the overall design process and corresponding constraints. In the investigated scenario, the room dimensions and the fire classification to resist are considered to be fixed. Therefore, the input parameters of the solution space can be easily elected among all corresponding variational features in Table 2. We assume the smoke vent in square shapes and utilize identical dimensions for every smoke vent according to practical experience. This simplification filters the solution space into a multiple-dimensional space with two adaptable input parameters: the number of the smoke vent n and the length of each square smoke vent l.

Table 2: Relevant features of the initial design

Parameter	h	z	fire strength	n	l
Value	$3.0\ m$	$0.5\ m$	Class 5	3	$1.75\ m$

After being evaluated by the checking algorithm, the initial design shows that it did not meet the requirement on minimum smoke ventilation area A_{min}. Based on practical experience, we vary the input parameters with comparable values following a two-sided truncated normal distribution. We fix the smoke vent family with limited length l in $[1\ m, 3\ m]$ and the number of smoke vent n in $[1, 9]$. Afterward, the built-in checking algorithm is reapplied to all adapted designs.

4.2.3 *Multicriterial searching toward Model Healing*

The two indices $(\mathcal{C}, \mathcal{D})$ of the healing metric for the i^{th} adapted design are expressed in Equation 2 and 3.

$$\mathcal{C}_i = n_i * l_i^2 - A_{min}, (i \in N^*) \tag{2}$$

$$\mathcal{D}_i = \sqrt{\left(\frac{n_i}{n_{init}} - 1\right)^2 + \left(\frac{l_i}{l_{init}} - 1\right)^2} \tag{3}$$

where n_i, n_{init} represent the number of the smoke vent of the i^{th} adapted design and the initial design, and l_i, l_{init} are the length of each square smoke vent of the i^{th} adapted design and the initial design.

Since the original "distance" on input parameters (n, l) are on different scales, a preliminary transformation is adopted to calculate the weighted quasi-distance $\mathcal{D}$ (Equation 3). Moreover, a logarithmic transformation function f_{log} is employed to moderate the distribution irregularity by keeping their original signs. The min-max normalization coefficients (α,β) in f_{norm} are captured from the non-negative values to convert positive values into $[0, 1]$, facilitating investigating the feasible region ($\mathcal{C}_i >= 0$) where the designs comply with the checking rules. f_{log} and f_{norm} for $\mathcal{D}$ are expressed in Equation 4 and 5, and same transformations are applied to $\mathcal{C}$.

$$f_{log}(\mathcal{D}_i) = \begin{cases} \log\left(\mathcal{D}_i + 1\right), & (\text{if } \mathcal{D}_i >= 0) \\ -\log\left(-\mathcal{D}_i + 1\right), & (\text{if } \mathcal{D}_i < 0) \end{cases} \tag{4}$$

$$f_{norm}(\mathcal{D}_i) = \frac{\mathcal{D}_i - \beta}{\alpha} \tag{5}$$

where $\alpha = \max(\mathcal{D}_1,\ldots,\mathcal{D}_i)$, $\beta = 0$, $i \in N^*$.

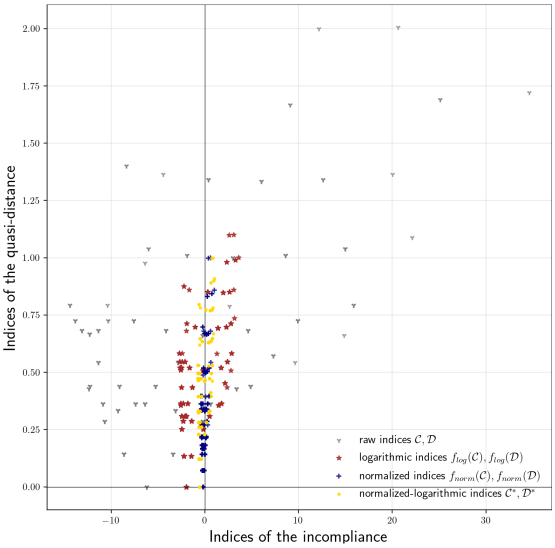


Figure 4: Illustration of the raw indices and other transformed indices

The raw indices and transformed indices are illustrated in Figure 4. In this prototype, the normalized-logarithmic indices $(\mathcal{C}_i^*, \mathcal{D}_i^*)$ are adopted.

$$\mathcal{C}_i^* = f_{norm}\left[f_{log}(\mathcal{C}_i)\right] = \frac{f_{log}(\mathcal{C}_i) - \beta_{log,\mathcal{C}}}{\alpha_{log,\mathcal{C}}} \tag{6}$$

$$\mathcal{D}_i^* = f_{norm}\left[f_{log}(\mathcal{D}_i)\right] = \frac{f_{log}(\mathcal{D}_i) - \beta_{log,\mathcal{D}}}{\alpha_{log,\mathcal{D}}} \tag{7}$$

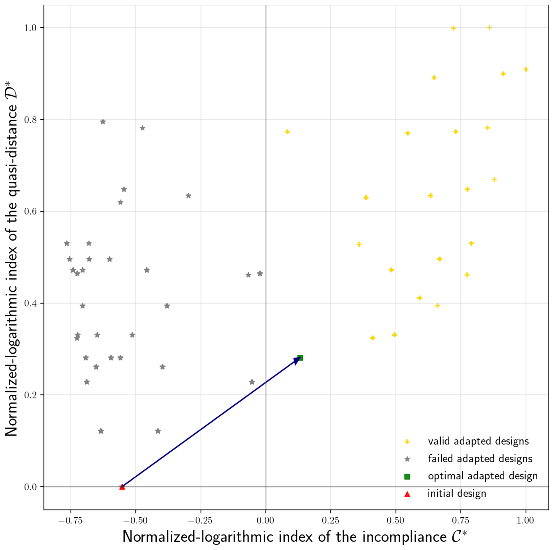


Figure 5: The results of searching via normalized-logarithmic indices $(\mathcal{C}^*, \mathcal{D}^*)$

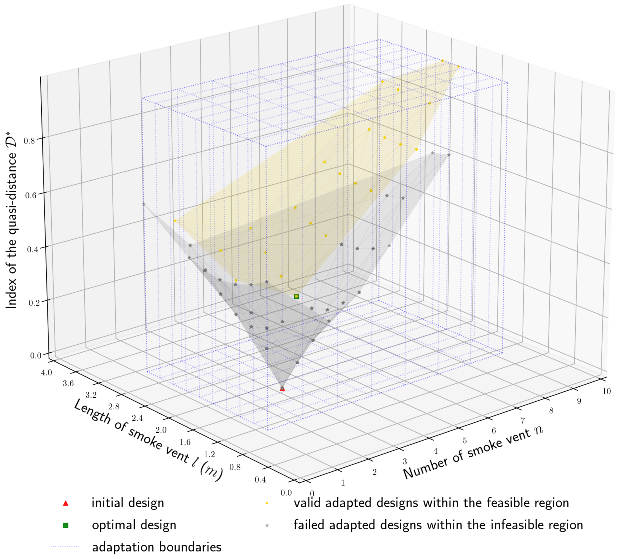


Figure 6: Solution space concerning the normalized-logarithmic indices of quasi-distance $\mathcal{D}$: (n, l)

where $\alpha_{log,\mathcal{C}} = \max\left[f_{log}(\mathcal{C}_1), \ldots, f_{log}(\mathcal{C}_i)\right]$, $\alpha_{log,\mathcal{D}} = \max\left[f_{log}(\mathcal{D}_1), \ldots, f_{log}(\mathcal{D}_i)\right]$, $\beta_{log,\mathcal{C}} = \beta_{log,\mathcal{D}} = 0$, $i \in N^*$.

The optimal design is selected according to the performance threshold and performance function in Equation 1, leading to compliant designs that deviate the least from the initial design (Figure 5). The solution space concerning $\mathcal{D}$ is described in Figure 6, depicting the feasible and infeasible regions within the overall adaptation boundaries.

The results achieved `Model Healing` by selecting the closest feasible option within the solution space. Those adapted options far from the initial design (in Figure 5) have a significantly different length or number of smoke vents. The initial design and optimal design selected from the adaptation considered the final solution for this experiment are illustrated in Figure 7. In this way, the initial model was "healed" to a compliant design by slightly enlarging the dimensions of smoke vents and adding one more smoke vent in the ceiling. The requirement of building codes is satisfied, and the initial design trend has also been kept.

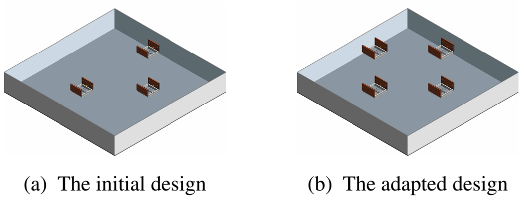

(a) The initial design (b) The adapted design

Figure 7: 3D views: (a) the initial design, (b) the adapted design

5 CONCLUSION AND FUTURE WORKS

This paper proposed a novel framework for automatic building design adaptation for non-compliant building designs. It addressed the gap that code compliance checking and automated design improvement have not been conjointly investigated to automate the design process.

The proposed framework investigates the rule checking results and selects adaptable parameters from design features associated to non-conformance. Within the framework, the compliance violation is considered the improvement target, which eases the design improvement from iterative processes. Based on the variational design, comparable variants are created by varying input parameters within the solution space. Metric indices representing the checking conformance and dissimilarity to the initial design are calculated to support model evaluation. The searching among adapted designs finally leads to a compliant BIM model that deviates the least from the input model. An experiment on German smoke and fire protection regulation shows that the proposed framework is applicable. This paper addresses the existing gap in interpreting rule checking results to support automatic building design adjustment.

We acknowledge the following limitations and corresponding future works:

- Industrial BIM models comprising enormous components will result in high-dimensional spaces, reducing the model representation effectiveness and hampering the broader adoption of this framework. The proposed variational features and hierarchical levels need further investigation to help select feasible input parameters of the solution space for practice building designs concerning specific checking rules.

- The scope of the experiment is limited. Employing an academic model simplifies the failure checks and solution space formation. Built-in algorithms temporarily undertook model checking. Nevertheless, challenges might be revealed when the checking demands expand to undeveloped checking rulesets in model checkers. Further experiments on industrial-practice BIM models and practical model checkers will expand the framework's application.

6 ACKNOWLEDGEMENTS

This research was funded by the International Graduate School of Science and Engineering (IGSSE) / TUM Graduate School.

REFERENCES

Abualdenien, J. & A. Borrmann (2019). A meta-model approach for formal specification and consistent management of multi-lod building models. *Advanced Engineering Informatics 40*, 135–153.

Amor, R. & J. Dimyadi (2021). The promise of automated compliance checking. *Developments in the Built Environment 5*(October 2020), 100039.

Anandan, S., S. Teegavarapu, & J. D. Summers (2006). Issues of similarity in engineering design. *Proceedings of the ASME Design Engineering Technical Conference 2006*, 1–10.

Borrmann, A., M. König, C. Koch, & J. Beetz (2018). *Building information modeling: Technology foundations and industry practice*.

Brown, N. C., V. Jusiega, & C. T. Mueller (2020). Implementing data-driven parametric building design with a flexible toolbox. *Automation in Construction 118*.

Brown, N. C. & C. T. Mueller (2019). Quantifying diversity in parametric design: A comparison of possible metrics. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM 33*(1), 40–53.

Cornelius Preidel; & André Borrmann; (2017). Refinement of the Visual Code Checking Language for an Automated Checking of Building Information Models Regarding Applicable Regulations. *Congress on Computing in Civil Engineering, Proceedings (2017)*, 157–165.

Dimyadi, J. & R. Amor (2013). Automated building code compliance checking - where is it at? *Proceedings of the CIB World Building Congress 2013 and Architectural Management & Integrated Design and Delivery Solutions (AMIDDS)* (380), 172–185.

Dimyadi, J., R. Amor, & M. Spearpoint (2016). Using bim to support simulation of compliant building evacuation. *eWork and eBusiness in Architecture, Engineering and Construction - Proceedings of the 11th European Conference on Product and Process Modelling, ECPPM 2016* (September), 511–520.

Dynamo. `https://dynamobim.org/`. Accessed: 2022-03-20.

Eastman, C., J. m. Lee, Y. s. Jeong, & J. k. Lee (2009). Automatic rule-based checking of building designs. *Automation in Construction 18*(8), 1011–1033.

Eastman, C., P. Teicholz, R. Sack, & K. Liston (2011). *BIM Handbook, A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, Second Edition*.

Etalle, S. (1996). An introduction to logic programming through prolog michael spivey, prentice hall international series in computer science, englewood cliffs, nj, 1996. isbn 0-13-536047-1. *Science of Computer Programming*.

Gassmann, O. & M. Zeschky (2008). Opening up the solution space: The role of analogical thinking for breakthrough product innovation. *Creativity and Innovation Management 17*(2), 97–106.

Gerber, D. J., S. H. Lin, B. Pan, & A. S. Solmaz (2012). Design optioneering: Multi-disciplinary design optimization through parameterization, domain integration and automation of a genetic algorithm. *Simulation Series 44*(8 BOOK), 33–40.

Graff, L., H. Harbrecht, & M. Zimmermann (2016). On the computation of solution spaces in high dimensions. *Structural and Multidisciplinary Optimization 54*(4), 811–829.

Granadeiro, V., J. P. Duarte, J. R. Correia, & V. M. Leal (2013). Building envelope shape design in early stages of the design process: Integrating architectural design systems and energy simulation. *Automation in Construction 32*, 196–209.

Hjelseth, E. & N. Nisbet (2010). Exploring semantic based model checking. *Proceedings of CIB W78 Conference 27*, 16–18.

Ismail, A. S., K. N. Ali, & N. A. Iahad (2017). A review on bim-based automated code compliance checking system. *International Conference on Research and Innovation in Information Systems, ICRIIS*.

Lee, G., R. Sacks, & C. M. Eastman (2006). Specifying parametric building object behavior (bob) for a building information modeling system. *Automation in Construction 15*(6), 758–776.

Lee, J. k. (2011). Building environment rule and analysis (bera) language and its application for evaluating building circulation and spatial program.

Lin, V. C., D. C. Gossard, & R. A. Light (1981). Variational geometry in computer-aided design. *Proceedings of the 8th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1981 15*(3), 171–177.

Ma, W., X. Wang, J. Wang, X. Xiang, & J. Sun (2021). Generative design in building information modelling (bim): Approaches and requirements. *Sensors 21*(16).

Markus, Z. & E. v. H. Johannes (2012). Computing solution spaces for robust design. *International* (February), 1102–1119.

Nawari, N. O. (2012). Automating codes conformance. *Journal of Architectural Engineering 18*(4), 315–323.

Nawari, N. O. (2019). A generalized adaptive framework (gaf) for automating code compliance checking. *Buildings 9*(4).

Peffers, A., E. Charles, J. The, M. Rossi, & W. Hui (2006). The design science research process. *17028071*, 83–106.

Preidel, C. & A. Borrmann (2015). Automated code compliance checking based on a visual language and building information modeling. *32nd International Symposium on Automation and Robotics in Construction and Mining: Connected to the Future, Proceedings*.

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

Robert, C. P. (1995). Simulation of truncated normal variables. *Statistics and Computing 5*(2), 121–125.

Shahin, T. M. (2008). Feature-based design - An overview. *Computer-Aided Design and Applications 5*(5), 639–653.

Solibri Office. `https://www.solibri.com/solibri-office`. Accessed: 2022-03-20.

Solihin, W., J. Dimyadi, & Y. Lee (2019). In search of open and practical language-driven bim-based automated rule checking systems. *Advances in Informatics and Computing in Civil and Construction Engineering* (January).

Solihin, W., J. Dimyadi, Y. C. Lee, C. Eastman, & R. Amor (2020). Simplified schema queries for supporting bim-based rule-checking applications. *Automation in Construction 117*(March), 103248.

Wu, J. & J. Zhang (2022). Model Validation Using Invariant Signatures and Logic-Based Inference for Automated Building Code Compliance Checking. *Journal of Computing in Civil Engineering 36*(3).

Zhang, J. & N. M. El-Gohary (2016). Semantic nlp-based information extraction from construction regulatory documents for automated compliance checking. *Journal of Computing in Civil Engineering 30*(2), 04015014.

Zhang, J. & N. M. El-Gohary (2017). Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Automation in Construction 73*, 45–57.

Zhang, R. & N. El-Gohary (2021). A deep neural network-based method for deep information extraction using transfer learning strategies to support automated compliance checking. *Automation in Construction 132*(October 2020), 103834.