Enriching Building Graphs with Parametric Design Constraints for Automated Design Adaptation

Conference Paper · September 2023

DOI: 10.13154/294-10116

2 authors:

Jiabin Wu
Technische Universität München
4 PUBLICATIONS **6** CITATIONS

Jonas Schlenger
Technische Universität München
12 PUBLICATIONS **15** CITATIONS

Enriching Building Graphs with Parametric Design Constraints for Automated Design Adaptation

Jiabin Wu[1*] and Jonas Schlenger[1*]
[1]Chair of Computational Modeling and Simulation, Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany
E-mail(s): j.wu@tum.de, jonas.schlenger@tum.de
*all authors have contributed equally

Abstract: Building designing is an iterative process of developing design concepts while fulfilling various requirements. Design parameters, dependencies, and constraints are embedded in the BIM-based design environment to support automated design adaptation techniques. However, only a small part of design constraints is explicitly represented in the digital models as design and engineering knowledge, and most studies focus on constraints on single object levels. To address this issue, this paper presents a workflow for enriching building knowledge graphs with design-oriented constraints. This research aims to extract constraints through embedded design parameters automatically. Data retrieval queries and analyses for model constraints are accomplished based on the extracted RDF graph that represents the intended building topology. Maintaining the users' design intent and obeying the consistency constraints, the graph-based approach dynamically computes the range of design parameters potentially associated with the requirement constraint fulfillment. Due to the graph structure, cascading effects of element displacements can be considered on various levels of adjacency.

Keywords: BIM, Parametric Modeling, RDF Graph, Design Constraints

Erschienen in Tagungsband 34. Forum Bauinformatik 2023, Bochum, Deutschland, *DOI:* will/be/assigned/by/editor

1 Introduction

The introduction of Building Information Modeling (BIM) related technologies has significantly changed the design process in the Architecture, Engineering, Construction, and Operation (AECO) industry. BIM technologies allow explicitly representing and managing various design information activities. Based on parametric dependencies and constraints [1], BIM supports the transformation of complex geometries to create variations of the same basic form for detailed design analyses with little effort.

In the context of parametric modeling, constraints refer to limitations or restrictions imposed on the design model upon specific properties of building elements. User requirements, building regulations, and construction rules imply constraints on a building design [2]. In practice, checking digital building models for compliance with constraints in different aspects allows for adapting the model toward compliant designs in an early stage. Parametric design constraints form an important intermediate connection between the building properties encoded in the model and how humans reason about the building elements and their relations. This ensures the model maintains the specified structure from the design intent and the regulations.

Nevertheless, fully realizing these benefits demands specialized parametrization setup, building elements with functional behavior, and appropriate management of elements in the BIM model. Only a small part of design knowledge is explicitly integrated into digital models as design constraints [3]. Manually integrating constraints with building models requires intensive modeling efforts by designers, especially for arbitrary geometric constraints that involve multiple building elements. Automated BIM-based design analyses are usually interrupted when inconsistencies occur during adaptation. Therefore, BIM-based parametric modeling must consider and quantify the potentially entangled constraints to avoid constraint violation issues. A parametric representation of the design constraints is crucial for designers to work with parameters and cope with the complex relationships between building elements.

To address this need, this paper proposes a graph-based approach for parametric constraint representation in early building design. A design constraint kernel is developed to represent existing design constraints with Resources Description Framework (RDF) graphs. As the starting point, existing data converters are used to transform IFC models into simplified RDF graphs using the Building Topology Ontology (BOT). The baseline graphs are enriched with details about parametric constraints by automatically extracting additional information from the design environment combined with arithmetic operations. The topological information from the BIM model allows efficient inferring of the relationships between building elements and parameter-based properties. The enriched graph, one of this paper's primary outcomes, represents quantitative consistency design constraints for maintaining the initially proposed design topology. This procedure provides a robust strategy to determine the permissible ranges of parameter values, especially for spatial constraints in design processes.

A case study is conducted on a multi-storey office building to represent both generic consistency and project-specific requirement constraints. Certain spatial constraints, e.g., wall movement constraints, are explicitly linked to related parameters and quantified as permissible ranges. The constraint identification and representation could be ameliorated with specific domain knowledge or project-related design and construction information, rendering varied. The observed parameter ranges will facilitate building design adjustments and improve the efficiency of automated design analyses.

The rest of the article is organized as follows: Section 2 presents state-of-the-art methods and related works. Section 3 describes the proposed graph-based approach. Section 4 is dedicated to the experimental setup and results. The outcome of this approach is discussed in Section 5 and Section 6 finally concludes the article.

2 Related works

BIM technologies enable designers to generate and manipulate building models by changing model parameters like storey height or wall thickness. In contrast to rough geometric designs, parametric designs define the essential features by object-oriented parameters. Lee, Sacks, and Eastman [4] explore the building object behavior (BOB) description as a means to embed design intent and knowledge in parametric elements. Parametric BIM models are typically defined using dependencies and constraints [1]. By creating parametrized element types, all the information, such as locations, dimensions, and relationships, can be easier controlled. Modifying the parameters of a specific component family promotes the modification of all the related family constituents, enabling architects and engineers to automate and optimize the building design processes. With advanced support of parametric design, designers can generate model variations by changing parametric values [5].

Building design constraints can be distinguished into two fundamental types: consistency constraints and requirement constraints [2]. Consistency constraints are typically embedded in the BIM model, including geometric and topological relationships, while requirement constraints derive from building regulations, best-practice construction rules, and client requirements. Conceptual spatial constraints might restrict the parametric model. Thus, parametric modeling systems typically use standard sets of geometric constraints that consist of spatial relations between geometric objects, e.g., collinearity, horizontal alignment, and perpendicularity [6]. The supported geometric constraints are usually defined by fixed numbers [7]. The correct application of constraints toward a satisfactory solution is complex [5] since design conflicts exist between contradictory constraints.

In their literature review, Pauwels, Zhang, and Lee [8] present the state-of-the-art in applying regulation compliance checking on RDF graphs. Tang, Zou, Feng, *et al.* [9] comprehensively review the geometric constraint-solving problem (GCSP) in parametric designs. Arora, Bielski, Eisenstadt, *et al.* [10] use graph-based representation and developed a constraint-based evaluator for the coherency of semantic spatial configurations. Kirchner and Huhnt [7] employ a constraint graph to extend the geometric model to store intended equation constraints between two connected walls. To solve the GSCP, Kirchner [3] proposes investigating the relationships between points, constraints, and parameters for computing the possible ranges of extruded rectangular walls. However, existing design constraint studies focus on object-level topological relations in parametric models. The previous approaches lack the ability to support automated design optimization processes. Formalizing design constraints as a user-oriented and computer-interpretable format is essential to improve design coordination.

3 Methodology

The proposed methodology investigates the building design constraints and enriches the building knowledge graph with additional design-oriented information. This graph-based approach determines the range of possible values for every location-related parameter to maintain consistency constraints (geometric and topological relationships). With the identified consistency constraints, the ease of use of such an enriched parametric model renders designers more adaptable for fulfilling other particular requirement constraints (building regulations, construction rules, and client requirements).

The starting point is a parametric design model that resides within a specific design environment. Even though the aim is to extract all required information from a vendor-neutral format like IFC to have a generic pipeline, some required information is not stored in the IFC file or is more easily accessible in the native format. In the case of the present paper, Autodesk Revit is used with its API to access design parameters, topological information, and custom attributes. A graph-based approach was chosen to store and connect all required information. It allows for capturing the strongly interlinked nature of the building structure, the topological relations between building elements, and the design parameters suitably. Furthermore, design change propagation across various depths of element and room neighborhoods is well-reflected in the graph structure.

The complete methodology is displayed in the workflow diagram in fig. 1. All procedural steps are represented with circles. Additionally, their numbers in black refer to the order in which they are executed. It is also referred to these numbers in the following paragraph.

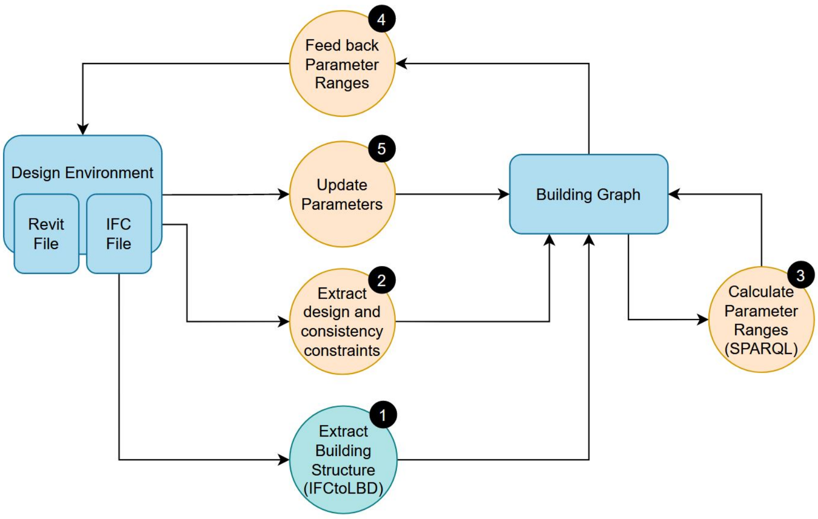

Figure 1: Workflow diagram depicting the proposed methodology

1. The IFCtoLBD Converter developed by [11] is applied to extract the building structure from the IFC file and create an RDF graph using BOT.
2. Information not explicitly represented in the IFC file is extracted through the Revit API. This includes design constraints, e.g., attributes that describe the structural relevance of walls, topological relationships between rooms and elements, and design parameters that define the position of the building elements in relation to reference grids. The graph resulting from step one is enriched with this information using Python and the rdflib.
3. Based on the enriched graph, the range of possible movement of building elements is calculated. The positions of the elements are described through parameters referring to related grids. The range of movement is calculated with the help of SPARQL queries.

4. The movement ranges which are allowed to comply with the consistency constraints are sent back to the design environment to support the user in subsequent design processes.
5. Once changes are applied to the design model, the location-related parameters are updated in the building graph. From this stage, the pipeline is rerun starting from step number three.

4 Case Study

The methodology proposed in section 3 was tested on a design model in Autodesk Revit representing an office building with two stories. The model is enriched with design parameters, which describe the position of walls, windows, and doors in relation to 2D reference grids (fig. 2). In the scope of the case study, the range of motion of all walls was investigated by calculating distance-related parameter ranges for every wall element. Walls are the main elements that dictate a building's design. The same methodology can be applied to other elements, like columns, doors, and windows.

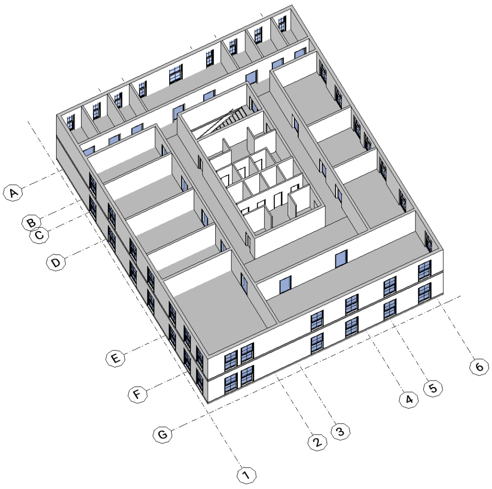

Figure 2: 3D view of the design model

Figure 3: The unfulfilled requirement and related neighbor elements

Following the methodology described in section 3, the IFC file of the building was converted using the IFCtoLBD converter to obtain an RDF graph depicting the fundamental structure of the building. Additional information was extracted from Revit through the Revit API to complement the data from the IFC file. This comprises attributes of walls describing if they are structural, external, part of the stairway, or none of the three and characteristics of doors differentiating between building exits, storey exits, and usual doors to transition from one room to the next. Moreover, the topological relations of all building elements and rooms were used to enrich the building graph. Finally, the distance-related design parameters in relation to 2D reference grids were automatically added to the graph to finalize the knowledge transfer from the design environment to the graph database. Taking the design constraints into account while at the same time maintaining the topological structure of the building, the displacement ranges of walls were calculated through a sequence of SPARQL requests. Finally, obtained values are stored in the graph, again using SPARQL.

An example demonstrates the information stored in the final building graph and how it can support automated design optimization processes. The room on the top left in fig. 3 was identified as not complying with specific building regulations (e.g., the International Building Code) due to too small room area, thus demanding wall adjustment to find a compliant building design. Since the walls on the top and the left are external, they cannot be moved according to typical requirement constraints once the building outline is determined. The other two walls and adjacent rooms need to be considered.

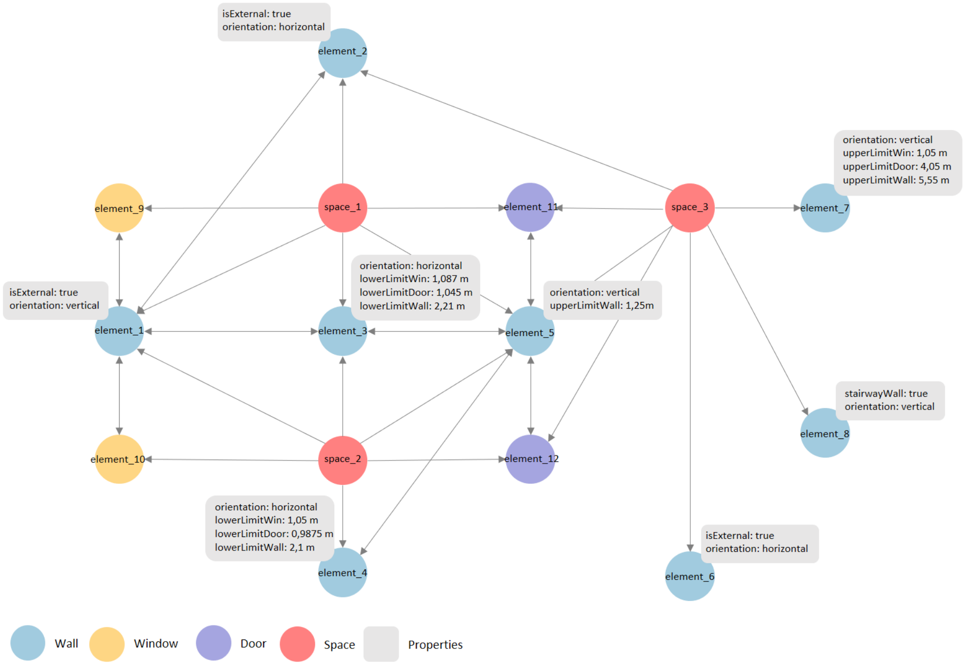

Figure 4: A subsection of the building graph showing the relevant nodes in the example case

The possible wall displacement is calculated for the two mentioned walls, maintaining the building topology and avoiding clashes with other walls, windows, or doors. These different types of limits for displacement are calculated through SPARQL queries and thereafter stored in the graph. The doors and windows are considered fixed or still adjustable, depending on the design requirement constraints. If necessary, rooms that are second-level neighbors of the initial room can also be considered to check for allowed displacement ranges. The stairway wall is a second-level neighbor of the initial office room in the example. It is considered fixed according to typical evacuation-related requirement constraints that specify stairway placement. Consequently, the corridor can not be widened to the right by narrowing the staircase. The room to the top of the stairs has no constraints and can be shrunk to allow for more space for the initial non-compliant office room.

The related section of the building graph that includes the primarily involved elements is displayed in fig. 4. It displays the topological relations between elements and space, which are indispensable for the calculations. Furthermore, a selection of node attributes is presented. With the provided displacement limits, the design changes can be accomplished via particular optimization algorithms or by users with domain expertise directly. Once the design is improved, the graph is updated with related changes and analyzed again for computing subsequent displacement limits.

5 Discussion

The proposed approach can adaptively provide parametric constraints as design variation ranges for automated design adaptation. The novelties of this approach compared to existing research are summarized here. First, design and engineering knowledge are explicitly investigated via the nodes and edges of the graph structure as consistency and requirement constraints. With graph-based queries and analyses for model constraints, the proposed workflow provides a dynamic way to fulfill requirement constraints by maintaining customized consistency constraints. Second, the proposed approach can be applied to other building elements and provide permissible variation ranges for all the building elements, which is crucial for various parametrization-based design optimization studies.

This study still has limits despite these contributions. First, the provided displacement changes are limited without considering combinatorial ranges for multiple parameters. Second, requirement constraints have been widely considered in previous code compliance checking studies. By investigating those requirement constraints in building regulation, the approach can integrate code-compliant ranges for parameters as design guidance. However, the initial building design topology might need to be improved to fulfill those requirement constraints, leading to more expansive graph rewriting techniques. Third, current query implementation is only tested on building elements that comply with the Manhattan world assumption. Increased computational cost is expected for more complex elements.

6 Conclusion and Outlook

This paper describes a methodology to dynamically provide the ranges of location-related parameters by maintaining the initial design intent from the user. Instead of being reasoned by designers with domain expertise, consistency and requirement constraints of the building design are effectively embedded into the building data structure, enriching the building knowledge graph with design-oriented information. The proposed graph-based approach provides greater cost-effectiveness on automated design adaptation, especially for generative designs aiming to fulfill particular requirement constraints. Broader development and adoption of such a graph-based constraint representation approach could significantly increase automated design efficiency.

A comprehensive design constraint classification will ease the data query and calculation processes. In addition to functional classes, hard and soft constraints should be distinguished to support an adjustable selection of constraints maintained by users. Additionally, the combinatorial ranges for multiple parameters should be computed to prevent leading to iterative and contradictory design works for architects and engineers.

Acknowledgements

We thankfully acknowledge the TUM International Graduate School of Science and Engineering (IGSSE) for their financial support. Additionally, the research presented in this paper has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement no. 958398, “BIM2TWIN: Optimal Construction Management & Production Control”. The authors thankfully acknowledge the support of the European Commission in funding this project.

References

[1] A. Borrmann, M. König, C. Koch, and J. Beetz, “Building information modeling: Technology foundations and industry practice”, in *Building Information Modeling: Technology Foundations and Industry Practice*, 2018, pp. 1–584. DOI: 10.1007/978-3-319-92862-3.

[2] A. Borrmann, J. Hyvärinen, and E. Rank, “Spatial constraints in collaborative design processes”, *Proceedings of the International Conference on Intelligent Computing in Engineering (ICE’09)*, pp. 1–8, 2009. [Online]. Available: http://www.andre-borrmann.de/docs/paper_ICE09.pdf.

[3] J. Kirchner, “Computation of Ranges in Interval-based Constraint-Geometry of Building Models”, in *International Workshop on Intelligent Computing in Engineering*, 2022, pp. 206–215. DOI: https://doi.org/10.7146/aul.455.c211.

[4] G. Lee, R. Sacks, and C. M. Eastman, “Specifying parametric building object behavior (BOB) for a building information modeling system”, *Automation in Construction*, vol. 15, no. 6, pp. 758–776, 2006. DOI: 10.1016/j.autcon.2005.09.009.

[5] S. Vilgertshofer and A. Borrmann, “Using graph rewriting methods for the semi-automatic generation of parametric infrastructure models”, *Advanced Engineering Informatics*, vol. 33, pp. 502–515, 2017. DOI: 10.1016/j.aei.2017.07.003.

[6] C. Schultz, M. Bhatt, and A. Borrmann, “Bridging qualitative spatial constraints and feature-based parametric modelling: Expressing visibility and movement constraints”, *Advanced Engineering Informatics*, vol. 31, pp. 2–17, 2017. DOI: 10.1016/j.aei.2015.10.004.

[7] J. Kirchner and W. Huhnt, “Benefits and Limits of Interval Constraints in Acyclic Constraint Graphs for Geometric Modeling”, *Proc. of the 17th International Conference on Computing in Civil and Building Engineering*, 2018.

[8] P. Pauwels, S. Zhang, and Y. C. Lee, “Semantic web technologies in AEC industry: A literature overview”, *Automation in Construction*, vol. 73, pp. 145–165, 2017. DOI: 10.1016/j.autcon.2016.10.003.

[9] Z. Tang, Q. Zou, H.-Y. Feng, S. Gao, C. Zhou, and Y. Liu, “A review on geometric constraint solving”, pp. 1–15, 2022. arXiv: 2202.13795. [Online]. Available: http://arxiv.org/abs/2202.13795.

[10] H. Arora, J. Bielski, V. Eisenstadt, *et al.*, “Consistency Checker: An automatic constraint-based evaluator for housing spatial configurations”, *Proceedings of the International Conference on Education and Research in Computer Aided Architectural Design in Europe*, vol. 2, pp. 351–358, 2021. DOI: 10.52842/conf.ecaade.2021.2.351.

[11] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, and R. Klein, “The IFC to linked building data converter - Current status”, *CEUR Workshop Proceedings*, vol. 2159, no. June, 2018.