2026 European Conference on Computing in Construction
Corfu, Greece
July 12-15, 2026

DEVELOPING A LINKED-DATA-BASED FRAMEWORK FOR REGULATION COMPLIANCE CHECKING IN ARCHITECTURAL DESIGN

Yutian Yi[1], Davide Lombardi[1], Zhelun Zhu[1]and Theodoros Dounas[2]
[1]Xi'an Jiaotong-Liverpool University, Suzhou, China
[2]Heriot-Watt University, Edinburgh, Scotland

Abstract

Manual regulation compliance checking remains common in architectural design, while BIM-based and semantic approaches operate outside the authoring environment, requiring specialist expertise, delaying feedback in iterative work. This paper presents a lightweight compliance-checking framework implemented as a pyRevit extension and prototyped in Autodesk Revit. The framework extracts model parameters, converts them to RDF aligned with ifcOWL and a project namespace, and validates constraints using Shapes Constraint Language. The case study shows non-compliances can be detected and reported via both documentation outputs and in-model graphical feedback. The contribution is an auditable linked-data mechanism for extensible regulation checking within BIM authoring tools.

Introduction

The Architecture, Engineering and Construction (AEC) industry is characterised by increasing project complexity, multidisciplinary collaboration, and fragmented information flows across design, approval, and construction stages (Michaud et al., 2019). Within this context, building regulations play a critical role in guaranteeing construction quality and safety. However, regulation compliance checking during architectural design remains largely manual and error-prone (Preidel & Borrmann, 2016). Architects frequently encounter situations where design proposals fail to satisfy specific code requirements because of misinterpretation, omission, or inconsistent misunderstanding of regulatory clauses. In addition, interpreting building regulations can often be subjective, as different practitioners may interpret the same clause in varying ways (Zheng et al., 2022). Such subjectivity often leads to drawings being returned by local authorities for rectification, triggering multiple cycles of revision. Furthermore, due to multiple revisions, data loss and format error might also occur. This extends the construction schedule and incurs additional consumption of labour and materials, as well as costs (Mäki & Kerosuo, 2015). Such ambiguity increases the risk of inconsistent compliance decisions and potentially opens up space for corruption. As a result, under such circumstances, designers struggle to enhance their work efficiency. Recently, smart supervision, which is defined as the application of digital technologies such as sensors and automation to monitor construction activities, has been increasingly adopted to enhance transparency and control (Mhmoud Alzubi et al., 2023). However, research and implementations keep focusing on construction and operation stages, leaving the design stage insufficiently supported. This gap highlights the need for a digital workflow that embed regulation checking directly into architectural design environments, enabling proactive and consistent compliance verification.

This study aims to build a direct link between architectural models and regulatory clauses by representing them in object-oriented and machine-readable forms. These encoded rules can be processed automatically, ensuring consistent and objective compliance checking directly within the Building Information Model (BIM) environment. Thus, this study develops a linked-data-based workflow that digitizes local building codes, automatically extracts corresponding element data from building model, and cross-checks design information against regulatory constraints. The resulting data is structured in a machine-readable format, facilitating the information interoperability and process automation (Zhong et al., 2019). Moreover, the framework is designed without requiring the users to possess advanced computer skills. All processing procedures are completed by providing the appropriate input data. The overarching goal of this research focuses on the minimisation of human errors, the prevention of wasted efforts from information discrepancies, and shortening the pre-construction process.

The remainder of the paper is organised as follows. In section two a brief overview of related work is presented so as to clearly identify the research gap. In section three, the linked-data based framework is introduced in detail. The framework is structured in four parts: code conversion, information extraction and semantic expression, and user-oriented performance. Section four validates the feasibility of the proposed framework by a real case, and section five provides the conclusion and highlights the future work emerging from this study.

Related Work

This study focuses on linked-data and automated compliance checking during architectural design phase.

The following sections elaborate on previous research of the relevant aspects.

Automated compliance checking

Automated Compliance Checking (ACC) is the process of using digital asset models and encoded contractual rules to automatically verify that the design model and as-built asset satisfy commissioning and operational readiness requirements, thereby enabling rapid and consistent compliance assurance (Dimyadi & Amor, 2013). Eastman et al. (2009) first summarised the rule checking process into four phases: Rule Expression Phase, Model Data Preparation Phase, Rule Enforcement Phase, and Check result reporting phase. Two main data sources are indispensable during a typical ACC process, namely the building model and normative knowledge (Amor & Dimyadi, 2021).

Recent papers show ACC progressing from rule execution prototypes toward more robust rule interpretation and institutional adoption. Early work demonstrated ACC for topologically complex requirements by combining model preparation with automated rule evaluation and clear reporting of non-compliances (Bloch et al., 2019). Building on this idea, Bloch et al. (2024) proposed graph-based semantic enrichment to improve robustness beyond hard-coded rule scripts. Doukari et al. (2021) proposed object-centred checking workflows and plug-in implementations that structure the pipeline from requirement definition through execution and result presentation. Furthermore, the recent growth trend indicates that many studies have applied artificial intelligence technology to compliance checks. Zhang & El-Gohary (2019) extract hierarchy information from building codes by dependency parsing based on deep neural network model training. Chen et al., (2024) leverages deep learning and Large Language Models to categorize categorise regulatory documents and conduct a preliminary screening of the text. From a regulatory standpoint, Fuchs et al., (2025) recognises that fully automated decisions may not be acceptable in regulatory contexts, and the need for human oversight still remains, thus, there is need for traceability and transparency. Being able to explain why a certain element passed or failed compliance plays a pivotal role particularly in audits. For better scalability and information sharing, Zhong et al., (2018) developed specific ontologies to represent varying building environmental information and interact it with SPARQL formed clauses.

Linked-data model

The vision of automated compliance checking in the AEC industry has long been limited by interoperability issues. In this vein, the Linked Data paradigm, built on Semantic Web technologies, presents a promising solution for meaningful data integration and machine interpretation (Boje et al., 2020). A key development is ifcOWL, a web ontology language of the IFC schema, which allows BIM models to be represented as Resource Description Framework (RDF) graphs, transitioning data from a closed format to an open, queryable knowledge graph (Pauwels & Terkaj, 2016; Tomczak, 2025). This semantic enrichment facilitates precise representation and reasoning about building information, enabling regulatory rules to be expressed in complex terms. The power of linked data is further demonstrated by its ability to integrate diverse data sources. For instance, the BRICK ontology offers a standardised schema for representing building equipment and their relationships within building automation systems, facilitating interoperable data exchange across platforms (Balaji et al, 2018; Brick Consortium, n.d.). Cerovšek & Omar (2025) illustrate how IFC-based semantic data can be aligned with operational datasets structured using BRICK, highlighting the role of RDF as an architectural layer for interoperability. Donkers et al. (2023) illustrate the dynamic interconnection of RDF data across knowledge domains through web-based applications.

Querying and validating these interconnected RDF graphs require specialised languages. For the specific task of regulation checking, which is inherently a validation process, the Shapes Constraint Language (SHACL) is applied as a standard for validating asserted RDF data, and was published as a W3C (World Wide Web Consortium, the main international standards body for the Web) Recommendation in 2017 (W3C, 2017). SHACL is designed for validating "what must be true" (Heimsbakk & Torkelsen, 2023). This provides a more declarative and direct method to formalise building codes as constraints against the ifcOWL representation. However, a significant gap remains between linked-data-based compliance checking and everyday architectural practice, as many research still depend on manual export-conversion-validation pipelines which interrupt iterative modelling and discourage routine use by design teams. In this study, lightweight refers to an operationally lean workflow that avoids full-model exchange and instead extracts only the minimum set of code-relevant elements and attributes required for checking selected quantitative clauses. In other words, the challenge is no longer how to represent codes and BIM data as knowledge graph, but how to configure and operationalise them as a live, in-context compliance helper inside BIM authoring environments. This leads to two research questions:

RQ1: What lightweight knowledge-graph representation and constraint configuration is sufficient to support reliable checking of quantitative residential-code clauses during design iterations?

RQ2: How can this linked-data checking pipeline be embedded into a BIM authoring workflow to provide real-time feedback for architects without requiring expert coding skills?

To answer these questions, this study bridges the gap by integrating ifcOWL-based data representing and SHACL validation directly within Autodesk Revit through a pyRevit extension which is an open-source add-in framework that enables custom Python-based tools to run directly inside Autodesk Revit, thereby turning semantic checking into a designer-centric tool rather than a separate post-processing task.

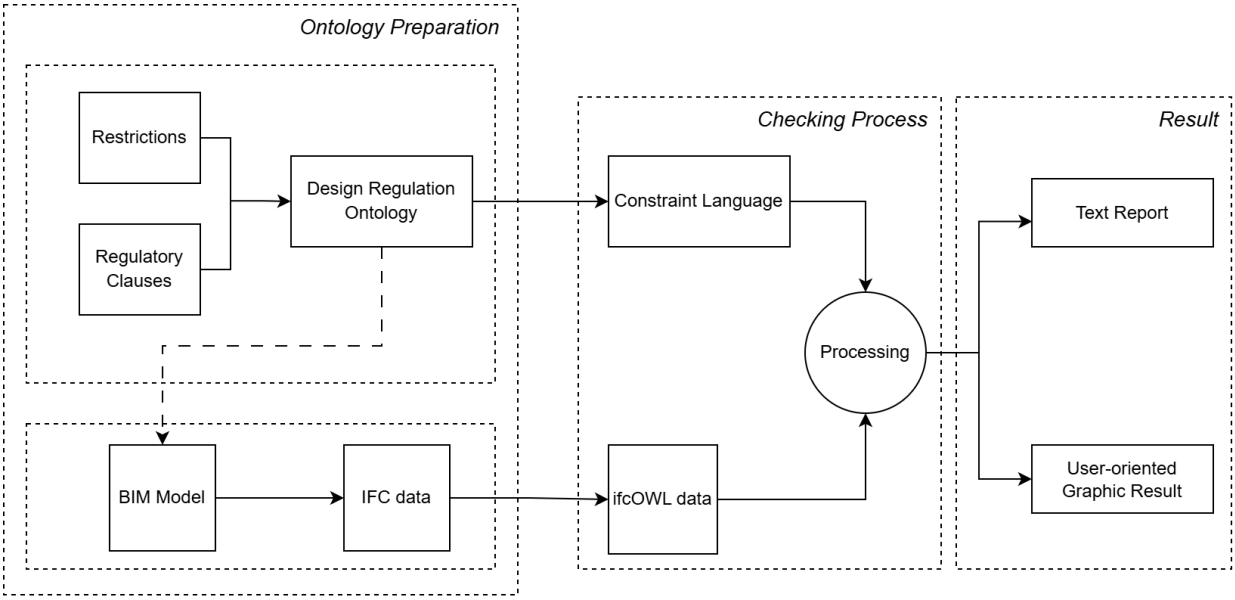

Figure 1: Proposed linked-data-based framework of automated compliance checking

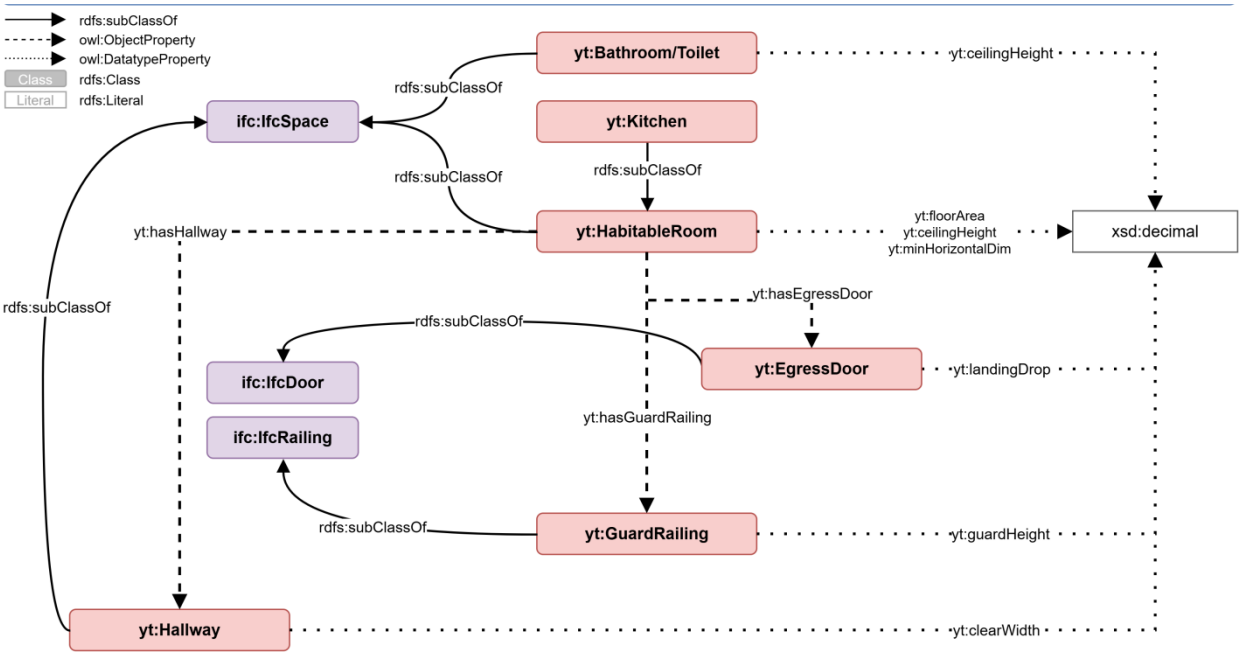

Figure 2: Snippet of the building regulatory ontology schema aligned with ifcOWL

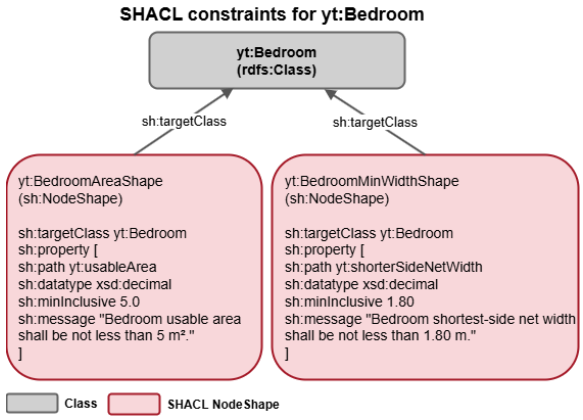

Figure 3: Example of SHACL rule definitions for compliance checking

Methodology

Scope definition and code sources

This study develops a flexible regulatory-checking framework aimed at the architectural design phase, where design teams need real-time, in-context feedback without relying on external consultants or expert programmers. To demonstrate feasibility of embedding code reasoning into the BIM operational environment, a subset of clear and quantitative requirements was extracted and encoded from the 2025 Edition China Residential Code (CRC2025), and it can be found on Github[1].

The framework is intentionally modular. The normative knowledge base acts as a replaceable code database which can be expanded or localised to other jurisdictions by updating the SHACL constraints and the associated regulatory terms. Each rule is expressed in a machine-readable format, enabling scalable adaptation as additional requirements are introduced and/or use of the framework for other regulations. By limiting the initial implementation to well-defined, quantitative provisions, the framework validates the workflow logic while ensuring its utilisation for non-technical design practitioners. This infrastructure provides a foundation for future inclusion of more complex and qualitative compliance criteria through the same linked-data structure.

Linked-data-based framework

This study proposes a linked-data based framework (shown as Fig 1.) for regulation compliance checking that is conceptually independent of any single BIM authoring platform, while being implemented and validated through a Revit-based prototype. The core idea is to decouple normative knowledge from software-specific model representations by first translating regulatory clauses into a regulatory ontology and then aligning model data with this ontology through shared semantic structures.

In the proposed framework, only the minimum required semantic information is extracted from the active BIM model and transformed into RDF with explicit mappings to ifcOWL and a project-specific regulatory ontology. This semantic abstraction enables regulation checking to be executed through SHACL validation at the ontology level, rather than being hard-coded to a specific BIM API. While the current prototype realises the workflow through a pyRevit extension, providing visual feedback, tabular summaries, and auditable reports within Revit, the same semantic rule set and validation logic can, in principle, be applied to models organising from other BIM environments that support IFC-based data exchange. Therefore, the use of IFC/ifcOWL and SHACL in this study is not intended to demonstrate cross-platform interoperability empirically, but to establish a software-agnostic regulatory reasoning layer, with Revit serving as a representative design environment for proof-of-concept validation.

[1] https://github.com/Yutian0425/BuildingRegulations-Ontology/tree/main

Encoding the regulatory clauses with ontology

The proposed framework translates selected quantitative clauses from the China Residential Code into a machine-interpretable regulatory ontology aligned with ifcOWL. Each clause is formalised as constraints over (i) a target building element class (e.g., space, door, railing) and (ii) measurable attributes (e.g., area, clear width, ceiling height). In this way, legal requirements are expressed as explicit semantics that can be consistently validated over BIM-derived data.

Figure 2 presents a subset of the developed ontology. The visual schema in Figure 2 follows the Ontology Design Template by Donkers (2022) to ensure consistent notation and readability. The schema reuses ifcOWL classes to represent BIM entities (e.g., ifc:IfcSpace, ifc:IfcDoor, ifc:IfcRailing) and extends them through a lightweight project namespace “yt:” (Yutian Yi), publicly released on GitHub for reuse, extension, and transparent review. Domain-specific regulatory concepts such as “yt:HabitableRoom”, “yt:EgressDoor”, and “yt:GuardRailing”. They are defined as subclasses of their closest IFC counterparts, preserving interoperability while allowing code-oriented distinctions. Quantitative requirements are then attached through datatype properties (e.g., “yt:floorArea”, “yt:clearWidth”, “yt:ceilingHeight”) whose values are represented as xsd:decimal. Encoding the regulatory knowledge in RDF triples ensures the resulting rule vocabulary is linked, queryable, and reusable across projects, enabling a traceable connection between legal clauses and digital building entities.

In-model data extraction

Within Autodesk Revit, a dedicated pyRevit extension automates extraction of compliance-relevant information directly from the authoring model. Using the Revit API, the extension identifies target element categories (e.g., rooms/spaces, doors, railings) and retrieves only the parameters required by the rule set, avoiding unnecessary geometry export and reducing operational overhead during iterative design.

To support rule execution and semantic alignment, each exported field is annotated with two layers of correspondence. First, an ifcOWL-oriented mapping associates the extracted elements with established AEC semantics (e.g., Room is mapped to ifc:IfcSpace, BalconyRailing is mapped to ifc:IfcRailing, Door is mapped to ifc:IfcDoor). Second, a regulatory mapping links the same elements and attributes to the project’s “yt:” ontology introduced above (e.g., habitable-space attributes to “yt:floorArea”, “yt:minHorizontalDim”, “yt:ceilingHeight”; egress-related attributes to “yt:landingDrop”; guard attributes to “yt:guardHeight”). This dual annotation enables reuse of standardized IFC semantics while preserving the code-specific vocabulary needed for regulatory checking. The extracted data are serialized as a compact JSON-LD payload, where numeric values are preserved alongside an explicit

semantic context that binds each field to ifcOWL and the "yt" regulatory vocabulary. This lightweight linked-data serialization avoids exporting a full IFC model while maintaining machine-interpretable interoperability across tools and rule sets.

Conversion to RDF graphs

The extracted dataset is first serialised as JSON-LD, where every Revit element assigned a stable identifier (e.g., "yt:Room_101") and an explicit semantic context that maps fields to the "yt" regulatory vocabulary and the ifcOWL schema. This means the payload is already a Linked Data representation rather than a purely syntactic JSON export. The workflow then materialises the JSON-LD into an RDF graph (and, when required by downstream tools, into a Turtle .ttl file). In this graph, element instances are represented as resources, attributes are expressed as RDF triples, and their conceptual types are connected to ifcOWL via rdf:type assertions. The conversion step automatically binds the "ifc:" and "yt:" namespaces and produces a standards-compliant RDF dataset for subsequent SHACL validation.

This semantic representation merges two domains that are usually separated in practice: BIM authoring data and rule-based knowledge graphs. By doing so, the building model can participate in the web of data—each wall, door, or space is no longer merely a 3D object but a node in a logically connected graph that can be validated, queried, and extended. The RDF layer therefore acts as a bridge between design tools and semantic reasoning technologies, while JSON-LD provides a lightweight and interoperable carrier format for moving data across components of the workflow.

Rules checking process via SHACL

Compliance checking is executed using the SHACL, which interprets the ifcOWL-encoded model as building model and normative knowledge as shape constraints. Each regulatory clause corresponds to a SHACL shape. Fig 3. presents an example, a "yt:HabitableRoomShape" enforcing sh:minInclusive 6.5 for "yt:floorArea", which means habitable room area must be greater than or equal to 6.5 square meters.

The SHACL engine (PySHACL library) parses the RDF graph, matches each instance of the target class, and then checks the relevant properties against the specified numeric thresholds. Non-conforming elements generate validation reports with precise pointers to both the violating node and the specific rule.

As both the data and the rules share the same ifcOWL backbone, reasoning occurs at the semantic level rather than through file-specific attributes. This alignment allows the framework to extend seamlessly to other jurisdictions or to add higher-order logic by simply appending new SHACL shapes.

Result Presentation

After validation, results are returned through two synchronised channels. On one hand, a textual report and an Excel summary are generated, listing all items and highlighting non-compliant element with its property values and the corresponding rule citation. These documents are automatically saved to the designer's local workspace and can be used for later verification or regulatory submission.

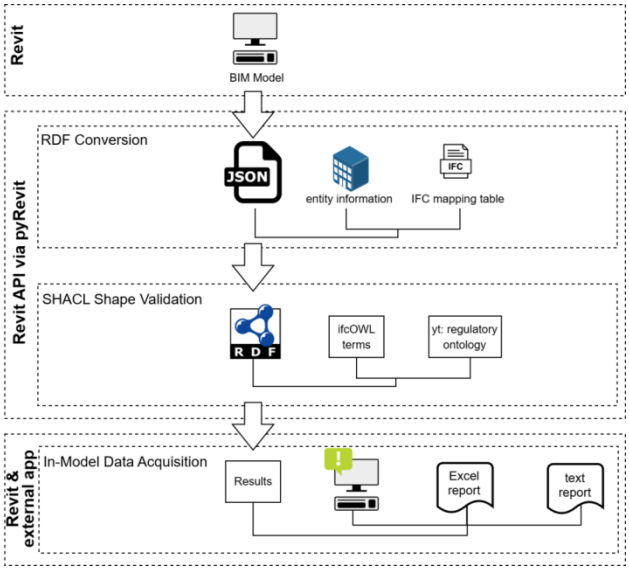

Figure 4: Overall data flow path of the prototype

More importantly, the pyRevit interface presents the feedback directly within the Revit user-side model. The script parses the SHACL report, identifies the offending element IDs, and visually highlights them in the active view using colour coding and annotations. This live feedback loop turns regulation checking into an interactive design activity, where architects can correct issues while modelling rather than after exporting or outsourcing verification. Fig 4. illustrates the data path across the whole framework. Through this integration, the traditionally disconnected processes of design and compliance merge together into a single continuous workflow governed by linked data semantics.

Case study

To demonstrate the feasibility and practical value of the proposed framework, a case study was conducted on a residential project comprising two-family dwellings currently under design development. The project type aligns directly with the applicability scope of the CRC2025, which governs low-rise residential buildings. The model was developed in Autodesk Revit 2026, and the pyRevit extension created in this study was used to perform data extraction, ontology conversion, and regulatory validation within the same digital environment.

Figure 5: The button of the developed pyRevit plug-in and the chosen case model

The chosen project represents an early-stage design model that is still undergoing iterative refinement, allowing the proposed framework to demonstrate its usefulness in real-time design evaluation rather than retrospective verification. Through the developed pyRevit plug-in shown in Fig 5., spatial and dimensional attributes of each building component were exported, converted into an RDF graph based on the ifcOWL schema and the extended "yt" ontology, and validated against the SHACL constraint set derived from local codes.

By suitable building model and normative knowledge input, the validation process successfully identified non-compliance instances, confirming the framework's capability to automatically trace code violations. For example, warnings in the SHACL validation report indicated in Fig 6.: usableArea: Missing or 0.

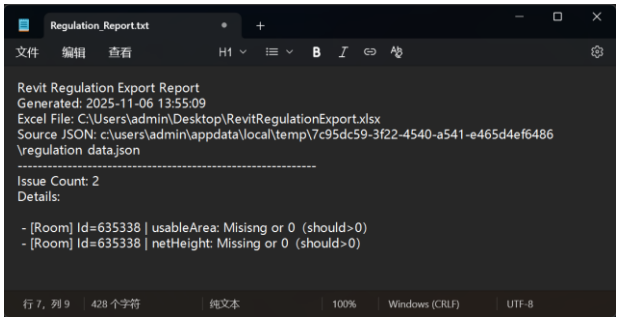

Figure 6: The generated result of regulation report text

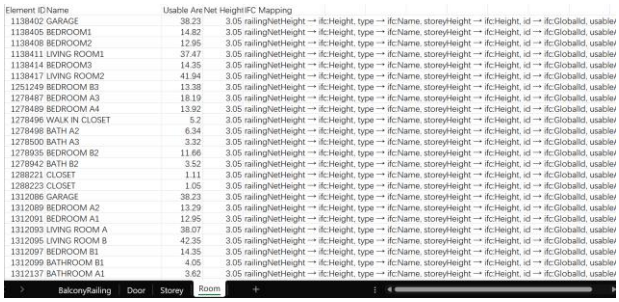

Figure 7: Example figure layout

Beyond textual validation, the framework produced two additional outputs that enhance its practical utility. First, it automatically generated a structured Excel file (Fig 7.), organising all model elements and their compliance status for item-by-item inspection by designers or reviewers. This tabular format enables traceability, documentation, and manual double-checking of design changes over time. Then the system delivered graphical feedback within Revit. Non-compliant elements were automatically selected and highlighted in the model view, enabling designers to visually identify and address each issue directly. If no violations were detected, a pop-up message appeared within Revit, indicating full compliance with the current rule set. Those are shown in Fig 8. This immediate interaction between reasoning output and model visualization underscores the system's lightweight, user-friendly nature.

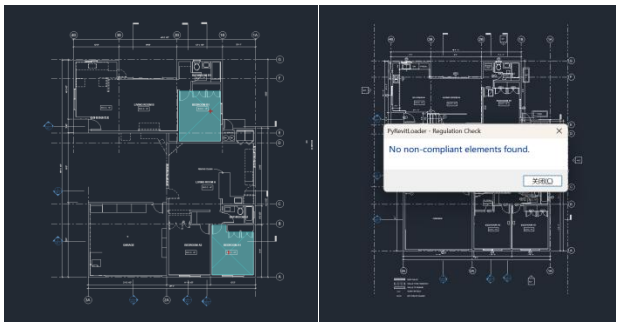

Figure 8: The graphical result shown inside Revit

Conclusions

This study presented a lightweight regulation compliance framework integrated within Autodesk Revit through a custom pyRevit extension. By linking BIM data with semantic web technologies with ifcOWL and SHACL, the system enables architects to perform specific regulation checks intuitively without leaving their design environment. The workflow automates data extraction, RDF conversion, and rule validation while providing both tabular and visual feedback, significantly improving accessibility for non-programming users.

Beyond its practical value, the study offers a theoretical implication by clarifying a reference architecture for "in-model semantic compliance": regulatory knowledge is treated as an explicit, modular layer which includes ontology and constraints that is conceptually separable from any particular BIM platform, while user interaction and feedback remain embedded in the authoring environment. This framing contributes to the Human-Data Interaction literature by positioning compliance checking as a human–data interaction problem where transparency, traceability, and actionable feedback are as critical as rule correctness.

A key limitation is that, at the current stage, a single prototype implementation was developed and validated in Revit; cross-platform interoperability across multiple BIM authoring tools will be empirically evaluated as primary goal in next step of the study. The current implementation is also restricted to quantitative and spatial clauses and still relies on manual encoding of regulatory knowledge. Future work will extend the rule set and test the workflow across additional BIM environments, as well as building on recent advances in AI-assisted code translation, natural language processing and large language models. Such integration of AI-driven knowledge modelling and semantic reasoning could significantly accelerate the creation of digital regulatory databases, improving adaptability, scalability, and adoption of compliance automation in the AEC industry.

Acknowledgments

The authors gratefully acknowledge BIM Engineer Hanqin TANG (WRNS Studio) for generously providing the Revit model used in the case study which enabled the practical validation of the proposed workflow within a real design project context.

References

Amor, R., & Dimyadi, J. (2021). The promise of automated compliance checking. Developments in the Built Environment, 5, 100039. https://doi.org/10.1016/j.dibe.2020.100039

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Bergés, M., Culler, D., Gupta, R. K., Kjærgaard, M. B., Srivastava, M., & Whitehouse, K. (2018). Brick: Metadata schema for portable smart building applications. Applied Energy, 226, 1273–1292. https://doi.org/10.1016/j.apenergy.2018.02.091

Bloch, T., Katz, M., Yosef, R., & Sacks, R. (2019). Automated model checking for topologically complex code requirements – security room case study. In Proceedings of the 2019 European Conference on Computing in Construction (EC3) (pp. 48–55). https://doi.org/10.35490/EC3.2019.157

Bloch, T., Wu, J., Alnuzha, M., & Borrmann, A. (2024). Leveraging graph based semantic enrichment for enhanced automated code-checking. In Proceedings of the 2024 European Conference on Computing in Construction (EC3).

Boje, C., Guerriero, A., Kubicki, S., & Rezgui, Y. (2020). Towards a semantic Construction Digital Twin: Directions for future research. Automation in Construction, 114. Scopus. https://doi.org/10.1016/j.autcon.2020.103179

Brick Consortium. Brick: A uniform metadata schema for buildings. Brick Ontology. Retrieved January 12, 2026, from https://brickschema.org/

Cerovšek, T., & Omar, M. (2025). Advancing Semantic Enrichment Compliance in BIM: An Ontology-Based Framework and IDS Evaluation. Buildings, 15(15), 2621. https://doi.org/10.3390/buildings15152621

Chen, N., Lin, X., Jiang, H., & An, Y. (2024). Automated Building Information Modeling Compliance Check through a Large Language Model Combined with Deep Learning and Ontology. Buildings, 14(7), 1983. https://doi.org/10.3390/buildings14071983

Dimyadi, J., & Amor, R. (2013). Automated building code compliance checking – Where is it at? In Proceedings of CIB World Building Congress 2013: Construction and Society (pp. 172–185). Brisbane, Australia

Donkers, A. (2022). Ontology Design Template. https://doi.org/10.5281/zenodo.681689

Donkers, A., Yang, D., Vries, de V., & Baken, N. (2023). A Visual Support Tool for Decision-Making over Federated Building Information. In C. Andriotis, M. Turrin, & A. Rafiee (Eds), Computer-Aided Architectural Design. INTERCONNECTIONS: Co-computing Beyond Boundaries: 20th International Conference, CAAD Futures 2023, Delft, The Netherlands, July 5–7, 2023, Selected Papers (Vol. 1819, p. 485~500). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-37189-9

Doukari, O., Greenwood, D., Rogage, K., & Kassem, M. (2021). Object-oriented compliance checking: An automated approach and a case study. In Proceedings of the 2021 European Conference on Computing in Construction (EC3) (pp. 293–302). https://doi.org/10.35490/EC3.2021.152

Eastman, C., Lee, J., Jeong, Y., & Lee, J. (2009). Automatic rule-based checking of building designs. Automation in Construction, 18(8), 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Fuchs, S., Fauth, J., Boden, M., & Amor, R. (2025). The challenge of automated compliance checking: A regulatory view. In Proceedings of the 2025 European Conference on Computing in Construction (EC3 2025) & CIB W78 Conference on IT in Construction (Porto, Portugal, July 14–17, 2025).

Heimsbakk, V., & Torkelsen, K. (2023). Using the Shapes Constraint Language for modelling regulatory requirements (No. arXiv:2309.02723). arXiv. https://doi.org/10.48550/arXiv.2309.02723

Mäki, T., & Kerosuo, H. (2015). Site managers' daily work and the uses of building information modelling in construction site management. Construction Management and Economics, 33(3), 163–175. https://doi.org/10.1080/01446193.2015.1028953

Mhmoud Alzubi, K., Salah Alaloul, W., Malkawi, A. B., Al Salaheen, M., Hannan Qureshi, A., & Ali Musarat, M. (2023). Automated monitoring technologies and construction productivity enhancement: Building projects case. Ain Shams Engineering Journal, 14(8), 102042. https://doi.org/10.1016/j.asej.2022.102042

Michaud, M., Forgues, E.-C., Carignan, V., Forgues, D., & Ouellet-Plamondon, C. (2019). A lean approach to optimize BIM information flow using value stream mapping. Journal of Information Technology in Construction (ITcon), 24, 472–488.

Pauwels, P., & Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Automation in Construction, 63, 100–133. https://doi.org/10.1016/j.autcon.2015.12.003

Preidel C, Borrmann A (2016). Towards code compliance checking on the basis of a visual programming language, ITcon Vol. 21, Special issue CIB W78 2015 Special track on Compliance Checking, pg. 402-421, https://www.itcon.org/2016/25

World Wide Web Consortium. (2017, July 20). Shapes Constraint Language (SHACL) (W3C Recommendation). https://www.w3.org/TR/shacl/

Zhang, R., & El-Gohary, N. (2019, June). A machine learning-based method for building code requirement hierarchy extraction. 2019 Canadian Society for Civil Engineering Annual Conference, Laval, Canada.

https://legacy.csce.ca/elf/apps/CONFERENCEVIEWER/conferences/2019/pdfs/PaperPDFversion_147_0423081134.pdf

Zheng, Z., Zhou, Y.-C., Lu, X.-Z., & Lin, J.-R. (2022). Knowledge-informed semantic alignment and rule interpretation for automated compliance checking. Automation in Construction, 142, 104524. https://doi.org/10.1016/j.autcon.2022.104524

Zhong, B., Gan, C., Luo, H., & Xing, X. (2018). Ontology-based framework for building environmental monitoring and compliance checking under BIM environment. Building and Environment, 141, 127–142. https://doi.org/10.1016/j.buildenv.2018.05.046

Zhong, B., Wu, H., Li, H., Sepasgozar, S., Luo, H., & He, L. (2019). A scientometric analysis and critical review of construction related ontology research. Automation in Construction, 101, 17–31. https://doi.org/10.1016/j.autcon.2018.12.013