2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

A SCALABLE STRATEGY FOR CONFIGURATOR ADOPTION IN RESIDENTIAL PROJECTS

Jianpeng Cao[1], Evangelos Pantazis[2], Furio Sordini[3], Giulia Curletto[3] and Konrad Graser[2]
[1]The University of Hong Kong, Hong Kong S.A.R., China
[2]Zurich University of Applied Sciences, Zurich, Switzerland
[3]Implenia, Zurich, Switzerland

Abstract

In the era of mass customization, industrialized construction could take advantage of product configurators to balance standardization and design flexibility, therefore maintaining productivity and economic efficiency. Previous studies have proposed conceptual frameworks for product configurators, but few studies illustrate how they could be implemented in a practical building project. This paper extends the framework to prototype development and explains the implementation process in collaboration with a construction company. The study contributes to a more generalized framework enabled by graph-based configuration rules and a scalable implementation plan of configurators in practice.

Introduction

Real estate sector not only offers space to live and work but also fundamentally simplifies and enriches the lives of its users. From initia customer research to production supply chain planning, the sector encounters demand from various stakeholders, including design, manufacturing, construction, and maintenance. To provide responsive control over the entire building life cycle, the industry needs to transform from a one-off project to a scalable product approach. Productization implies the adoption of industrialized construction (IC), also referred to as modern methods of construction, which offers potential benefits, such as reduced construction time, enhanced quality control, and minimized on-site labor, by integrating advanced manufacturing techniques, streamlined workflows and offsite fabrication. However, meeting unique client needs while maintaining standardized processes remains a significant challenge for industrialized construction. Mass customization, which aims to provide individualized design options on a large scale by relying on standardized construction processes, offers a potential solution in the long-standing problems within the AEC industry. Effective implementation of mass customization would enable design flexibility that aligns with both customers' preferences and manufacturers' capabilities.

One emerging technology to support mass customization in industrialized construction is product configurators (Veenstra et al., 2006). The unique selling proposition (USP) of the configurator lies in automatically generating a variety of building configurations using predefined real estate products to obtain economies of scale. A configurator is a decision support system that automates the combination of kit-of-parts into efficient modules for production under predefined rules while enabling rapid generation of product variety to meet the desired product features for the customer (Cao et al., 2021). There is an increasing application of configurators in the construction industry, such as KOPE, TESSA, TESTFIT, etc. Despite some success stories, the application of configurators in construction is limited and immature, particularly among construction firms transitioning to industrialized construction.

This work presents a novel strategy for configurator adoption by the real estate sector. The strategy applies the conceptual configurator framework proposed in our previous work to a residential project in collaboration with a real estate developer. The framework contains two pillars: a kit-of-part library and a configuration rule engine. The kit-of-part library maintains a set of standard prefabricated products, such as planar products (e.g., panels). Configuration rules specify how the kit-of-parts and their properties can be defined to meet a given requirement. Specifically, this work contributes to a more intelligent configurator enabled by graph-based configuration rules. This approach enhances flexibility by using graph structures to represent relationships between components and their properties, allowing for adaptable configurations to meet various project requirements. Additionally, the implementation plan details the practical steps needed to adopt configurators within the construction process, aligning with real-world constraints. This work effectively bridges the gap between theoretical concepts and practical applications, offering a scalable strategy for configurator adoption in the construction industry.

The paper is organized as follows. In the next section, we conduct a literature review on the study of configurator, focusing on the different category of configurators and

techniques adopted. Next, we generalize the graph-based representation and configuration rules. After that, we gave an illustrative example of how the configurator approach is applied by a real estate developer to a floor plan of a residential project. Finally, we discuss the efficiency obtained by the configurator and future research and conclude with contributions to the literature.

Related work

Previous studies proposed a Building Information Model (BIM) application as a configurator due to its capability of parametric design (Piroozfar et al., 2019). However, the concept is limited to small-scale systems, such as facades. The BIM-as-a-Configurator strategy is appropriate for this type of system as the major task is to adapt the size/dimensions of the kit-of-parts (e.g., mullions) to customized façade design. For more complicated industrialized systems, such as buildings, the configuration task also includes the selection of kit-of-parts and assignment of them to the architectural layout. Early studies used Ontology and Semantic Web Rule Language (SWRL) to formulate configuration rules as description logic, such as composition, compatibility, dependency, and cardinality (Yang and Dong 2013). The techniques are also adopted in modular building design (Cao and Hall 2020). However, the application of SWRL rules is limited in representing complex adjacency relationships. A recent study on large language models (LLM) also shows the potential to advance automatic configuration (Wei et al., 2024). However, the research is still limited in the conceptual paradigm stage and lacks practical application. There is a need to bridge the gap between cutting-edge research and a company's existing solutions and routines.

Graph-based representation and analysis

Graph representation has been widely used for generic floor plans, capturing spatial relationships and structural elements (Ślusarczyk 2018). This allows for a computational analysis of the design (Pizarro et al., 2022). This work builds a graph representation of a floorplan by creating walls, rooms, and apartments as nodes and defining their relationships as edges. Walls are added as nodes with attributes such as start and end points, panel type, height, thickness, room, and apartment information. Rooms are represented as separate nodes and have "*belongs_to*" edges connecting them to their corresponding wall nodes. Similarly, apartments are represented as separate nodes and have "*belongs_to*" edges connecting them to their corresponding room nodes (shown in Figure 1). Additionally, internal walls shared by two rooms are counted twice, once for each room, resulting in duplicate wall nodes that are linked using "identical" edges. In this way, the configurator could distinguish exterior walls from internal ones if a wall node has no "*identical*" edge. The graph is defined as:

$$G = (V, E)$$

Where:

- V is the set of nodes
- E is the set of edges
- $v \in V$ represents a wall node
- $r \in V$ Represents a room node
- $A \subseteq G$ represents an apartment subgraph, containing walls and rooms related to a specific apartment.

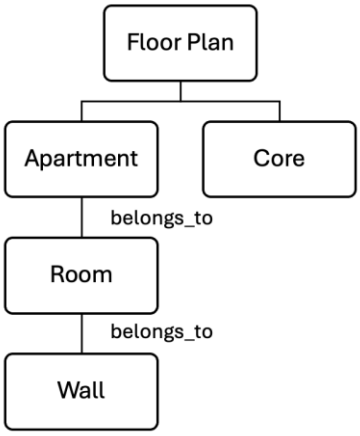

Figure 1: Hierarchical structure of the floor plan

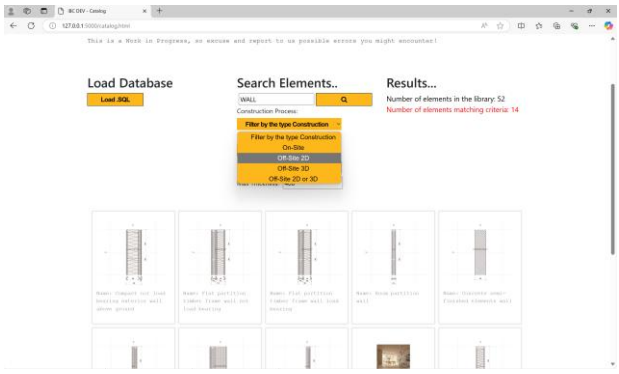

Figure 2: Various panel types in the kit-of-part library

Built upon the graph representation, the selection of the panel type from a kit-of-part library could be inferred using graph-based configuration rules. Figure 1 shows an example of a prefabricated panel library. Three types of rules are generalized:

Node-attribute-based rules

Rules that classify nodes based solely on their properties:

$$\forall v \in V, if\ f(v) \in C \Rightarrow g(v) \leftarrow x$$

Where:

- V represents the set of nodes (walls).
- $f(v)$ extracts an attribute of node v (e.g., room type).
- C is a predefined condition set (e.g., Core, Bathroom).
- $g(v)$ updates a property of v (e.g., panel type).
- x is the assigned classification (e.g., WAL_26, WAL_33).

Edge-attribute-based rules

Rules that classify nodes based on their relationships (edges) with other nodes:

$$\forall e(v,u) \in E, if\ type(e) \in C \Rightarrow g(v) \leftarrow x$$

Where:

- E represents the set of edges.
- e is an edge between nodes v and u.
- $type(e)$ extracts the relationship type (belongs_to, identical).
- C is a predefined set of edge conditions.
- $g(v)$ updates a property of v (e.g., panel type).

Subgraph-attribute-based rules

Rules applied within subgraphs, considering structural relationships:

$$\forall v \in A \subseteq G, if\ \nexists e \in E' \Rightarrow g(v) \leftarrow x$$

Where:

- A is an induced subgraph of G (e.g., all walls within an apartment).
- $E'$is the edge set of A.
- $\nexists e$ ensures no edges satisfy a given condition (e.g., identical).
- $g(v)$ updates v's property based on its isolation in A

These rules could be combined and applied iteratively by analyzing node attributes and edge types within the graph or subgraph. By encoding spatial relationships and adjacency conditions, the corresponding panel types could be selected based on context-specific constraints For instance, walls directly connected to certain room types (e.g., core, bathroom) are assigned specific panel types. The graph representation enables efficient traversal, allowing rules to propagate through hierarchical relationships, detect shared elements, and resolve edge cases, such as walls adjacent to multiple spaces. This approach ensures scalability and precision in automating configuration decisions across diverse building layouts.

Validation case

A case study from Implenia AG, a Swiss real estate developer, is presented to demonstrate the practical application of the theoretical framework. The configurator implementation begins with an architectural floor plan in PDF format as the project input. The process consists of four sequential operations: *geometry extraction*, *graph representation*, *floor plan configuration*, and *3D model generation*. Once these steps are completed, the configurator outputs a bill-of-material (BOM) sheet and a building information model (BIM) in IFC format, ensuring seamless integration with digital construction workflows.

Geometry extraction

An Interactive Graphical User Interface (GUI) (shown in Figure 3) is developed to facilitate geometry extraction from the floor plan. To accurately convert pixel distances to real-world measurements, users first draw a reference line and specify its actual length. They then select the room type using a predefined color schema and extract the room geometry, which is recorded as enclosed polylines representing both room boundaries and walls. Each polyline captures the spatial relationships and dimensions of the walls. Once all rooms are defined, users can further group them into apartments. The interface provides real-time feedback, allowing users to review processed geometric data, including wall lengths, room areas, and apartment areas, ensuring accuracy before proceeding to the next stage.

Figure 3: Graphical user interface for geometry extraction

Graph representation

The extracted information is structured into a graph representation using the NetworkX Python package. In this graph, walls, rooms, and apartments are represented as nodes, while their relationships, such as "*belongs_to*"

and "*identical*" are established as edges. Besides, key attributes such as start and end coordinates, height, and thickness of the wall are assigned to wall nodes, enabling rule-based configuration in the next step. The graph-based floor plan representation of the colored floor plan in Figure 3 is shown in Figure 4.

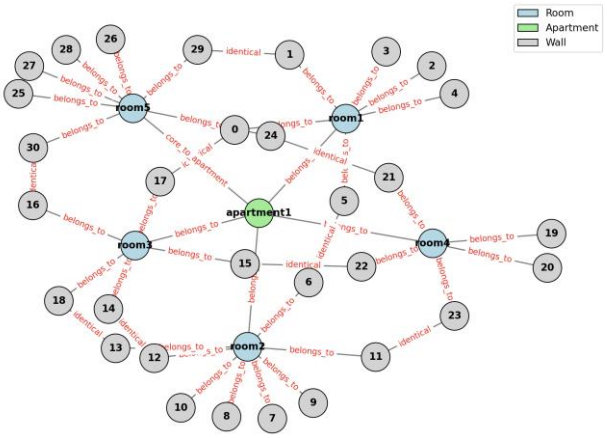

Figure 4: Graph representation of floor plan

Floor plan configuration

The floor plan configuration assigns wall panels from a standard prefabricated kit-of-part library to the customized floor plan using configuration rules. These rules, developed in collaboration with industry professionals, ensure that each wall type is appropriately classified based on its spatial and functional role within the building. The key configuration rules are as follows:

1. WAL_21 is assigned to all exterior walls.
2. WAL_26 is assigned for walls along the core.
3. WAL_24 is applied to apartment partition walls.
4. WAL_33 is used for the inner perimeter of bathrooms.
5. WAL_31 is assigned to walls that are positioned between a bathroom and either a corridor or a living room, ensuring that the wall faces the corridor or living room.
6. WAL_25 is assigned to all remaining walls.

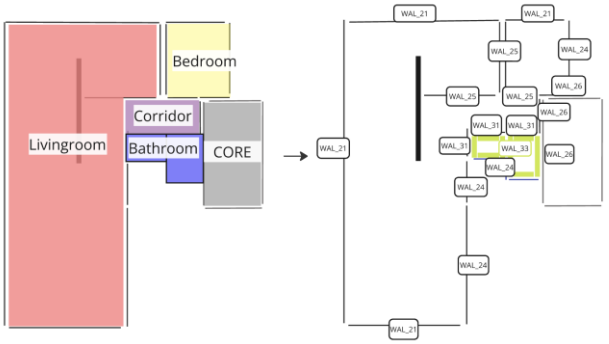

Figure 5: Floor plan configuration

Built upon the graph representation and configuration rules introduced before, these rules could be formulated to conduct automatic inference on kit-of-part selection and assignment. The output of the floor plan configuration is a bill of material (BOM) that represents all panels adopted in the floor plan in a structured format. The file contains panel type, coordinates of start and end points, height, wall-associated room, and wall-associated apartment, which could be used for cost estimates and 3D geometry development.

name	panel_type	start_point	end_point	height	room	apartment
13	WAL_25	8.01,7.03,0	8.06,9.08,0	3	living_room	Apartment 1
8	WAL_21	8.12,17.95,0	3.19,17.95,0	3	living_room	Apartment 1
1	WAL_24 WAL_33	12.06,9.08,0	12.06,11.59,0	3	bathroom	Apartment 1
4	WAL_21 WAL_33	10.01,10.52,0	8.12,10.52,0	3	bathroom	Apartment 1
25	WAL_21	14.53,6.98,0	17.14,7.03,0	3	core	None
18	WAL_25	8.06,9.08,0	8.01,7.03,0	3	corridor	Apartment 1
5	WAL_33	8.12,10.52,0	8.06,9.08,0	3	bathroom	Apartment 1
9	WAL_21	3.19,17.95,0	3.19,2.62,0	3	living_room	Apartment 1
30	WAL_26	12.06,9.08,0	12.06,7.03,0	3	core	None
0	WAL_33	8.06,9.08,0	12.06,9.08,0	3	bathroom	Apartment 1
14	WAL_25	8.01,7.03,0	10.58,7.03,0	3	corridor	Apartment 1
28	WAL_21	12.12,13.08,0	12.06,11.59,0	3	core	None
17	WAL_31	12.06,9.08,0	8.06,9.08,0	3	corridor	Apartment 1
11	WAL_25	10.58,2.87,0	10.58,7.03,0	3	living_room	Apartment 1
24	WAL_26	12.06,7.03,0	14.53,6.98,0	3	core	None
15	WAL_25	10.58,7.03,0	12.06,7.03,0	3	corridor	Apartment 1
21	WAL_24	14.53,6.98,0	12.06,7.03,0	3	bedroom	Apartment 1
20	WAL_21	14.53,2.87,0	14.53,6.98,0	3	bedroom	Apartment 1
26	WAL_21	17.14,7.03,0	17.14,13.08,0	3	core	None
2	WAL_21 WAL_33	12.06,11.59,0	10.01,11.59,0	3	bathroom	Apartment 1
23	WAL_25	10.58,7.03,0	10.58,2.87,0	3	bedroom	Apartment 1
10	WAL_21	3.19,2.62,0	10.58,2.87,0	3	living_room	Apartment 1
27	WAL_21	17.14,13.08,0	12.12,13.08,0	3	core	None
29	WAL_26	12.06,11.59,0	12.06,9.08,0	3	core	None
12	WAL_25	10.58,7.03,0	8.01,7.03,0	3	living_room	Apartment 1
3	WAL_21 WAL_33	10.01,11.59,0	10.01,10.52,0	3	bathroom	Apartment 1
22	WAL_25	12.06,7.03,0	10.58,7.03,0	3	bedroom	Apartment 1
7	WAL_21	8.12,10.52,0	8.12,17.95,0	3	living_room	Apartment 1
19	WAL_21	10.58,2.87,0	14.53,2.87,0	3	bedroom	Apartment 1
6	WAL_31	8.06,9.08,0	8.12,10.52,0	3	living_room	Apartment 1
16	WAL_24	12.06,7.03,0	12.06,9.08,0	3	corridor	Apartment 1

Figure 6: Bill of material of wall panels

3D model generation

The BOM file is parsed to create an IFC file using the IFCopenshell Python package. For each wall, the 2D profile for the cross section is built and extruded into a 3D solid. Next, the solid is assigned to an IfcWallStandardCase entity, built on which material property could be added. Finally, we place it in the building story and write the IFC file for information exchange with the downstream supply chain. The process is also valid for horizontal elements.

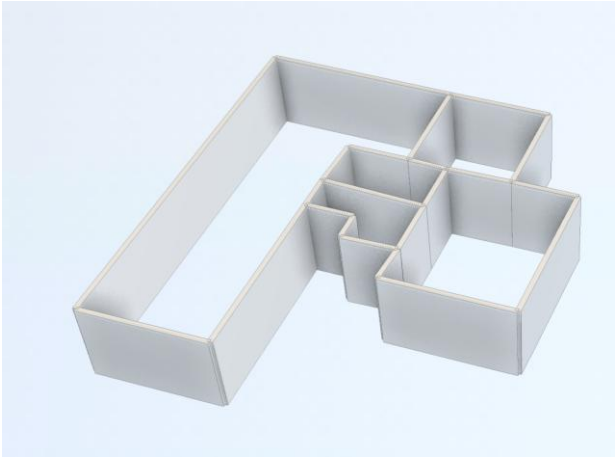

Figure 7: Floor plan configuration

Discussion

The configured output was cross-referenced with manually created configurations by designers for the same floor plan design from accuracy and efficiency aspects. Accuracy was assessed through element-level comparison, measuring the percentage of correctly assigned kit-of-parts in the automated output relative to the manual benchmark. In the demonstrated case, the configurator achieved 100% accuracy, indicating

complete alignment with expert-generated configurations. Efficiency was measured by comparing the time required to complete configurations. While manual configuration typically took several hours depending on design complexity, the configurator generated equivalent outputs in a matter of seconds. The results show that the configurator effectively automates the time-consuming and error-prone manual configuration process, especially when designers lack familiarity with the kit-of-parts and configuration logic. This issue often arises when floor plans are developed by third-party designers, while the kit-of-parts are managed by real estate developers or manufacturers. Nonetheless, further experiments are required to validate the accuracy and robustness of the approach across diverse design scenarios.

The proposed workflow does not aim to replace the creative role of architects but rather to support them in adopting mass customization in their designs. By utilizing conceptual floor plans as inputs, architects retain their design flexibility while the system automatically selects and assigns kit-of-parts. This approach optimizes the balance between customization and economies of scale, facilitating an efficient and scalable design process.

Furthermore, the proposed workflow does not rely on BIM files as input, instead generating IFC files as output. It begins with a floor plan, which could be a sketch or a detailed plan, from which the geometric information gets extracted. This feature enables the configurator to offer service in any project phase. Unlike existing sketch-to-BIM or PDF-to-BIM tools which could only generate the 3D geometry, our approach utilizes the configuration rules to automate the kit-of-part selection and assignment, leading to a design solution for industrialized construction. If BIM files are used as input, the workflow becomes further simplified, as configuration rules can be directly applied to BIM entities to perform inference and replace generic components with engineered kit-of-parts. This approach not only streamlines feasibility assessments but also enhances interoperability in downstream processes by automating BIM file generation, ensuring seamless integration across the construction workflow.

This study has some limitations. The interactive GUI for geometry extraction, while designed to be user-friendly, still relies heavily on manual operation. To enhance usability, we have implemented supporting features such as grid lines, point snapping, vertical/horizontal alignment aids, and automated enclosed polyline checking. These tools help guide users through the geometry extraction process and reduce input errors. However, despite these efforts, the accuracy of the extracted geometry is not fully guaranteed. For example, misaligned line segments and the creation of duplicated points can still occur, affecting the quality of subsequent BIM generation. Future research should explore the use of computer vision techniques to automate the geometry extraction process, further improving accuracy, scalability, and user experience. Besides, another research opportunity should focus on incorporating manufacturability checks during the kit-of-part selection process to ensure that selected kit-of-parts align with production constraints and construction feasibility.

Conclusions

This study proposed a novel strategy for adopting configurators in industrialized construction. Built upon an existing configurator framework, the study integrates graph-based configuration rules to automate the configuration process of floor plans. This strategy was demonstrated in a case study in collaboration with a real estate developer who owned the kit-of-part library. The implementation plan follows an efficient process from geometry extraction to 3D model generation. The configurator bridges traditional design approaches with industrialized construction through the integration of a kit-of-parts, and contributes to a scalable strategy for configurator adoption in the construction industry.

Acknowledgments

This work was supported by Innovationsprojekt / Projekt Nr. 108.408.1 IP-SBM

References

Veenstra VS, Halman JIM, Voordijk JT. A methodology for developing product platforms in the specific setting of the housebuilding industry. Res Eng Des. 2006;17(3):157-173. doi:10.1007/s00163-006-0022-6

Cao J, Bucher DF, Hall DM, Lessing J. Cross-phase product configurator for modular buildings using kit-of-parts. Autom Constr. 2021;123:103437. doi:10.1016/j.autcon.2020.103437

Piroozfar P, Farr ERP, Hvam L, Robinson D, Shafiee S. Configuration platform for customisation of design, manufacturing and assembly processes of building façade systems: A building information modelling perspective. Autom Constr. 2019;106(July). doi:10.1016/j.autcon.2019.102914

Yang D, Dong M. Applying constraint satisfaction approach to solve product configuration problems with cardinality-based configuration rules. J Intell Manuf. 2013;24(1):99-111. doi:10.1007/s10845-011-0544-2

Cao J, Hall D. Ontology-based Product Configuration for Modular Buildings. Proc Int Symp Autom Robot Constr. 2020;(October):171-176. doi:10.22260/isarc2020/0026

Wei Y, Li X, Wu C, Zahedi A, Guo Y, Yang Z. Generation for Configuration: A Conceptual Paradigm of a Natural Language-Based Configurator for Modular Buildings with ChatGPT. The 28th International Symposium on Advancement Of Construction Management and Real Estate. Lecture Notes in Operations Research. Springer Nature Singapore; 2024:1491-1501. doi:10.1007/978-981-97-1949-5

Ślusarczyk G. Graph-based representation of design properties in creating building floorplans. CAD Comput Aided Des. 2018;95:24-39. doi:10.1016/j.cad.2017.09.004

Pizarro PN, Hitschfeld N, Sipiran I, Saavedra JM. Automatic floor plan analysis and recognition. Autom Constr. 2022;140(December 2021):104348. doi:10.1016/j.autcon.2022.104