Enriching IFC Models with Spatial Design Logic and Parametrics to Improve Design Adaptability – the case of alignment grids

Jiabin Wu, Sebastian Esser, Stavros Nousias, André Borrmann

Technical University of Munich, Arcisstr. 21, 80333 Munich, Germany
j.wu@tum.de

Abstract
The Industry Foundation Classes (IFC) data model is broadly used in architectural and construction engineering design. Despite the comprehensive features of the IFC data model, the transfer of design logic and parametrics across different platforms is limited. This is mainly due to the insufficient use of advanced IFC features by the export modules of authoring tools, resulting in the loss of parametrics. To support the exchange of the design logic, this paper introduces an automated enrichment method to enhance IFC data models with explicit design parametrics toward a more adaptable design process. In this paper, we focus on identifying grid logic, which is essential for the spatial reference system. This approach encompasses grid estimation, alignment, and parameter computation, utilizing quantitative thresholds for improved flexibility. By aligning and merging the grid references, relationships between building elements and grids are articulated through grid-based design parametrics. The effectiveness of this approach is demonstrated through a case study. This innovative strategy provides a retrieval mechanism for automatically identifying spatial grid logic, laying the groundwork for rapid design adaptation. The proposed enrichment method can significantly improve the variability of IFC data models in complex architectural and engineering scenarios.

Keywords: Building information modeling, Industry foundation classes, Parametric modeling, Reverse engineering.

1 Introduction

In building design and construction, the adaptability of design models is essential for reducing tedious manual reworks. To achieve a higher design efficiency, it is necessary to exchange the design knowledge successively to support various design decisions. Modern Building Information Modeling (BIM) authoring tools, such as Autodesk Revit, Graphisoft ArchiCAD, Vectorworks, Allplan, and others, provide comprehensive parametric modeling capabilities to allow users to create flexibly adaptable building models by defining constraints, dependencies, and parameters to steer dimensions and distances while preserving the consistency of the overall model [1]. Parametric models can be altered through a list of parameters, resulting in the re-generation of the model [2]. The resulting flexible model allows a quick reaction to changes during the design process and enables the reuse of designs across different projects.

A key role in the parametric design of buildings plays the concept of reference planes and reference lines, to which building elements can be aligned. If the position or orientation of the reference entity is changed, the bound building components automatically update their position and dimensions. If, for example, the storey height needs to be adjusted due to a change in the boundary conditions, all walls bound to the respective reference planes are automatically adjusted and continue extending over the full height of the storey.

Another example is the grid lines in a floorplan, which are used to set up a regular layout of columns and walls to enable an efficient design of the structural system. Typically, the gridlines are equidistantly spaced. The gridline distance in each of the two coordinate axes is typically defined as a single parameter that can be varied, then modifying all distances at once. If the system of gridlines and bound components (walls, columns) is set up thoroughly and comprehensively, the floorplan layout can be altered consistently by the user by modifying the grid distance parameters. This allows the rapid exploration of different design options and the assessment of the consequences of modified grid distances on the space program, the architectural quality, and the structural efficiency of the building. By establishing a system of reference planes and gridlines and parametrically binding elements to them (**Fig. 1**), the designer is able to make the design intent explicit and capture the engineering knowledge that goes along with the design of floorplans.

Industry Foundation Classes (IFC) is the established vendor-neutral data model for the exchange of semantically rich 3D BIM models. IFC incorporates building objects, their attributes, and interrelationships by structuring the data around four layers [1], supporting the sharing of construction information and inter-domain exchange for particular purposes [3]. A Model View Definition (MVD) specifies a subset of the overall IFC schema for a dedicated data exchange scenario. Different MVDs are prepared for specific domain uses from a single IFC project, such as architectural view, mechanical system view, structural frame view, etc. A model view provides consistency and predictability across different software platforms to particular configurations [4]. Depending on the exchange requirements, different design information, such as engineering properties, 3D geometry, and topological information, can be included in the view. Current BIM authoring software's export modules mainly implement the *Coordination View* (IFC2x3) and *Reference View* (IFC4). By purpose, they provide only limited support for procedural and parametric geometry to make their implementation easy, especially for viewing and quantity-take-off applications.

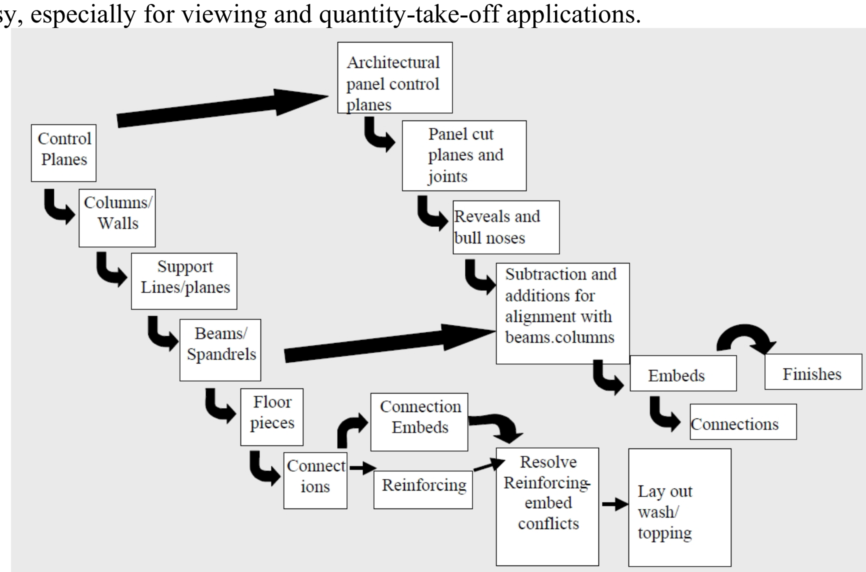

Fig. 1. A system of reference planes and lines forms the basis for setting up a comprehensive parametric building design (Figure from lecture material by Chuck Eastman, GeorgiaTech)

While IFC provides advanced capabilities to represent reference lines and planes by means of the entity *IfcGrid* and *IfcGridPlacement*, these elements are hardly used by the export modules of the respective authoring tools. In consequence, the parametric data is lost in the exchange process, making the exchanged model static and hard to modify. The insufficient support of specific use cases by an IFC exchange includes using the IFC model for subsequent design phases that may require the parametric change of grid line distances. There is also limited support for using the IFC model as a base model for similar projects that might require a change in grid line distances.

These disadvantages limit the reusability of IFC models, posing challenges in accurately reflecting the underlying engineering knowledge and design intents [5], especially when participants accomplish the design activities parallelly using different software systems. The capabilities of the IFC data model to support more advanced exchange scenarios are constrained in the absence of a detailed representation of parametrics and design logic. There's a need for more thorough investigations into entities like *IfcGrid* to assist in making the IFC-based model exchange more capable [6].

To overcome this issue, we propose a method that allows for re-engineering the grid lines from the evaluated (static) model and enriching the IFC model with them. An encoding approach is developed to uncover the grid logic and translate it into parametrics, encompassing grid estimation, alignment, and parameter computation. We set quantitative thresholds to provide flexibility for grid estimation and alignment. With the adjusted grid references, the relationships between building elements and the grids are adapted and represented via design parametrics. As a result, the underlying parametrics are re-established, and the model becomes variable and adaptable. In this sense, the developed approach also contributes to the concept of capturing engineering knowledge and making implicit design intent explicit again in a vendor-neutral building information model. **Fig. 2** exemplifies how semantic and geometric data are utilized to derive grid lines for developing the design parametrics.

We demonstrate the effectiveness of the proposed approach through a case study. The results show that the gridline-oriented enrichment approach effectively reveals the spatial design logic and creates design parameters. The flexible grid estimation and alignment enable designers to reach adaptable spatial reference systems in an automated manner. Through primary parameters and their dependencies, the approach ensures that the underlying design knowledge and intentions are effectively communicated to the recipient of the IFC data model. The approach establishes a foundation for the resilient adaptation of models in response to evolving design concepts.

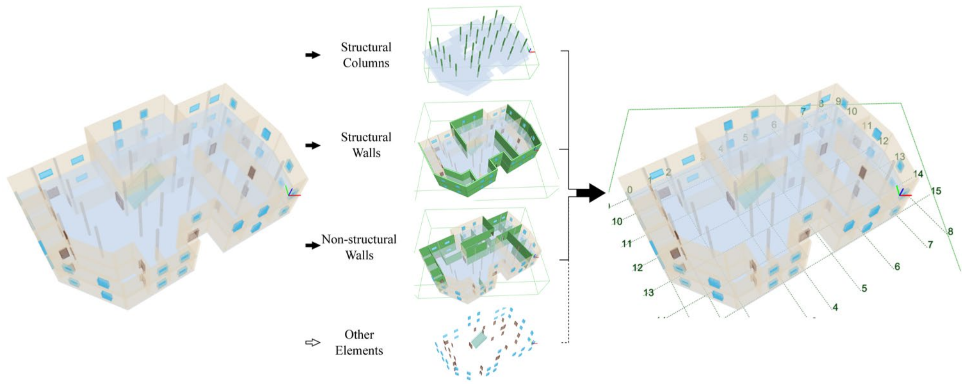

Fig. 2. Enrich the IFC model with logical grid lines from semantic and geometric design information.

The current state of the IFC schema and the need for representing design logic and parametrics for design adaptation is addressed in Section 2 of this paper. In Section 3, the proposed approach is presented. Section 4 is dedicated to the experiment and the related results. A summary of the contributions and discussions about future research is provided in Section 5, which concludes this paper.

2 Background

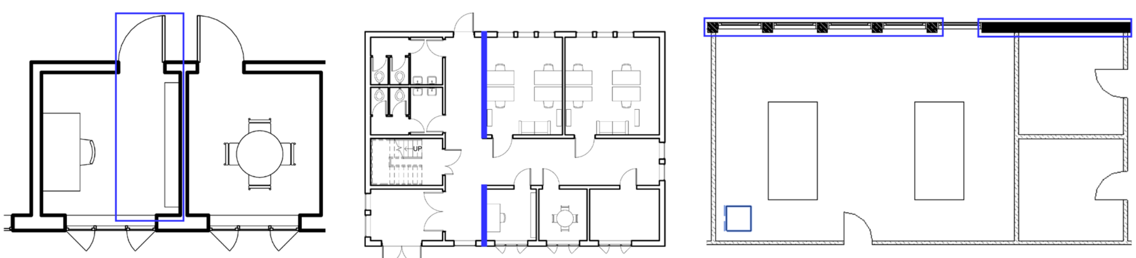

Fig. 3. Design examples embedding engineering knowledge and design intent: a) minimum distance of door placement, b) wall alignment for the straight corridor, and c) alignment among structural columns and walls.

The continuously developing computer-aided design (CAD) methods have replaced conventional design drawings with digital modeling approaches [7]. Creating design models and simulations with digital parametric systems offers numerous benefits regarding adaptability and productivity. Through parametric modeling, designers define specific properties, dimension values, and interrelations to express the logic of the related design conditions. With the geometric, semantic, and topological information in BIM models [1], the parametric modeling approaches support embedding domain design knowledge and specified design intentions. Taking the building design as an example, many building elements are identical to other parts and used repetitively for identical functionalities [7]. Besides, certain relations hold in the general case for all buildings (e.g., doors are placed with a minimum distance to the wall edges to leave enough spaces for shelves, interior walls are aligned to maintain a straight corridor, and the locations for columns are usually aligned for structural needs, illustrated in **Fig. 3**.) By the help of parametric design, domain knowledge, and design intents can be captured in the model.

The introduction of BIM has improved the efficiency in design optimization and evaluation. Building designs must undergo regular checks to fulfill design codes and regulations to avoid design errors and ensure building safety [8]. The design, review, and coordination processes typically involve multiple software applications. The exchange of BIM models across these applications is often realized with the help of the vendor-neutral format IFC. The usual process is to export an IFC file from the design authoring system for subsequent design checks using a model checker software, after which specific design analyses and simulations are performed. When design activities are accomplished parallel, the underlying design knowledge and design intent in the original model need to be preserved, which requires flexible parametric systems to keep the variability in different design stages. Furthermore, there is a continuous need for BIM-based configurators to re-configure and adapt the design across the building lifecycle [9].

The Design Transfer View (DTV) was initially proposed to support the handover of parametric designs for further design editing across different platforms [10]. However, the aimed “parametric exchange” is not comprehensively investigated. Parametric transfers can be complicated for objects carrying complex geometric features like stairs and curtain walls, while the schema supports less detailed definitions. The buildingSMART Technical Roadmap [11], as published, delineates the anticipated generations of IFC along with prioritized proposals for enhancements within the IFC ecosystem. The present iteration of IFC is being optimized to facilitate collaboration in a BIM-supported connected Common Data Environment (CDE) among project stakeholders and to support a range of automated services. Future developments in IFC predominantly focus on modularization and consistency. In addition, a notable trend is the separation of geometry and semantics in entities such as *IfcGrid* and *IfcAlignment* to enhance the variability of the IFC instances [6].

Current initiatives focus on exchanging and transferring design parameters on a single object level. However, the accomplished flexibility is limited from a design perspective. While building design is a complex and iterative prosses of continuous improvement, the IFC data models are not sufficiently comprehensive to capture all the design intent underlying the geometries, and the advanced design parametrics are only stored in the native formats of the authoring systems [5]. Although simple explicit geometries are sufficient for coordination via the IFC data model, more parametric geometries are required to exchange design concepts and specific design requirements [12]. On the other hand, incorporating all design information into IFC models might overload the model exchange process. Therefore, a pragmatic approach is to take the IFC model as a ‘common denominator’ for reasoning and externalizing the embedded design knowledge. Based on the standard geometric model, the engineering knowledge and the underlying design intentions can be investigated to reveal the initial design trends without increasing the complexity of the IFC data model. For example, automated interpreted information exchange is accomplished for model transformation with connectivity adjustments from the Design Transfer View to the Structural View [13].

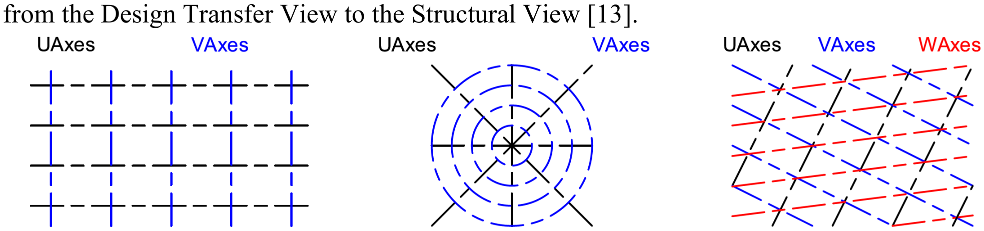

Fig. 4. Different forms of grids on which building elements can be placed [1].

Current investigations into the broader applications of existing IFC classes, particularly in revealing design logic for building projects, are incomplete. Basic semantic properties, like the *LoadBearing* attribute, are usually assigned to elements from *IfcWall*, *IfcColumn*, *IfcBeam*, and *IfcSlab*, which enables the elements to be classified into structural and non-structural elements. Load-bearing walls, typically aligned directly above or parallel to structural supports like beams and columns, are crucial in identifying the design logic. Often positioned perpendicular to floor joists and extended from the foundation to the roof, these structural walls transfer loads across different floors and ensure the building’s stability. Another existing class, the *IfcGrid* class, known for its flexibility in defining reference systems, is a foundational tool for aligning and connecting various building elements. As shown in **Fig. 4**, different forms of grids, including rectangular, radial, and triangular layouts, can be placed to reflect the designers’ original layout intentions.

The coordinate system of any IFC model is organized in a hierarchical structure that affords greater flexibility when design changes occur [1]. Following the Local Placement concept, all relative coordinates remain unchanged when the location of an individual building element is modified. The structural grid lines can be created following the distribution of the structural elements in the building layout, providing a feasible reference system for linking different aspects. Besides, integrating the grids with other building elements allows for analyzing spatial relationships such as collinearity, horizontal alignment, and perpendicularity. Building elements can be linked to estimate the design parameters based on the spatial relations. For example, using locational parameters between reference grids and building elements can ease displacement changes and provide flexibility for various design adaptation objectives [14]. These fundamental design characteristics, rooted in engineering principles and design intentions, can be incorporated to enrich the IFC data model.

This enrichment of design logic and parametrics closely resembles the principles and methodology of reverse engineering (RE) or re-engineering [15]. Reverse engineering is a multifaceted process designed to replicate or modify an existing part, beginning with existing data to either replicate or innovate a current object. The RE approaches are widely adopted in various industrial sectors [16], especially for reconstructing 3D pre-existing objects from the acquired point clouds [17]. Geometric models, such as CAD models, are generated by data capture, segmentation, and surface fitting [18]. Different from these existing studies, we apply the RE principles not to replicate but to enrich the current design model based on the existing data. The IFC data model is re-engineered by investigating the geometric and semantic information to reveal the original design knowledge and intent.

3 Proposed Approach

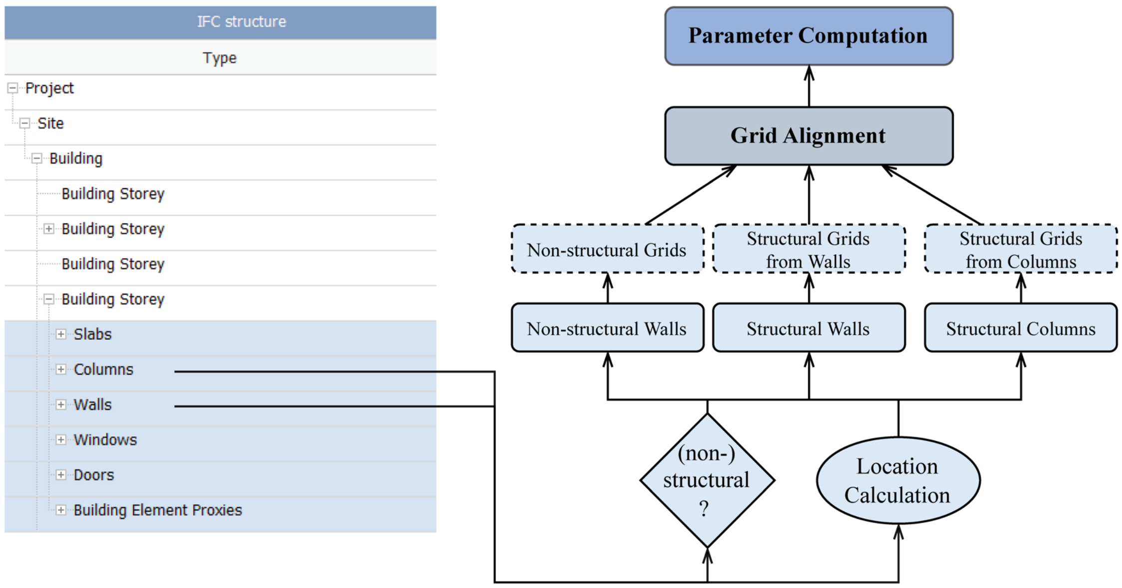

Fig. 5. The proposed approach for enriching the spatial design logic and parametrics via logical grid elements.

This section presents the details of the proposed enrichment approach, which aims to uncover the spatial design logic and parametrics based on the extracted design information from the IFC data model. As illustrated in **Fig. 5**, the approach comprises three main components: (1) estimation of the grids, (2) grid alignment, and (3) parameter computation. Using the OpenCASCADE geometry kernel [19], we parse the semantic and geometric information of the building elements in the IFC model. Properties related to structural integrity, particularly the *LoadBearing* attribute, are requisite. In addition, the approach requires accurate global and local positioning data (*IfcCartesianPoint*, *IfcLocalPlacement*), along with orientation details (*IfcDirection*) of the building elements.

3.1 Grid Initialization

We analyze the wall orientations to identify the dominant direction, revealing the primary global directions of the building layouts. Reference grids are initialized using major building elements from the design layouts, notably *IfcColumn* and *IfcWall*. These grids are divided into non-structural and structural, depending on the originating element type. All *IfcColumn* elements are classified as structural elements, while the classification of an *IfcWall* as structural or non-structural is assessed via the *Load Bearing* attribute. We utilize the positions of the columns, denoted as P_c, to identify all columns that align with a given column within a distance threshold $T_{c,dist}$. One grid line is established when at least $T_{c,num}$ columns are aligned. **Algorithm 1** details the process of creating structural grids L_c based on columns.

The generation of wall-based grid elements is conducted similarly. There is a notable difference when working with elements from *IfcWall* compared to those from *IfcColumn*. For *IfcColumn*, the center location of a column (*IfcCartesianPoint*) is readily accessible through the related *IfcMappedItem*. In contrast, the global locations of *IfcWall* elements cannot be directly parsed from the IFC model. To determine the exact coordinates of an *IfcWall*, the start point coordinates, the element length, and the reference direction are all parsed for calculating the endpoint coordinates. The reference direction for linear walls is given by the *IfcDirection*, which originates from the *IfcLocalPlacement*. Taking the starting and endpoint for each *IfcWall*, the wall-based grid lines are generated. Distance and quantity thresholds $T_{w,dist}, T_{w,num}$ are applied to provide a flexible spatial reference system. This study concentrates explicitly on straight wall elements, typically the essential components of building layouts.

Algorithm 1 Estimate column-based structural grid lines.

Input: $P_c, T_{c,dist}, T_{c,num}$
Output: L_c
$L_c \leftarrow \emptyset$
$P_{c,pairs} \leftarrow CombinationPairs(P_c)$ ▷ combination tuples of column points
for (p_1, p_2) in $P_{c,pairs}$ **do**
  $L_{c,new} \leftarrow \emptyset$
  if $p_1.X \neq p_2.X$ **and** $p_1.Z = p_2.Z$ **then**
    $slope \leftarrow (p_2.Y - p_1.Y)/(p_2.X - p_1.X)$
  else
    $slope \leftarrow$ **inf**
  end if
  for p_{new} in P_c **do**
    if $p_{new} \neq p_1$ **and** $p_{new} \neq p_2$ **and** $p_{new}.Z = p_1.Z$ **then**
      if $|(p_{new}.Y - p_1.Y) - slope * (p_{new}.X - p_1.X)| \leq T_{c,dist}$ **or** $p_{new}.X = p_1.X = p_2.X$ **then**
        $L_{c,new} \leftarrow L_{c,new} \cup p_{new}$ ▷ if columns are located on the same line
      end if
    end if
  end for
  if $len(L_{c,new}) \geq T_{c,num}$ **and** $L_{c,new} \notin L_c$ **then**
    $L_c \leftarrow L_c \cup L_{c,new}$ ▷ if no less than $T_{c,num}$ columns are on the same line
  end if
end for

3.2 Grid Alignment

After initializing grid lines from non-structural and structural elements, we proceed with a grid alignment process. This process aims to synchronize grids with similar functions and align secondary grid lines with adjacent primary grids. The building layout is meticulously designed, especially for the structural system. For instance, load-bearing walls and columns are typically aligned to specific reference lines to enable an efficient design of the structural system. Following the primary structural system, the location of secondary elements, such as separation walls, can be determined based on respective functional areas. As depicted in **Fig. 6**, an alignment relation is identified between a structural-wall-based grid and a structural-column-based grid and between two non-structural-wall-based grids.

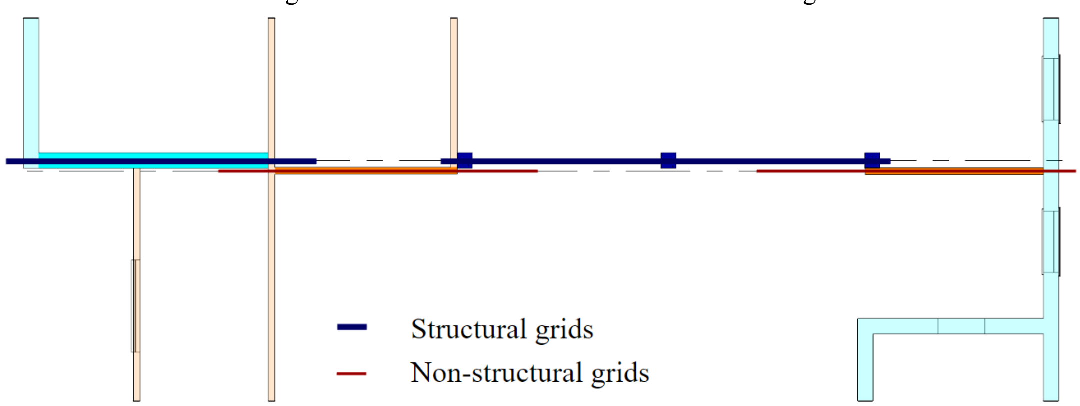

Fig. 6. Identification of alignment relations among structural and non-structural grid lines.

For the (non-)structural grids derived from elements of the same structural type, e.g., between a column-based grid L_c and a structural-wall-based grid $L_{w,st}$, we set a specific threshold, $T_{self,dist}$. If one grid is parallel to another grid of the same type and the distance between them is less than this threshold, we align it with the existing grid.

Non-structural grids ($L_{w,nt}$) are merged with nearby structural grids ($L_c, L_{w,st}$). This merging process, which depends on parallelism ($IsParallel$) and proximity ($T_{cross,dist}$), is detailed in **Algorithm 2**. This algorithm outlines how non-structural grids are integrated with structural ones, ensuring a cohesive and functional grid layout.

Algorithm 2 Merge non-structural grids into structural grids based on parallelism and proximity.

Input: $L_c, L_{w,ns}, L_{w,st}, T_{cross,dist}$
Output: L_{st}, L_{ns}

$L_{st} \leftarrow L_c \cup L_{w,st}$ ▷ structural grid lines (from columns and structural walls)
$L_{ns,init} \leftarrow L_{w,ns}$ ▷ non-structural grid lines (from non-structural walls)
$L_{ns,merge} \leftarrow \emptyset$
for l_1 in L_{st} **do**
  for l_2 in $L_{ns,init}$ **do** ▷ Merge non-structural grid lines to parallel and approximate structural grid lines
    if $IsParallel(l_1, l_2)$ **and** $Distance(l_1, l_2) \leq T_{cross,dist}$ **then**
      if $l_2 \notin L_{ns,merge}$ **then**
        $L_{ns,merge} \leftarrow L_{ns,merge} \cup l_2$
      end if
    end if
  end for
end for
$L_{ns} \leftarrow L_{ns,init} \setminus L_{ns,merge}$

3.3 Parameter computation

The grid initialization and alignment are based on the building elements of type *IfcColumn* and *IfcWall*, of which the dependencies are adjusted when grid alignment and merges occur. As depicted in **Fig. 7**, the non-structural grid is merged with the neighboring structural grid, thus adjusting their relationships with the associated elements accordingly.

Following the alignment and merge relationships, distance parameters are established. These parameters are determined in relation to either the building element and its initially relevant grid line or the element and the newly aligned or merged grid. In this way, the secondary elements in the building layout, especially the non-structural building elements, are binding to the primary structures. Therefore, the relationships between building elements and grids are determined and represented by parametric dependencies, ensuring that the spatial design logic is accurately reflected and maintained with the necessary flexibility for effective design adaptation.

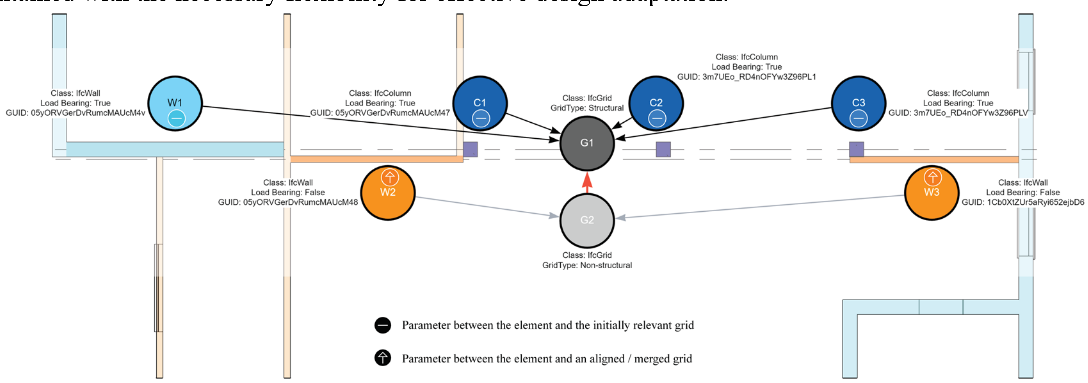

Fig. 7. The merge of non-structural *IfcGird* G2 (generated from aligned non-structural *IfcWall* elements W2 and W3) into the adjacent structural *IfcGird* G1 (developed from aligned *IfcColumn* elements C1, C2, C3 and structural *IfcWall* elements W1).

4 Experiment and results

This experiment aims to demonstrate the applicability of the proposed approach for enriching the spatial design logic and parametrics. We utilize a two-storey building model, specifically an IFC Design Transfer View, as the design for this experiment (**Fig. 8**). The experiment is focused exclusively on straight walls.

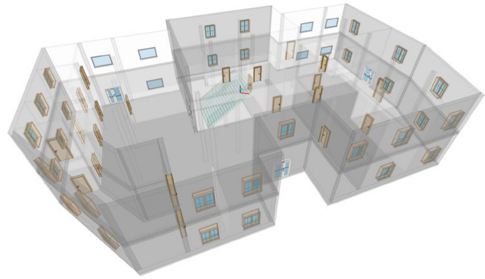

Fig. 8. The overall structure of the two-storey design in the IFC Design Transfer View.

As described in Section 3.1, the distribution of the wall orientations is collected and analyzed. With few non-Manhattan elements detected, the primary global directions of the building layout are determined (**Fig. 9**). Afterward, the global locations of walls and columns are calculated by parsing the geometric information with IfcOpenShell [20]. **Fig. 10** presents the initialized grid elements with selected threshold values in the Level 1 view of the building structure.

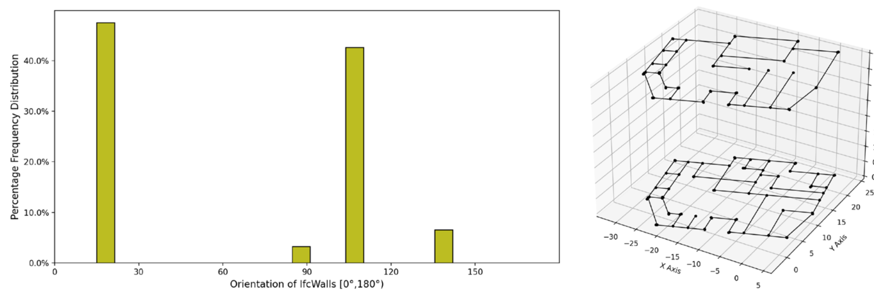

Fig. 9. Illustration of the orientation distribution (left) and global locations (right) of the building walls.

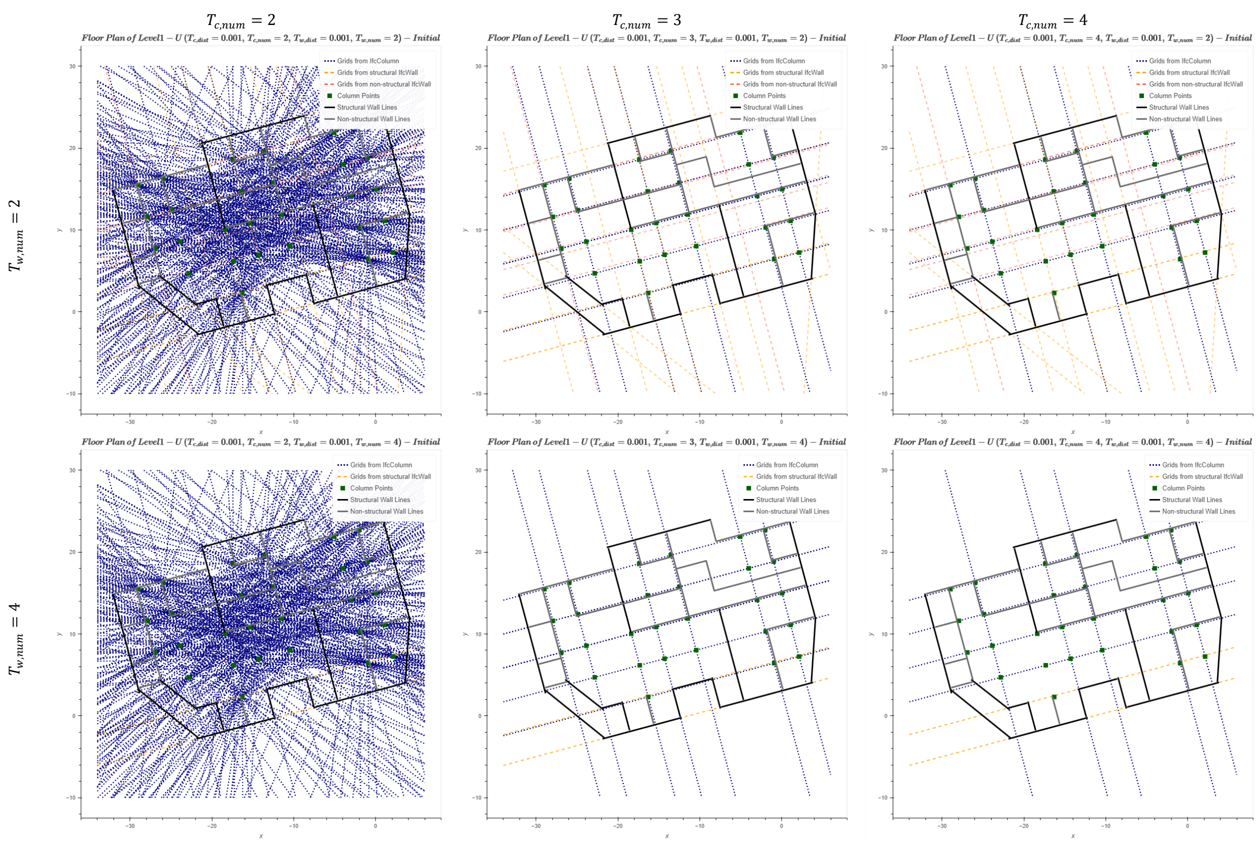

Fig. 10. Initial *IfcGrid* with different combinations of thresholds ($T_{c,dist} = T_{c,dist} = 0.001\ m$; $T_{c,num} = 2,3,4$; $T_{w,num} = 2,4$).

As described in Section 3.2, the alignment of grids is conducted such that each grid is aligned with an adjacent grid of the same type, while non-structural grids are merged with their neighboring structural grids. Detailed results with selected values for thresholds $T_{self,dist}, T_{cross,dist}$ are presented in **Fig. 11**. Adjusting the tolerance values $T_{self,dist}, T_{cross,dist}$ enables the creation of either strict or more flexible spatial reference systems. We create distance or locational parameters based on the alignment and merge relationships (Section 3.3). Therefore, the adaptable grid elements facilitate the linkage of building elements that are either joined or approximately aligned, thereby quantitatively representing the nature design logic.

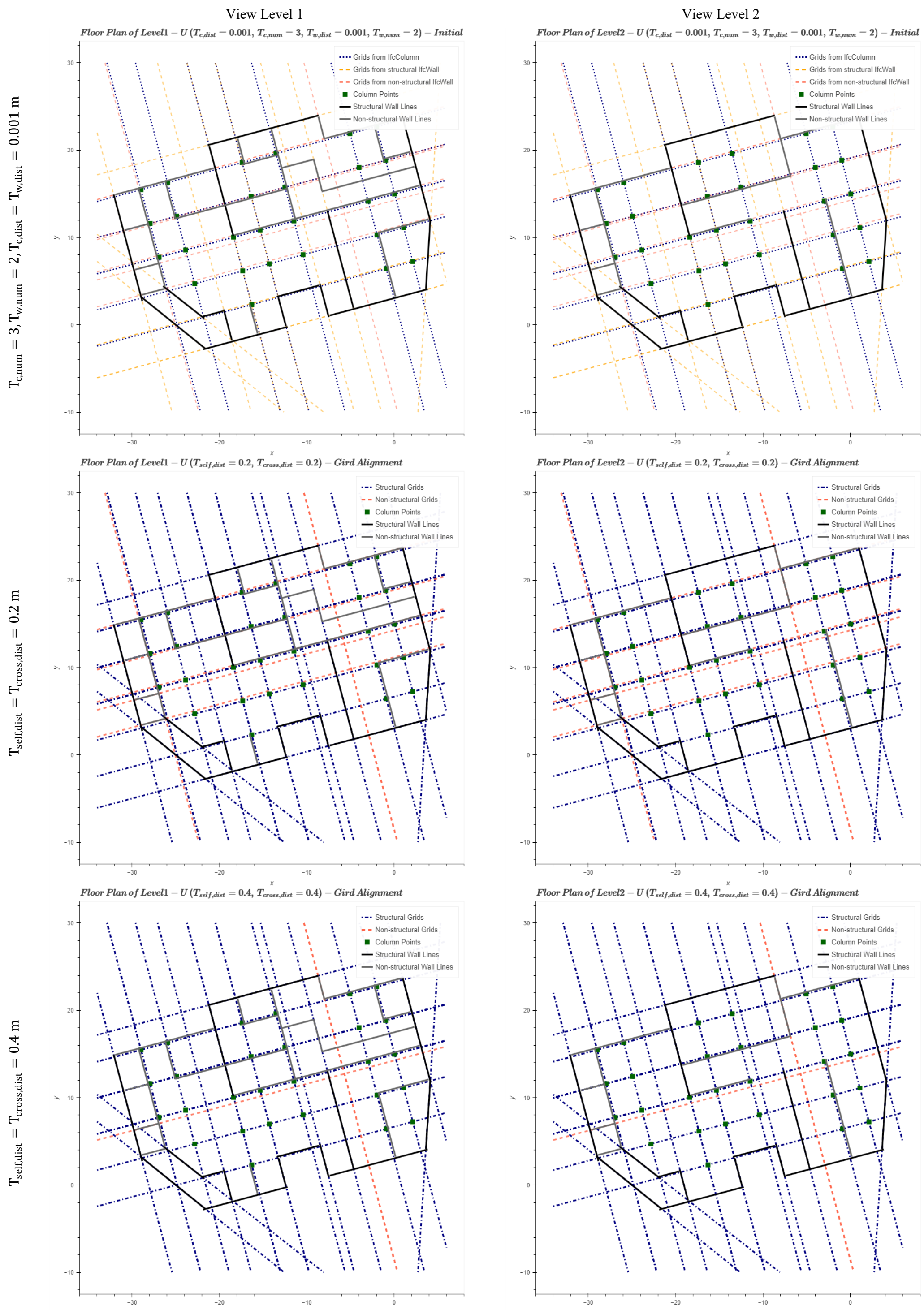

Fig. 11. Adjusted *IfcGrid* elements with different combinations of thresholds ($T_{self,dist}, T_{cross,dist}$).

As illustrated in **Fig. 12**, building elements are grouped into different clusters through the established distance parameters, each of which consists of elements linked to one specific reference, *IfcGrid*. Elements initially connected with the host *IfcGrid* are characterized by parameters bearing zero values. Furthermore, the integrated parametrics have implications for further elements such as *IfcDoor*, *IfcWindow*, and *IfcSpace*. In this context, the *IfcGrid* elements function as spatial references, enabling the determination of topological relationships and the identification of recurring design features. This 're-parametrization' process significantly enhances design adaptability, fostering a resilient parametric system.

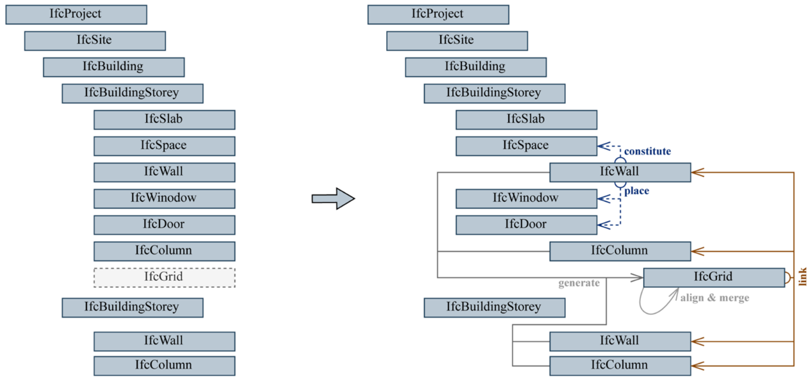

Fig. 12. The enriched IFC data model with grid logic. Solid lines: generation and adjustment of *IfcGrid*, direct linkage among *IfcWall*, *IfcColumn*, and *IfcGrid*; Dashed lines: building elements implicated by the identified relations (*IfcSpace*, *IfcWindow*, *IfcDoor*).

5 Conclusions

This research pioneers re-engineering the IFC data model by establishing reference grids, which is crucial for uncovering the spatial design logic. Firstly, it efficiently reveals the design logic inherent in the native model by extracting and analyzing the semantic and geometric data. Secondly, it represents the design rationale by incorporating the design parametrics with spatial elements of *IfcGrid* as foundational reference elements. The creation and alignment of reference grids offer a universal method for linking building elements, enabling more flexible design control and facilitating the adoption of the IFC data model for automated design adaptations. While the experimental focus was primarily on buildings with straight walls, the methodology's scope extends beyond rectilinear structures. It demonstrates proficiency in recognizing and aligning oblique structures as well. Doing so, this study explores the potential of investigating the IFC data model with a focus on intelligent parametric modeling from a design perspective.

Several limitations are acknowledged in this study. Firstly, there is a need for more advanced 3D spatial query methods to extract nuanced design logic [21], such as identifying building outlines and differentiating regions or zones. Secondly, striking an optimal balance between the flexibility and comprehensive coverage of design parametrics presents a significant challenge, as adjusted thresholds can lead to over- or under-parameterization. Additionally, the current approach focuses on linear layouts and excludes curved walls, requiring complex geometric computations and suggesting a path for future research. Finally, the outcomes of the proposed approach stay external to authoring tools, presenting integration challenges with standard BIM tools and indicating a need for further research to bridge this gap.

Acknowledgments

This research is supported by the TUM International Graduate School of Science and Engineering (IGSSE).

References

[1] A. Borrmann and V. Berkhahn, "Principles of geometric modeling," in *Building Information Modeling: Technology Foundations and Industry Practice*, Springer International Publishing, 2018, pp. 27–41. doi: 10.1007/978-3-319-92862-3_2.

[2] J. J. Shah and M. Mäntylä, *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*. John Wiley & Sons, Ltd, 1995.

[3] R. Sacks, C. Eastman, G. Lee, and P. Teicholz, "Collaboration and Interoperability," in *BIM Handbook*, John Wiley & Sons, Ltd, 2018, pp. 85–129. doi: 10.1002/9781119287568.ch3.

[4] buildingSMART, "Design Transfer View - buildingSMART Technical," buildingSMART. Accessed: Nov. 28, 2023. [Online]. Available: https://technical.buildingsmart.org/standards/ifc/mvd/

[5] Y. Ji, A. Borrmann, J. Beetz, and M. Obergrießer, "Exchange of Parametric Bridge Models Using a Neutral

Data Format," *J. Comput. Civ. Eng.*, vol. 27, no. 6, pp. 593–606, 2013, doi: 10.1061/(asce)cp.1943-5487.0000286.

[6] L. van Berlo, T. Krijnen, H. Tauscher, A. van Kranenburg, and P. Paasiala, "Future of the Industry Foundation Classes: towards IFC 5," *Proc. Conf. CIB W78*, no. April, pp. 123–137, 2021.

[7] R. Sacks, C. M. Eastman, and G. Lee, "Parametric 3D modeling in building construction with examples from precast concrete," *Autom. Constr.*, vol. 13, no. 3, pp. 291–312, 2004, doi: 10.1016/S0926-5805(03)00043-8.

[8] J. Wu, R. K. Dubey, J. Abualdenien, and B. André, "Model Healing: Toward a framework for building designs to achieve code compliance," in *Proc. of European Conference on Product and Process Modeling*, CRC Press, Mar. 2022, pp. 450–457.

[9] S. Benjamin, R. Christopher, and H. Carl, "Feature modeling for configurable and adaptable modular buildings," *Adv. Eng. Informatics*, vol. 51, no. November 2020, p. 101514, 2022, doi: 10.1016/j.aei.2021.101514.

[10] I. J. Ramaji and A. M. Memari, "Interpretation of structural analytical models from the coordination view in building information models," *Autom. Constr.*, vol. 90, no. February, pp. 117–133, 2018, doi: 10.1016/j.autcon.2018.02.025.

[11] buildingSMART, "Technical Roadmap 2020." Accessed: Dec. 07, 2023. [Online]. Available: https://www.buildingsmart.org/about/technical-roadmap/

[12] F. Noardo, K. A. Ohori, T. Krijnen, and J. Stoter, "An inspection of ifc models from practice," *Appl. Sci.*, vol. 11, no. 5, pp. 1–28, 2021, doi: 10.3390/app11052232.

[13] I. J. Ramaji and A. M. Memari, "Interpreted information exchange: Implementation point of view," *J. Inf. Technol. Constr.*, vol. 25, no. August 2018, pp. 123–139, 2020, doi: 10.36680/j.itcon.2020.008.

[14] J. Wu and J. Schlenger, "Enriching Building Graphs with Parametric Design Constraints for Automated Design Adaptation," in *34th Forum Bauinformatik*, 2023, pp. 234–241. doi: 10.13154/294-10116.

[15] M. Hammer, "Reengineering Work: Don't Automate, Obliterate," Harvard Business Review Home. Accessed: Dec. 10, 2023. [Online]. Available: https://hbr.org/1990/07/reengineering-work-dont-automate-obliterate

[16] Z. Geng and B. Bidanda, "Review of reverse engineering systems–current state of the art," *Virtual Phys. Prototyp.*, vol. 12, no. 2, pp. 161–172, 2017, doi: 10.1080/17452759.2017.1302787.

[17] A. I. Protopsaltis and I. Fudos, "A feature-based approach to re-engineering CAD models from cross sections," *Comput. Aided. Des. Appl.*, vol. 7, no. 5, pp. 739–757, 2010, doi: 10.3722/cadaps.2010.739-757.

[18] T. Várady, R. R. Martin, and J. Cox, "Reverse engineering of geometric models - An introduction," *CAD Comput. Aided Des.*, vol. 29, no. 4, pp. 255–268, 1997, doi: 10.1016/s0010-4485(96)00054-1.

[19] "Open CASCADE Technology." Accessed: Dec. 18, 2023. [Online]. Available: https://dev.opencascade.org/

[20] "IfcOpenShell - The open source IFC toolkit and geometry engine." Accessed: Dec. 11, 2023. [Online]. Available: https://ifcopenshell.org/

[21] Y. W. Zhou, Z. Z. Hu, J. R. Lin, and J. P. Zhang, "A Review on 3D Spatial Data Analytics for Building Information Models," *Arch. Comput. Methods Eng.*, vol. 27, no. 5, pp. 1449–1463, 2020, doi: 10.1007/s11831-019-09356-6.