Regulated Creativity

A Hierarchical Parametric Graph for Regulation-Aware Algorithmic Layout Design

Miguel Caroço[1], Tiago Farinha[2], Inês Caetano[3], António Leitão[4]
[1,2,3,4]INESC-ID/Instituto Superior Técnico, University of Lisbon
[1]miguel.caroco@tecnico.ulisboa.pt, [2] tiago.agostinho.farinha@tecnico.ulisboa.pt, [3]ines.caetano@tecnico.ulisboa.pt, [4] antonio.menezes.leitao@tecnico.ulisboa.pt

Abstract. *Early-stage architectural design requires negotiating functional intent, spatial relationships, and regulatory constraints while preserving creativity. However, existing frameworks often rely on post-hoc validation through external tools, hindering informed exploration. This paper proposes a hierarchical parametric graph representation that bridges abstract spatial reasoning, geometric generation, and regulation-aware exploration for housing projects. The method encodes spaces, adjacencies, and containment hierarchies as graph entities enriched with parametric attributes. Through a procedural pipeline, these configurations are translated into volumetric layouts, linking topological reasoning and geometric instantiation at early design stages. Unlike IFC-based models focused on instantiated elements, this approach represents spatial intent and regulatory logic independent of element visualization. Building regulations are formalized as evaluable rule-sets that query the graph, enabling automated compliance checking during design exploration. By coupling generation and evaluation, the framework allows designers to explore compliant alternatives iteratively. The approach is validated through a residential repurposing case study, demonstrating its capacity to reduce verification effort and support regulation-aware design in early stages. The paper concludes by discussing research contributions, limitations, and future work.*
Keywords. *Computational Design, Spatial Representation, Graph-Based Design, Parametric Design, Regulation-Aware Design, Early-Stage Architectural Design.*

INTRODUCTION

Early-stage architectural design requires balancing creative exploration with strict regulatory compliance. In residential buildings, building codes significantly influence spatial viability, yet traditional workflows treat compliance as a "post-hoc" verification. This mismatch often leads to late-stage revisions, as design intent is developed independently from legal constraints.

While Parametric Design (PD) supports the generation of spatial alternatives, current approaches fail to integrate regulatory logic as a design driver, keeping abstract spatial reasoning separated from real-time automated compliance. This paper addresses this gap by proposing a framework that encodes building regulations that are applicable to a hierarchical parametric graph. By transforming static regulations into evaluable logic filters, the system provides immediate feedback during the layout generation process. This approach embeds regulatory intelligence into the building's topological structure, enabling a workflow of *informed creativity* .

RELATED WORK

This section presents relevant research efforts addressing the topological representation of spaces and rule validation, along with their respective strengths and limitations.

Topologic representation

Graph theory abstracts architectural configurations, enabling the precise modelling of spatial hierarchies and relationships. Aish and Pratap (2013) introduced an approach that prioritized topological relationships over explicit geometric definitions, a logic later materialized via *Topologic* (Jabi et al., 2018). This non-manifold approach treats space as a system of interconnected cells rather than independent geometric units, allowing architects to perform spatial reasoning and explore circulation, visibility, and accessibility connections as a generative driver before committing to fixed physical dimensions (Aish et al., 2018).

Building on this, Xie and Ding (2023) use a spatial graph where nodes represent volumes and edges denote interconnectedness, including adjacency, intersection, and containment. Their Orthogonal Orientation Digraph (OODG) represents spaces through relative 3D positions, facilitating designer interaction by allowing design requirements to be set and evaluated throughout the design stages. However, as noted by the researchers, these models often lack integration with broader environmental and regulatory constraints, such as setbacks and lot-specific limitations.

In the context of high-rise residential buildings, Liu et al. (2024) introduced a framework centered on a Graph-Constrained Generative Adversarial Network (GC-GAN), which uses Knowledge Graphs (KGs) to explicitly model entities and attributes. This provides a richer data organization than relational and traditional file-based models. This aligns with the *TopologicPy* proposal (Jabi, 2025), where the graph goes beyond a mere wireframe by articulating geometry and design preferences, such as connectivity/circulation metrics, unit counts and sizes.

While these GAN-based frameworks, including *House-GAN++* (Nauata et al., 2021) and *Graph2Plan* (Hu et al., 2020), demonstrate significant performance gains, they often operate as "black boxes". As Liu et al. (2024) highlight, these approaches still rely heavily on the designer's experience to filter quality results and resolve latent topological conflicts. Furthermore, while Shekhawat and Pinki (2020) provide rigorous mathematical methods for translating adjacency graphs into floor plans, a gap remains in embedding evaluable regulatory rule sets directly into the hierarchical structure.

Rule validation

The transition from purely topological models to regulation-aware systems is a critical frontier in PD. While previous representations provide a framework for spatial generation, the utility of such models in a professional workflow depends on their capacity for automated code compliance checking. Current research primarily approaches this by processing existing Building Information Modeling (BIM) data into KGs for retrospective validation.

One example is the work by Nakhaee et al. (2024), where BIM data standardized through the IFC format is combined with unstructured regulatory data into a hybrid KG. Within the graph, building components, such as walls or stairs, are explicitly linked to their corresponding regulatory requirements, and compliance is checked via graph queries generated by Large Language Models (LLMs). This approach enables flexible, user-driven interrogation of regulatory constraints, illustrating the potential of combining KGs with AI. However, it requires querying regulatory rules individually, which hinders the evaluation of comprehensive regulation sets. Moreover, experimental validation remains limited in scope, focusing on fire protection requirements and resulting in a KG comprising only 14 nodes. This raises questions regarding the scalability of the approach to more complex, multi-layered regulatory frameworks.

KG regulatory representations are also applied in FU GPT (Feng et al., 2025), a conversational system built on top of a regulatory KG. Instead of performing automated compliance validation of an entire rule set, the system restricts interaction to a question-and-answer format. The system operates as an AI-powered chatbot for regulatory consultation, covering extensive regulatory corpus with an *F1-score* of 93%. While the *F1* score reflects a balance between correct and complete responses, the system remains limited to a question-and-answer interface rather than providing automated generative validation.

Similarly, Peng and Liu (2023) proposed a workflow where building information is first exported from BIM models using the IFC format. The extracted data then undergoes multiple intermediate file format conversions using *Protégé* (Bonduel et al., 2018), an IFC-to-RDF (Resource Description Framework) to reach a format compatible with semantic web technologies.

To bridge this gap, Yang et al. (2022) demonstrate the potential for automated semantics and topology representation specifically for residential building spaces. The authors argue that embedding evaluable rule sets directly into a hierarchical parametric graph enables *informed creativity,* treating compliance as a generative driver rather than a final hurdle. However, even these advancements often treat topology and regulation as two separate sequential steps.

COMPLYSCAN FRAMEWORK

Current workflows often suffer from a semantic disconnection where building regulations are verified only after the geometric model is fully instantiated. To bridge the gap between abstract spatial reasoning and automated regulatory compliance, this research proposes *ComplyScan*, a Julia-based framework where regulations are encoded algorithmically to support the spatial representation of buildings via parametric graphs. ComplyScan is implemented as a standalone validation engine that communicates with BIM authoring tools through a standardized data interface, allowing building information to be exported for rule evaluation by ComplyScan and the resulting compliance results to be returned to the modeling environment for visualization. This section details this approach, describing how the system accesses and formalizes regulatory rule sets, supports iterative interactions, and ensures user-friendliness.

Formulation of regulation rule sets

Any regulation consists of multiple rules. Instead of representing these as explicit nodes, the system treats them as implicit logic filters, avoiding data duplication and simplifying the representation of relational rules (i.e., those that enforce relationships between elements). For example, a rule requiring minimum wall thickness could be represented by attaching a separate rule to each wall node, but would create a heavy, redundant graph. Moreover, while it is straightforward to determine which element an individual rule targets, relational rules require information about other elements and thus cannot be evaluated independently.

One possible approach would be to evaluate relational rules at the first common parent in the graph hierarchy that contains all relevant elements as children. As an example, consider a rule that imposes a minimum combined area for all bathrooms. As each bathroom node does not have access to the attributes of other nodes, compliance checking must be done at the parent node level (the one containing all bathroom nodes). While feasible, this approach lacks scalability.

To ensure scalability, the system implements rule validation by executing each rule using the graph root node (the building itself) as the primary input. The rule then autonomously iterates over the graph's data to evaluate target elements. This strategy ensures a clean graph structure and allows rules to access any information within the building, surpassing the limitation of standard hierarchical traversal, where child nodes lack direct access to information from their parent nodes.

System architecture

The system operates as illustrated in figure 1**,** where the solid arrows indicate the sequence of steps executed by the program whenever changes are made to the building model. The dashed lines indicate the responsibilities that each component must fulfill prior to program execution. Rather than directly extracting information from the design tool model, the validation tool expects to receive the specific information required for the execution of the implemented compliance rules. The first step consists of the design tool transforming the building model data into the standardized building data required for regulation evaluation.

Different regulations rely on different subsets of building information to assess compliance. Instead of defining isolated data requirements for each specific regulation, the system aggregates these needs into a single unified data interface. As new rules are introduced, this interface is incrementally extended. Theoretically, over time, this process leads to functional convergence. Because many regulations reuse common building concepts (e.g., rooms, openings, circulation spaces, and materials), newly introduced rules frequently rely on attributes already present in the interface, reducing the need for structural modifications and promoting attribute reuse. In practice, design tools need to implement a single mapping to this shared representation, while new regulations just need to append missing attributes to the existing schema.

The next step is automated rule validation, where the building model is evaluated against registered regulations. The validation system identifies non-compliances, returning a mapping between non-compliant elements and their corresponding issues to the design tool. Each rule is interpreted from the corresponding regulation and implemented as code, containing the necessary checks to determine compliance. Finally, the design tool is responsible for rendering the building and visualizing the detected non-compliances within the user's modeling environment.

Conflict resolution

Beyond the detection of non-compliances, the system assigns each violation a *severity score* ranging from 0 to 1, reflecting the magnitude of the regulatory deviation. For example, a corridor width of 10 cm receives a higher *severity score* than one measuring 80 cm. The mechanism used to link building elements to reported issues is the assignment of unique *IDs*. Each element associated with a rule, such as a bathroom subjected to a minimum area requirement, must have a unique ID to ensure traceability within the feedback loop.

However, some violations arise from missing elements, where no direct ID can be assigned because the element does not exist. In these cases, the issue is associated with the ID of the relevant parent element within the hierarchy. For example, if an apartment floor contains no bathrooms, the non-compliance is linked to the ID of the floor itself.

Finally, it is the user's responsibility to ensure that the applicable regulations use the same measurement units as the building model.

User interaction and feedback loops

The design tool utilizes the provided IDs to visually present non-compliant elements to the user. To prevent information overload during the creative process, users can select specific regulations and

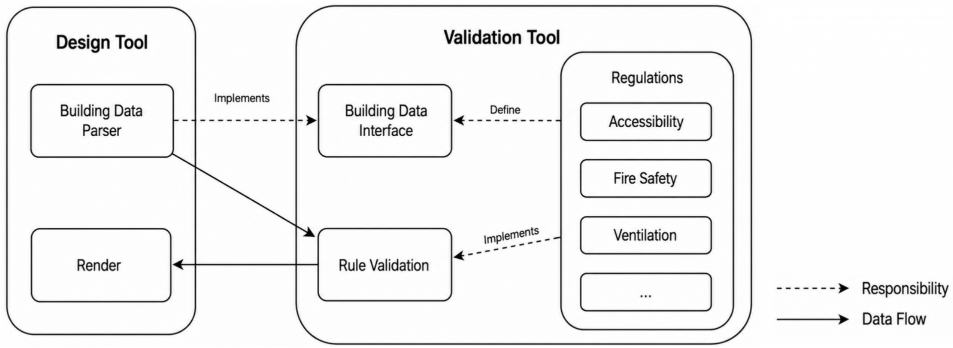


Figure 1
System architecture illustrating the interaction between the design tool (left) and ComplyScan framework (right). Building data is parsed and mapped to a standardized interface, validated against the implemented regulatory rules, and the resulting non-compliances are returned to the design environment for visualization.

toggle the display of non-compliant elements using checkboxes (Figure 2). This solution helps manage visual clutter when multiple issues are detected.

Additionally, the system only displays the first non-compliance detected per element, helping to keep feedback focused and avoid overwhelming the user. For example, if a bathroom fails both minimum area requirements and accessibility constraints, only the first issue is displayed, as addressing it often resolves subsequent violations.

One alternative would be ordering non-compliances by their *severity score*. However, *severity values* are not always directly comparable, as some rules produce multiple metrics rather than a single scalar, hindering consistent score comparison. For example, a rule enforcing a specific ratio between width and length may generate separate severity scores for each dimension,

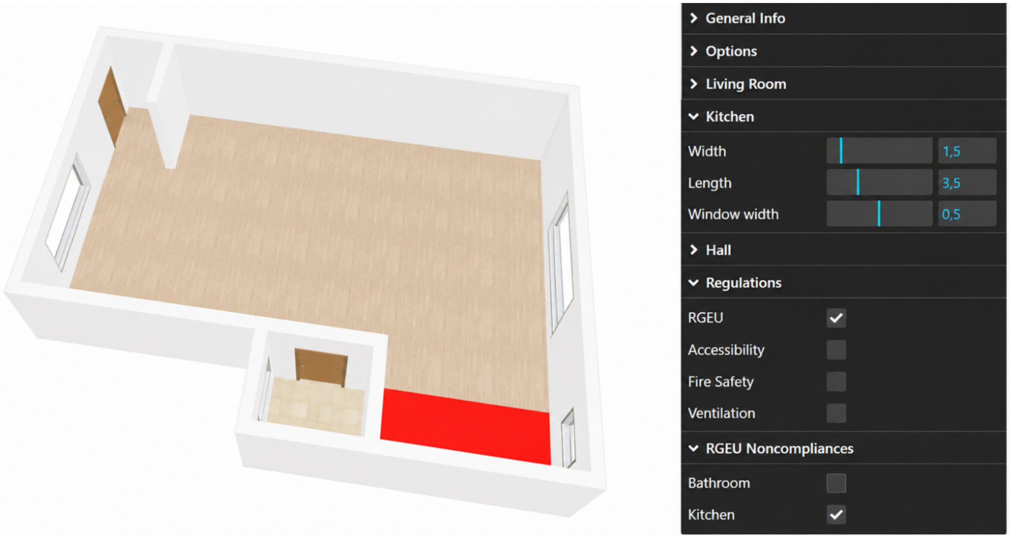


Figure 2
Validation of a zero-bedroom apartment layout against the Portuguese building regulation (RGEU). The right side menu allows the user to change dimensions and select the regulations to check in real-time, which are displayed in the design environment.

hindering a consistent ranking of violations.

Limitations

The proposed framework offers an alternative to post-hoc BIM validation but still presents some limitations. The first one arises from the mapping of building data between the design tool and the format required by the rule validation system, which may introduce conversion issues. Since the rule sets are encoded to process specific geometric types, such as polygons for area and maneuverability checks, any non-conforming geometry must undergo approximation. For example, to check whether a wheelchair has space to perform a 360º maneuver in a building entrance, the validation system expects the entrance geometry to be provided as a polygon. If the entrance is modelled using a different geometric entity, such as a semi-circle, there is no direct way to perform this validation. In such cases, the design tool must approximate the geometry, for instance by representing the semi-circle as a polygon with a sufficient number of sides to approximate its curved boundary. While this enables compliance checking, it introduces a localized semantic gap where the precision of the validation is tied to the quality of the geometric conversion rather than the designer's original intent.

Additionally, in cases where conflicting regulations impose mutually exclusive requirements, the system does not provide automatic conflict resolution. Therefore, users are responsible for selecting the appropriate regulatory set for the intended building evaluation.

Furthermore, the system's tool-independent architecture relies on a shared interface logic that, while avoiding the interoperability friction of multi-step file conversions (e.g., IFC-to-RDF), shifts the burden of implementation to the design tool. For the framework to be fully effective, the host software must implement robust data mapping for every required attribute. This also assumes a consistent semantic interpretation of building concepts (e.g., room types, circulation spaces, or building elements) between the design environment and the validation framework. Divergent classification schemes may therefore require additional mapping layers to ensure correct rule evaluation. In complex projects involving multiple overlapping regulations, the effort to maintain these interfaces can become significant, potentially limiting the system's agility in highly heterogeneous software environments. Although the framework promotes attribute reuse through a shared interface, the continued expansion of regulatory coverage may increase schema complexity and require careful management of interface definitions.

Finally, the current implementation simplifies feedback loops by displaying only the first detected non-compliance per element to avoid information overload during the creative process. While this prioritization helps maintain focus, it may temporarily mask secondary violations. Additionally, the use of a scalar *severity score* (0 to 1) simplifies complex spatial conflicts into a single metric, which may not fully capture the nuances of rules that generate multi-dimensional severity values.

CASE STUDY

The methodology presented in the previous section defines how regulations are formalized and integrated within the framework, addressing both design decisions and technical limitations. This section applies these concepts in practice through a case study involving the Portuguese building regulation—*Regulamento Geral das Edificações Urbanas* (RGEU). The selected PD tool allows users to configure the properties of a residential unit, such as spatial typology, room dimensions, and the positioning of openings.

To demonstrate the system in operation, a simple example is presented focusing on the minimum area requirements. In RGEU, a kitchen must have an area greater than or equal to 6 m^2, while a bathroom requires at least 3.5 m^2. Figure 3 illustrates the visual feedback loop: the left example

Figure 3
Visual interface for rule validation: In the right-side example, the kitchen is highlighted in red with a descriptive label indicating an area violation; in contrast, the left-side example shows no annotations, because all areas comply with the selected regulations.

shows a compliant layout version of a no-bedroom apartment, that is, where all rules from RGEU are satisfied, with a clear status menu; in contrast, the right example identifies non-compliance in the kitchen of the same apartment through red floor highlighting and descriptive labels. It also shows the system's user-controlled filtering: although the bathroom is also non-compliant, its feedback can be hidden via the checkbox interface to manage visual clutter. Although this example focuses on minimum area requirements for demonstration purposes, the implemented framework also supports rules involving wall thickness requirements, minimum numbers of rooms by type, accessibility constraints, minimum window areas, and other regulatory conditions.

To evaluate the system's performance in a complex, real-world architectural context, the *ComplyScan* framework was applied to the renovation of *Quarto Palazzo*, an abandoned office building in San Donato Milanese, Italy, from the 1980s. This project repurposes the existing structure into a dwelling complex aimed at addressing the local housing situation.

The apartments are organized into modular clusters designed to adapt to the building's original structural grid. The primary architectural challenge lies in the building's long span, almost 27m in width, which requires the strategic insertion of winter gardens within each apartment to ensure adequate access to natural light and ventilation. In this scenario, the system validates the internal spatial configurations of these modular apartments as they are repeated and adapted throughout the building. As the designer manipulates the modular clusters across each floor, the hierarchical parametric graph ensures that every individual unit remains compliant with area requirements. Figure 4 illustrates the modular layouts developed for the *Quarto Palazzo* repurposing project, showing the relationship between the functional spaces of each apartment and the integrated winter gardens. The figure displays real-time area data and validation results, confirming whether or not key apartment spaces meet the required regulatory thresholds through immediate visual verification.

The complexity of this repurposing project requires constant iteration of room dimensions

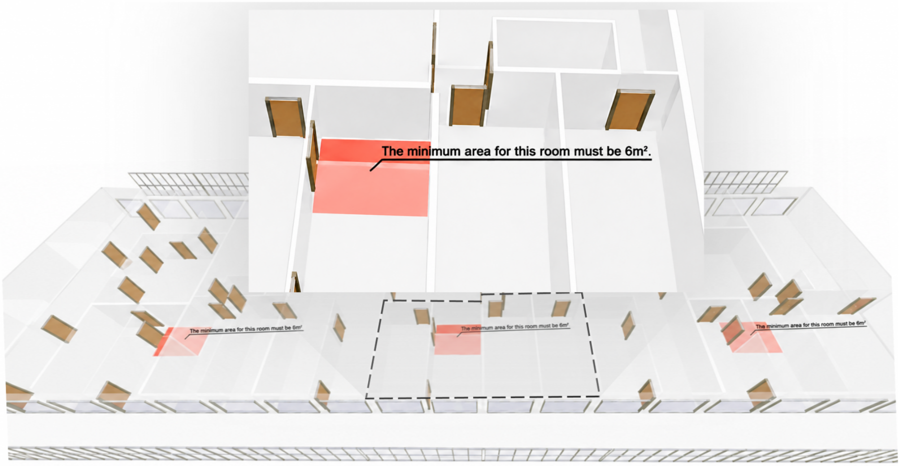


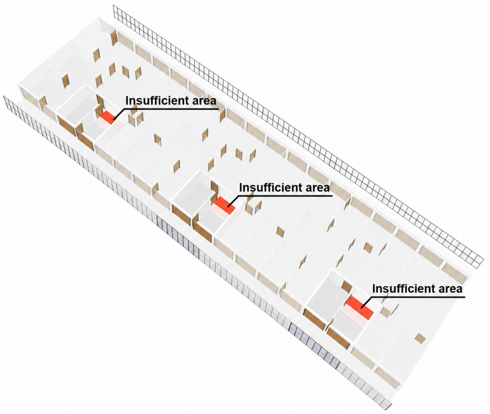
Figure 4
Modular apartment layouts for the Quarto Palazzo repurposing project, with non-compliant areas highlighted in red.

to accommodate existing structural spans. The framework supports this by providing real-time feedback on the internal areas of the dwelling units. For instance, if a winter garden is expanded to improve the apartment's environmental quality, the system immediately flags if the adjacent kitchen or living area falls below the minimum thresholds (6 m^2 or 10 m^2, respectively) defined by the regulation. Similarly, if a new piece of furniture is added to a room in a way that obstructs the space required for a wheelchair to perform a 360º rotation maneuver, the system immediately detects and flags the resulting accessibility violation. This allows the designer to treat compliance as a generative boundary, ensuring that while the building is repurposed for a new use, the internal spatial quality of every modular unit remains aligned with legal constraints. Figure 5 presents a floorplan configuration for different dwelling units and circulation cores, highlighting how the individual compliant modules aggregate into the larger building scale. As visible in the figure, the system enables users to filter the visualization to highlight only non-compliant apartments, while maintaining a clear view of those that satisfy all regulations. This feature reduces cognitive load and allows the designer to isolate and address specific topological or area violations within a complex building layout.

RESULTS AND DISCUSSION

The application of the framework to the test cases demonstrated that compliance checking can be performed in real time. As users modify the building by adjusting dimensions or changing spatial configurations, the system continuously evaluates the design against the applicable regulations and immediately identifies any non-compliant elements. This real-time feedback allows users to understand the impact of their design decisions instantaneously, rather than relying on a separate validation step after the design is completed. Non-compliances are presented in a clear and structured manner, enabling users to quickly identify which elements are not compliant and why. Overall, this approach supports an interactive design process where compliance is continuously monitored, reducing the accumulation of errors and facilitating iterative refinement while maintaining alignment with regulatory requirements.

Figure 5
The building floor plan showing the system's ability to selectively highlight only non-compliant apartments through automated area validation.



The application of the framework to the Quarto Palazzo repurposing project further demonstrated its ability to support iterative design processes in a realistic architectural context. As apartment modules were adapted to the constraints imposed by the existing structural grid, the system continuously verified compliance with a predefined regulation, allowing spatial adjustments to be evaluated without interrupting the design workflow. This suggests that the proposed approach can support regulatory feedback not only in simplified examples but also in larger adaptive reuse projects involving repeated dwelling configurations.

FUTURE WORK

Several aspects of the current system open directions for future development. One important area is scalability to other building typologies, such as apartments with multiple bedrooms, hospitals, or schools. Different typologies involve different, demanding regulatory frameworks and specific spatial configurations. While a larger number of building elements must be checked, computational growth is expected to be approximately linear, as each rule operates independently and most of them scale proportionally with the number of relevant elements.

Another relevant topic is the resolution of conflicts between rules and parametric results. In practice, different rules may impose constraints that are difficult to satisfy simultaneously, and adjustments made to resolve one non-compliance may inadvertently introduce new violations elsewhere. Future work should explore mechanisms to automatically generate compliant buildings for the user.

Furthermore, alignment with industry standards such as IFC-spaces represents an important step towards interoperability. By creating a data mapping from IFC-based representations to the system's data, it becomes easier to integrate with existing construction documentation workflows, reducing duplication of effort and increasing the practical applicability of the system.

CONCLUSION

This paper presented *ComplyScan*, a framework for integrating regulatory compliance into the early stages of architectural design, formalizing building codes as evaluable logic filters rather than post-hoc validation checks. The proposed framework bridges the gap between abstract spatial reasoning and automated rule enforcement via a transparent, rule-based approach that treats compliance functions as a generative driver rather than a retrospective technical hurdle. The implementation of a tool-independent interface demonstrates that regulatory intelligence can be applied directly over the topological structure of a building, reducing the need for complex file conversions and ensuring that design intent remains aligned with legal constraints from the earliest conceptual phases.

ACKNOWLEDGMENTS

Work supported by national funds through *Fundação para a Ciência e a Tecnologia*, I.P. (FCT) under projects UID/50021/2025 (DOI: https://doi.org/10.54499/UID/50021/2025) and UID/PRR/50021/2025 (DOI: https://doi.org/10.54499/UID/PRR/50021/2025).

REFERENCES

Aish, R., Jabi, W., Lannon, S., Wardhana, N. M., & Chatzivasileiadi, A. (2018). Topologic: Tools to explore architectural topology. *AAG 2018: Advances in Architectural Geometry*, 316–341.

Aish, R., & Pratap, A. (2013). Spatial Information Modeling of Buildings Using Non-Manifold Topology with ASM and DesignScript. *Advances in Architectural Geometry 2012*, 25–36.

Bonduel, M., Oraskari, J., Pauwels, P., Vergauwen, M., & Klein, R. (2018). The IFC to Linked Building Data Converter - Current Status. *Proceedings of the 6th Linked Data in Architecture and Construction Workshop*. https://github.com/jyrkioraskari/IFCtoRDF-Desktop

Feng, T., Yuan, Z., Xu, X., He, Z., Li, W., & Yuan, P. F. (2025). Integrating Regulatory Compliance In AI-Assisted Architectural Design. *CAAD Futures 2025 – Catalytic Interfaces: Conference Proceedings, Vol. I*. https://doi.org/10.25442/hku.29349179

Hu, R., Huang, Z., Tang, Y., Kaick, O. V., Zhang, H., & Huang, H. (2020). Graph2Plan: Learning Floorplan Generation from Layout Graphs. *ACM Transactions on Graphics*, *39*(4). https://doi.org/10.1145/3386569.3392391

Jabi, W. (2025). TopologicPy: A Syntopic Integration of Geometry, Topology, and Semantics in Architectural Design. *Formal Methods in Architecture. FMA 2024. Digital Innovations in Architecture, Engineering and Construction*. https://doi.org/10.1007/978-3-032-02782-5_1

Jabi, W., Aish, R., Lannon, S., Chatzivasileiadi, A., & Wardhana, N. M. (2018). Topologic: A toolkit for spatial and topological modelling. *Computing for a Better Tomorrow - Proceedings of the 36th eCAADe Conference - Volume 2*, 449–458. https://doi.org/10.52842/conf.ecaade.2018.2.449

Liu, J., Qiu, Z., Wang, L., Liu, P., Cheng, G., & Chen, Y. (2024). Intelligent floor plan design of modular high-rise residential building based on graph-constrained generative adversarial networks. *Automation in Construction*, *159*, 105264. https://doi.org/10.1016/j.autcon.2023.105264

Nakhaee, A., Elshani, D., & Wortmann, T. (2024). A Vision for Automated Building Code Compliance Checking by Unifying Hybrid Knowledge Graphs and Large Language Models. *Scalable Disruptors. DMS 2024*. https://doi.org/10.1007/978-3-031-68275-9_36

Nauata, N., Hosseini, S., Chang, K.-H., Chu, H., Cheng, C.-Y., & Furukawa, Y. (2021). House-GAN++: Generative Adversarial Layout Refinement Network towards Intelligent Computational Agent for Professional Architects. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13632–13642. https://ennauata.github.io/houseganpp/page

Peng, J., & Liu, X. (2023). Automated code compliance checking research based on BIM and knowledge graph. *Scientific Reports*, *13*(1). https://doi.org/10.1038/s41598-023-34342-1

Shekhawat, K., & Pinki, P. (2020). Construction of Architectural Floor Plans for given Adjacency Requirements. *RE: Anthropocene, Design in the Age of Humans - Proceedings of the 25th CAADRIA Conference - Volume 1*, 315–323. https://doi.org/10.52842/conf.caadria.2020.1.315

Xie, X., & Ding, W. (2023). An interactive approach for generating spatial architecture layout based on graph theory. *Frontiers of Architectural Research*, *12*(4), 630–650. https://doi.org/10.1016/j.foar.2023.04.003

Yang, B., Jiang, T., Wu, W., Zhou, Y., & Dai, L. (2022). Automated Semantics and Topology Representation of Residential-Building Space Using Floor-Plan Raster Maps. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*, 7809–7825. https://doi.org/10.1109/JSTARS.2022.3205746