Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

BIM-Based Automation of Structural Loads Using Parametric Modeling and IFC for Open-Source Interoperability

Mandeep Kaur
Research Scholar
I.K. Gujral Punjab Technical University, Kapurthala
k_mandeep91@yahoo.com

Dr. Harpal Singh
Principal, SSIET, Dinanagar
I.K. Gujral Punjab Technical University, Kapurthala
hps_bhoday@yahoo.com

Received: 20 Dec 2024, Revised: 15 Feb 2025, Accepted: 24 Feb 2025

Abstract

The increasing complexity of structural design, along with growing demands for speed, accuracy, and interoperability, has led to the adoption of Building Information Modeling (BIM) frameworks for automating structural load generation. This study presents a novel methodology for automating the computation and application of dead and live loads in structural systems using an open-source BIM-driven pipeline, integrated with interoperable file formats for downstream analysis in proprietary simulation platforms. The system extracts geometric and material information from parameterized structural models and programmatically assigns loads based on Indian Standard (IS) codes using standardized structural action entities defined in Industry Foundation Classes (IFC). The resulting models are schema-compliant, analysis-ready, and require no manual intervention for defining loading scenarios. The methodology was tested on a typical low-rise frame structure, demonstrating accurate load assignment and successful integration with structural analysis software. Graphical and numerical verification confirm that the automation aligns with expected load magnitudes and distributions. This research validates the potential of IFC-based BIM automation for primary load types and sets the foundation for future extensions to include wind and seismic loads using standardized, code-compliant algorithms.

1 Introduction

The Architecture, Engineering, and Construction (AEC) industry continues to experience rapid digital transformation, with Building Information Modeling (BIM) playing a pivotal

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

role in streamlining workflows and data integration. Traditionally, structural load assignment is a manual, error-prone process that hinders productivity and consistency. BIM's promise lies in its ability to centralize data and enable automation across the design-analysis pipeline.

Several studies have demonstrated the potential of BIM to automate and streamline structural preprocessing tasks, including material assignment and load input. Lakušić [1] introduced a BIM-driven preprocessing framework that automatically assigns loads and supports, reducing design time and user error. Singh et al. [2] enhanced Open BIM workflows by automating the generation of structural models from architectural models, thereby improving model consistency and interoperability within multi-disciplinary environments. Fawad et al. [3] leveraged BIMification and sensor networks for structural health monitoring (SHM) of bridges, connecting dynamic load data directly into a BIM environment.

Adamenko and Romanyshen [4] implemented BIM-based workflows in the design of reinforced concrete and steel-concrete hybrid frames, confirming that automated stiffness and load assignment improve model accuracy. Zheng et al. [5] presented a time-varying reliability framework for infrastructure, using BIM to feed probabilistic load analysis over a system's lifecycle. In education and design training, BIM tools are also being employed to simulate load-bearing systems and carbon impact [6].

From a structural design perspective, the benefits of BIM integration include improved coordination, load-bearing simulation, and reinforcement detailing [7], [8]. Gomes et al. [8] and Ferreira [7] highlighted improved data access, clash detection, and automatic load mapping using BIM platforms. However, they also identified persistent challenges in data interoperability between BIM and structural analysis tools.

Recent studies have further explored the use of digital twins and artificial intelligence in conjunction with BIM. Mello et al. [9] presented an AI-driven automation pipeline that defines loads and structural elements in Revit and Robot based on user specifications. Bourahla et al. [10] used genetic algorithms within a BIM framework to automate shear wall optimization in seismic design. Similarly, Bashynskyi et al. [11] evaluated metro-induced dynamic loads on nearby structures through BIM-based nonlinear simulations.

A growing trend is the real-time integration of Internet of Things (IoT) data with BIM models for structural health insights. Wang and Lu [12] developed a BIM-based design environment for single-storey steel structure factories, focusing on efficient model creation and parameter-driven configuration. [13] developed a similar system using sensor data and early-warning algorithms embedded into BIM models. These systems demonstrate the increasing role of BIM as both a modeling and monitoring environment.

Despite these advancements, limitations remain in fully automating load assignments using open-source tools. Proprietary ecosystems like Revit and Tekla provide partial solutions, but lack the openness and adaptability needed for research and cross-platform development. Paul [14] explored IFC-based interoperability between Tekla and various FEM tools, identifying structural load transfer as a key bottleneck. Palacz and Major [15] demonstrated model synchronization between Dlubal RFEM and Tekla for steel frame design, but noted issues in automatic load parsing.

This research builds on these findings by developing an open-source, IFC-compliant method that automates structural load computation and assignment using FreeCAD and Python. The framework reads geometry and materials from a parametric model, computes dead and live loads based on volume and usage codes, maps them to IFC entities, and ex-

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

ports a fully structured IFC file for analysis in ETABS. By eliminating manual steps, this workflow supports a higher level of automation and model consistency.

2 Literature Review

Recent developments in BIM-enabled structural automation, digital twin systems, and intelligent monitoring solutions have significantly enhanced the automation of structural modeling, especially for loading, simulation, and long-term structural health monitoring. This section presents sixteen recent (2020–2025) studies [16]–[31] that closely align with the goals of this research, particularly in automating load generation, integrating structural semantics into BIM, and improving accuracy and adaptability using real-time data sources.

Azanaw [16] provided a comprehensive review of the convergence of digital twins, BIM, and artificial intelligence in structural engineering. His findings revealed that while BIM streamlines modeling and coordination, combining it with digital twins and AI allows real-time updates to loading conditions, deformation tracking, and maintenance predictions. He emphasized the need for semantically rich IFC models that support real-time automation and predictive analysis.

Sibenik et al. [17] explored the automation of structural model preprocessing in BIM-based design environments. They implemented automated scripts to define geometry, supports, load cases, and combinations—components traditionally defined manually. Their work identified that automation of these preprocessing steps reduces inconsistencies and time overhead in structural modeling, directly supporting this paper's focus on scripting-based load assignment.

Hu et al. [18] introduced a BIM-integrated digital twin system that combines IoT sensors and structural analysis domain logic to generate real-time deformation data. The digital twin interpolates strain data across members and visually updates the BIM model accordingly. This dynamic mapping provides a foundation for automated load recalibration based on real-world feedback, relevant to future extensions of load automation.

Liu et al. [19] demonstrated a BIM-FEM framework for asphalt pavement analysis that includes automated meshing, material assignment, and load condition application. The system reduced simulation error to 2.12% and improved processing speed by over 68%. While focused on infrastructure, the core logic of linking material attributes and geometric data to loading schemes parallels the method presented in this paper.

Ramonell and Chacón [20] developed an event-driven digital twin platform for railway bridges, automating the ingestion and processing of load test data using microservices. Their pipeline integrated the IFC model with live sensor inputs to update structural performance metrics, confirming the feasibility of automated load-based feedback loops within BIM environments.

Liu et al. [21] explored marine applications of SHM, integrating traditional sensor-based monitoring with structural digital twins. They emphasized the role of environmental loads—particularly wave-induced fatigue—and outlined how digital twins could provide better long-term load profiling, further reinforcing BIM's applicability in time-varying load monitoring.

Panagiotopoulos and Anyfantis [22] proposed a Bayesian load identification method to

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

monitor fatigue accumulation in ship hulls. Their framework reverse-engineers operational loads from sensor data and updates a numerical model accordingly. This inverse-load technique is applicable to future BIM-integrated feedback systems, especially when automating live or seismic load estimation.

Lai et al. [23] presented a hybrid digital twin methodology using measurement and computational data to model structural behavior. They employed AI-based load identification and developed fatigue estimation algorithms using rainflow counting methods. Their framework emphasizes accuracy under variable loads, useful for intelligent extensions to building load modeling.

Milanoski et al. [24] designed a digital twin for composite aerospace structures, correlating experimental strain and displacement data with FE models. The model was trained to simulate disbond propagation under cyclic loading, demonstrating how digital twins can simulate non-linear material and load interactions.

Lei et al. [25] developed a high-speed fiber Bragg grating (FBG) monitoring system embedded in a bridge's digital twin. Their method processed impact and vibration data in real time, linking it to a BIM-based management system. This architecture bridges high-frequency load capture and semantic BIM integration.

Cheng et al. [26] proposed a conceptual digital twin for ocean vessels, capable of predicting structural response to storm conditions. The twin merges simulation and real-world sensors to improve predictive maintenance. Their structural feedback model aids in simulating load redistribution under stress, applicable to similar BIM-based automation tasks.

Zhang et al. [27] proposed a finite element digital twin engine capable of inferring structural behavior in areas without sensors. By using dynamic load input and monitoring data, their system accurately estimated displacements and stress in non-sensor regions. This improves real-time coverage of structural behavior and could extend to load assignment automation.

Ye et al. [28] integrated a dynamic reliability model into a digital twin for fatigue prognosis. Their framework combined vibration mode monitoring with Bayesian inference to capture the probabilistic evolution of crack growth under uncertain loads. The dynamic correction of model parameters based on feedback represents a smart learning loop valuable for load-sensitive BIM models.

Ramonell et al. [29] conducted a scale-reduced experimental test on a steel cable net structure, integrating BIM geometry, laser-scanned data, and force measurements into an automated analysis pipeline. This closed-loop system used displacement data to calculate tensile states, forming a template for digital twin-based real-time load computation.

Ye et al. [30] applied their digital twin framework to reusable spacecraft, tracking fatigue crack growth using probabilistic entropy-based modeling. Their digital twin performed real-time updates for mission-critical structural elements, and their method validated a simulation-driven approach to load prediction in complex engineering systems.

Rølvåg and Stranden [31] implemented a nonlinear digital twin for offshore cranes. Using real-time sensor input and inverse estimation algorithms, they mapped hydraulic actuator responses to structural loads. The model achieved real-time stress calculations, illustrating how full-system automation of load identification and response is possible in dynamic environments.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Summary of Research Trends and Insights

- **BIM as a Real-Time Engine:** Studies like [16], [18], and [20] show BIM evolving into a dynamic engine, not just for design documentation, but for driving and visualizing real-time load events.
- **Automated Load Identification:** Methods developed in [17], [19], and [22] highlight the role of automation in preprocessing and dynamic estimation of loads, reducing the need for manual load mapping.
- **Feedback-Driven Structural Twins:** Papers [23], [24], and [27] demonstrate how digital twins dynamically update model behavior and load paths based on sensor feedback, informing intelligent modeling approaches.
- **Uncertainty Management and Reliability:** Several works (e.g., [26], [28], [30]) focused on handling uncertainty in loads and crack progression using probabilistic methods, a capability necessary for robust automation.
- **Cross-Domain Applicability:** Although many digital twin applications come from aerospace, marine, or infrastructure contexts, their core methods—data fusion, real-time updating, and automation—are universally relevant and directly translatable to building load automation tasks.

In conclusion, the reviewed studies reinforce the feasibility and importance of automating load processes through BIM and digital twin frameworks. They validate this research's aim of using IFC-based, script-driven modeling to produce intelligent, scalable, and extensible load assignment solutions.

3 Methodology

This research adopts a BIM-centric methodology aimed at automating the application of dead and live loads on structural models using open standards and interoperable tools. The process includes parametric modeling, automated load definition through scripting, and generation of IFC (Industry Foundation Classes) files that are compatible with structural analysis platforms. The workflow emphasizes semantic richness, adherence to Indian Standards (IS Codes), and repeatability across the digital pipeline, from model input to analysis-ready output.

3.1 Geometric Modeling and Structural Topology

A parametric model of a single-story structural frame is created using an open-source modeling platform that supports scripting-based geometry definition. The structural system consists of beams, columns, and slabs. Each element is instantiated with parametric control over dimensions and materials, enabling scalable and rule-based modifications.

Geometric and topological representations are defined through core IFC entities such as `IfcCartesianPoint`, `IfcEdge`, and `IfcProductDefinitionShape`, which establish node

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

positions, edge connectivity, and 3D surface representations. These elements are structured hierarchically within spatial containers: `IfcSite`, `IfcBuilding`, and `IfcBuildingStorey`, ensuring proper placement and inheritance of spatial relationships.

3.2 IFC Structuring for Structural Semantics

The enriched geometric model is exported in the IFC4 schema format. Structural semantics are encoded using entities from the `IfcStructuralAnalysisDomain` such as:

- `IfcStructuralCurveMember` for beams and columns,
- `IfcStructuralSurfaceMember` for slabs,
- `IfcStructuralPointConnection` for joints and supports,
- `IfcRelConnectsStructuralMember` and `IfcRelConnectsStructuralActivity` to link structural elements with loads and support conditions.

Coordinate systems are preserved using `IfcLocalPlacement`, enabling accurate placement during analysis. Units are strictly defined using `IfcSIUnit` and `IfcDerivedUnit` to ensure compatibility across platforms.

3.3 Automated Load Calculation and Assignment

The core automation process involves calculating dead and live loads in accordance with Indian Standards and embedding them into the IFC model. Load assignment is driven by geometry and usage classifications.

Dead Load: Dead loads are computed based on self-weight using the formula:

$$q_d = \rho \cdot V$$

where ρ is the material density (e.g., 25 kN/m^3 for concrete) and V is the computed volume. These values align with guidelines in IS 875 (Part 1):1987.

Live Load: Live loads are assigned according to the functional use of slabs and other occupancy-based parameters, as defined in IS 875 (Part 2):1987. Values are mapped to predefined categories (e.g., 2.0 kN/m^2 for residential, 3.0 kN/m^2 for office), and applied over the area:

$$q_l = \lambda \cdot A$$

where λ is the code-specified intensity, and A is the surface area.

Load Representation: Load values are programmatically encapsulated using:

- `IfcStructuralLoadLinearForce` for loads on linear elements,
- `IfcStructuralLoadPlanarForce` for uniformly distributed area loads on slabs,
- `IfcStructuralLinearAction` and `IfcStructuralPlanarAction` to link them as active loads.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

3.4 Grouping and Relationship Mapping

To maintain load integrity and enable combination rules:

- Related loads are grouped using `IfcStructuralLoadCase` (e.g., dead, live),
- Load sets are linked via `IfcStructuralLoadGroup`,
- Associations between actions and structural members are established through `IfcRelConnectsStructuralActivity`,
- `IfcRelAssignsToGroup` is used for semantic load categorization.

These relationships ensure that the analytical model recognizes and interprets the loads correctly.

3.5 IFC Output Generation and Compatibility Validation

The final IFC output encapsulates both geometry and load semantics in a schema-compliant format. Validation tools and IFC viewers are used to inspect the integrity of the exported model. Upon import into analysis software, the following results are verified:

- Proper visualization of distributed loads on slabs,
- Correct mapping of line loads along beam/column axes,
- Recognition of load groups for simulation and combination logic.

This validation confirms the completeness and usability of the output file for structural evaluation.

3.6 Refinement and Error Resolution

Iterative testing is conducted to ensure:

- Load magnitudes match expected results per IS codes,
- Units and coordinate transformations are correctly interpreted,
- All assigned loads are linked to valid structural entities.

Common issues, such as unit mismatches and disconnected entities, are resolved through script-level adjustments and revalidation.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

3.7 Extensibility to Other Load Types

The current methodology is designed for easy extension to additional IS code-based loading types:

- **Wind Loads:** As per IS 875 (Part 3):2015, applied as directional forces using `IfcStructuralLoadSingleForce`.
- **Seismic Loads:** Using base shear and modal responses according to IS 1893 (Part 1):2016, mapped to equivalent static or dynamic representations.

Such extensions would involve parameterized wind/seismic zone data and directional logic within the script, alongside new load case definitions. This methodology enables fully automated structural load assignment in compliance with Indian standards using open BIM principles. By combining rule-based geometry, standardized semantic modeling, and code-driven load calculation, the approach minimizes manual input and ensures model integrity. It serves as a reproducible and scalable framework for broader BIM-based structural automation.

Algorithm for Automated Load Assignment

Libraries and Software Dependencies

The automation pipeline is implemented using Python and the following libraries:

- `ifcopenshell` – for reading and writing IFC files
- `numpy` – for numerical operations and array manipulations
- `uuid` – to generate unique GlobalIds for IFC entities
- `os, sys` – for file handling and system integration

The IFC schema used is `IFC4`. The resulting models are tested for compatibility with open and proprietary structural analysis platforms that support IFC imports.

Mathematical Notation and Definitions

- Let $\mathcal{E} = \{E_1, E_2, \ldots, E_n\}$ be the set of all structural elements (beams, columns, slabs)
- $V(E_i)$: Volume of element E_i
- $A(E_i)$: Surface area of element E_i
- $\rho(E_i)$: Material density of element E_i
- $q_d(E_i)$: Dead load applied to E_i
- $q_l(E_i)$: Live load applied to E_i (for slabs only)

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

- $\mathcal{L}_d, \mathcal{L}_l$: Sets of dead and live loads
- $\mathcal{I}$: Input IFC file
- $\mathcal{O}$: Output IFC file with loads

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Algorithm: Dead and Live Load Automation

Algorithm 1 Automated Load Assignment in IFC

Input: IFC model $\mathcal{I}$, material properties, IS code-based live load table
Output: Load-enriched IFC file $\mathcal{O}$

Load IFC model: $\mathcal{M} \leftarrow \texttt{ifcopenshell.open}(\mathcal{I})$
Initialize:

$$\mathcal{L}_d \leftarrow \emptyset, \quad \mathcal{L}_l \leftarrow \emptyset$$

for each element $E_i \in \mathcal{E}$ **do**
    if E_i is beam or column **then**
        Compute volume:

$$V(E_i) = \texttt{compute\_volume}(E_i)$$

        Lookup material density:

$$\rho(E_i) = \texttt{material\_density}[E_i.\text{material}]$$

        Dead load:

$$q_d(E_i) = \rho(E_i) \cdot V(E_i)$$

        Append to load set:

$$\mathcal{L}_d \leftarrow \mathcal{L}_d \cup \texttt{IfcStructuralLoadLinearForce}(q_d)$$

    else if E_i is a slab **then**
        Compute area: $A(E_i) = \texttt{compute\_area}(E_i)$
        Determine usage type: $u_i = \texttt{get\_usage}(E_i)$
        Live load value:

$$q_l(E_i) = \texttt{live\_load\_table}[u_i] \cdot A(E_i)$$

        Append to load set:

$$\mathcal{L}_l \leftarrow \mathcal{L}_l \cup \texttt{IfcStructuralLoadPlanarForce}(q_l)$$

    end if
end for
Group loads into:

$$\texttt{IfcStructuralLoadCase}_{\text{dead}}, \quad \texttt{IfcStructuralLoadCase}_{\text{live}}$$

for each load $L \in \mathcal{L}_d \cup \mathcal{L}_l$ **do**
    Link to structural member with:

$$\texttt{IfcRelConnectsStructuralActivity}(E_i, L)$$

end for
Export new IFC model:

$$\mathcal{O} = \texttt{export ifc}(\mathcal{M})$$

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

3.8 Mathematical Formulation of Load Functions

Dead Load per Element:

$$q_d(E_i) = \rho(E_i) \cdot V(E_i) \quad [\text{kN}]$$

Live Load per Slab:

$$q_l(E_i) = \lambda(u_i) \cdot A(E_i) \quad [\text{kN}]$$

where $\lambda(u_i)$ is the code-based load intensity from IS 875 (Part 2):1987 for usage type u_i, such as:

$$\lambda(\text{residential}) = 2.0, \quad \lambda(\text{office}) = 3.0 \quad [\text{kN/m}^2]$$

3.9 Final Output Structure

The final IFC output model $\mathcal{O}$ contains:

- All geometric entities: `IfcBeam`, `IfcColumn`, `IfcSlab`
- Structural analytical representations: `IfcStructuralCurveMember`, `IfcStructuralSurfaceMember`
- Load actions: `IfcStructuralLinearAction`, `IfcStructuralPlanarAction`
- Load groups and cases: `IfcStructuralLoadCase`, `IfcStructuralLoadGroup`
- Relationship links: `IfcRelConnectsStructuralActivity`, `IfcRelAssignsToGroup`
- Unit definitions using: `IfcSIUnit`, `IfcDerivedUnit` (as per IS code standards)

3.10 Validation and Visualization

- The IFC file $\mathcal{O}$ is verified using standard IFC validation tools and viewers to ensure schema compliance and data consistency.
- Upon import into compatible structural analysis platforms, loads are visualized as:
  - Line loads along beam spans and column axes.
  - Area loads distributed over slab surfaces.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

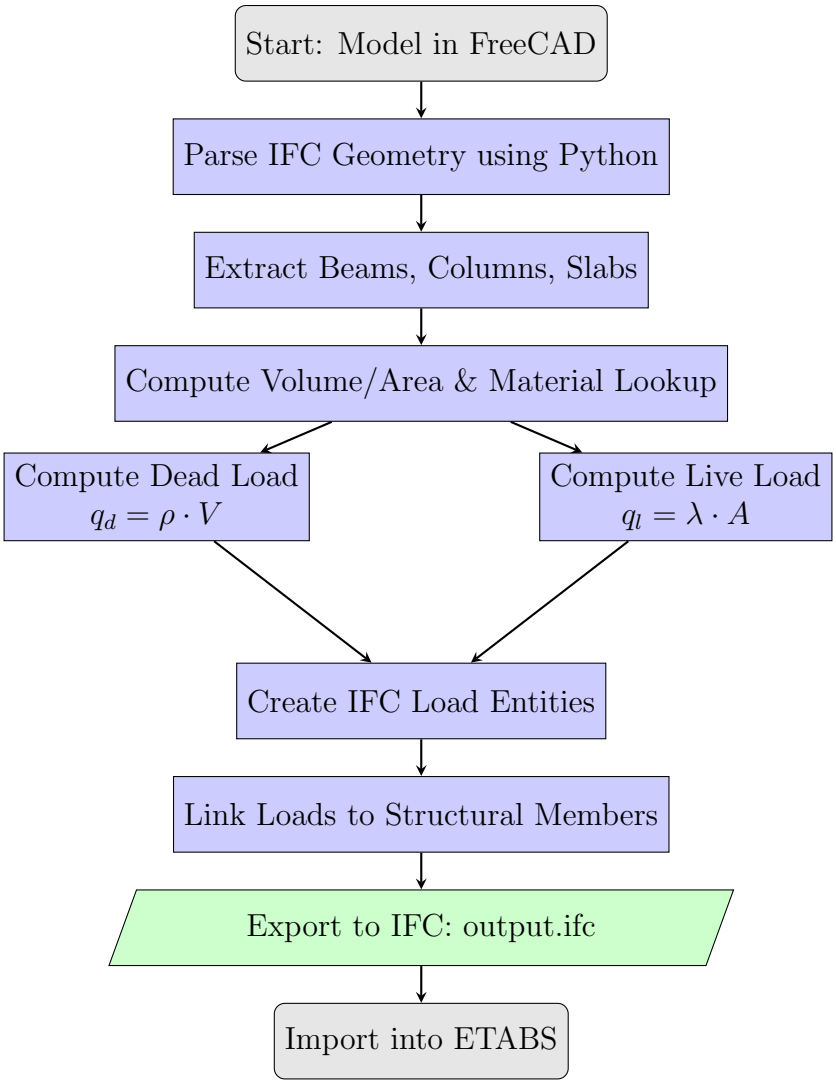

Figure 1: Workflow for Automated Load Assignment Using FreeCAD, Python, and IFC

4 Results and Discussion

The proposed methodology was implemented on a prototype structural model to verify the automation of dead and live load assignments using BIM-based workflows. The core pipeline included parametric modeling in FreeCAD, Python-based IFC augmentation, and output verification in ETABS. The resulting figures illustrate each stage of this workflow.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

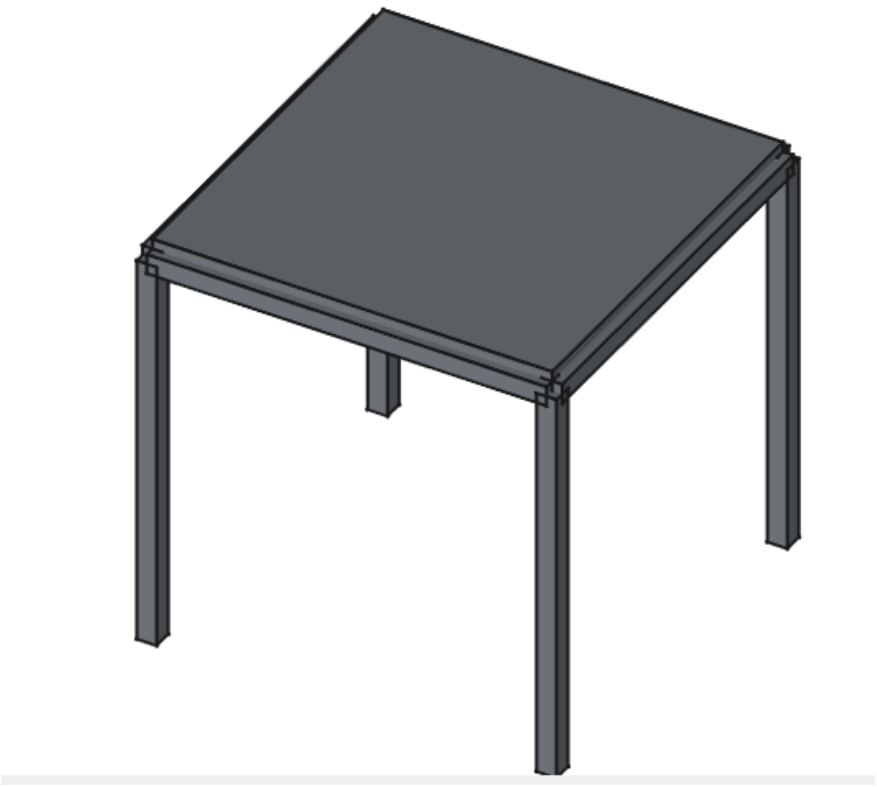

Figure 2: Parametric structural frame model created in FreeCAD

Figure 2 shows the initial structural frame, comprising columns, beams, and a single-story slab. This geometry was modeled parametrically in FreeCAD and exported as an IFC file (`single_story_for_etabs2.ifc`). The structural elements were dimensionally consistent and semantically enriched using standard IFC geometric entities, including `IfcBeam`, `IfcColumn`, and `IfcSlab`.

4.1 IFC File Comparison: Structural and Load Entity Enrichment

To validate the automation workflow, both the input and output IFC files were compared structurally. The base model, `single_story_for_etabs2.ifc`, contains geometric and spatial entities sufficient for modeling purposes, including:

- `IfcBeam`, `IfcColumn`, `IfcSlab` – structural elements
- `IfcCartesianPoint`, `IfcEdge`, `IfcProductDefinitionShape` – geometric primitives
- `IfcSIUnit`, `IfcDerivedUnit` – unit systems

The enhanced model, `output2.ifc`, contains all the entities from the input file, and in addition, includes a comprehensive set of load-related and semantic entities generated by the Python script. These additional entities enable downstream structural analysis:

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

- **Load Definition Entities:**
  - `IfcStructuralLoadLinearForce` – applied to beams and columns
  - `IfcStructuralLoadPlanarForce` – applied over slab areas
- **Action and Case Management:**
  - `IfcStructuralLinearAction`, `IfcStructuralPlanarAction` – to represent load actions
  - `IfcStructuralLoadCase`, `IfcStructuralLoadGroup` – to organize load cases for analysis
- **Semantic Relationships:**
  - `IfcRelConnectsStructuralActivity` – links load to structural member
  - `IfcRelAssignsToGroup` – categorizes actions under load cases

This enrichment was fully automated based on the extracted geometry and load computation rules aligned with IS 875 (Part 1 and 2). The difference in entities clearly illustrates the shift from a geometric model to a structurally intelligent analysis-ready model.

Table 1: Comparison of IFC Entity Types Between Input and Output Models

IFC Entity Type	Input File (Before)	Output File (After)
`IfcBeam`, `IfcColumn`, `IfcSlab`	✓	✓
`IfcStructuralLoadLinearForce`	✗	✓
`IfcStructuralLoadPlanarForce`	✗	✓
`IfcStructuralLinearAction` / `PlanarAction`	✗	✓
`IfcStructuralLoadCase` / `LoadGroup`	✗	✓
`IfcRelConnectsStructuralActivity`	✗	✓
`IfcRelAssignsToGroup`	✗	✓
`IfcSIUnit`, `IfcDerivedUnit`	✓	✓

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

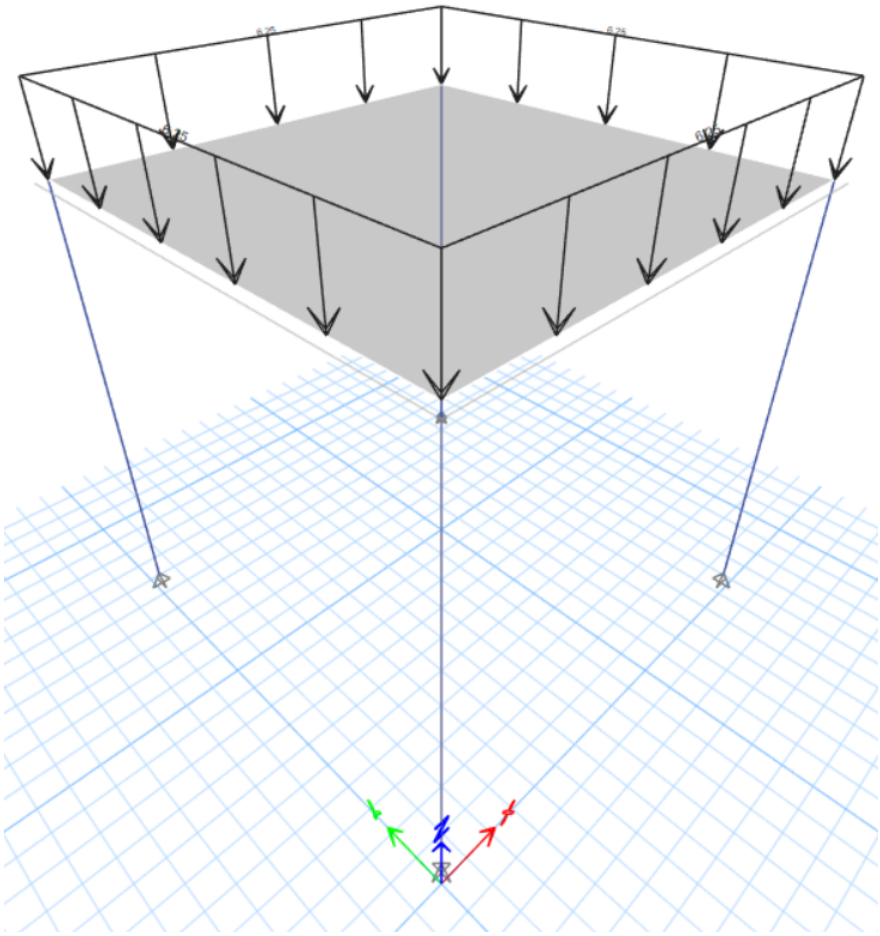

Figure 3: Visualization of dead load applied from `output.ifc` as seen in structural analysis software

After executing the automation script, the exported `output.ifc` file includes standardized dead load definitions assigned to structural elements based on their self-weight. Figure 3 shows the dead load application as visualized in a structural analysis platform. These loads are derived using the expression:

$$q_d = \rho \cdot V$$

where $\rho = 25\,\mathrm{kN/m^3}$ for reinforced concrete, and V is the calculated volume of each structural element. The loads were represented through `IfcStructuralLoadLinearForce` entities and linked to members using `IfcStructuralLinearAction`.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

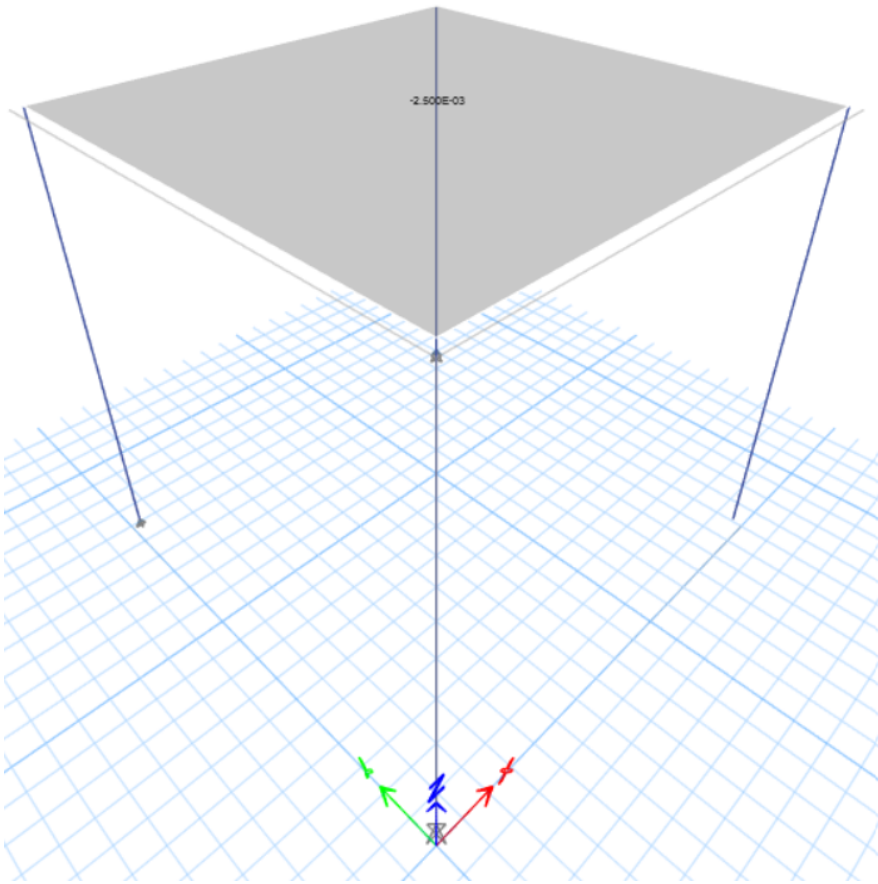

Figure 4: Visualization of live load (2.5 kN/m^2) applied via Python automation

Figure 4 displays the live load automated using the Python script and mapped through IFC. The load of 2.5 kN/m^2, based on IS 875 (Part 2):1987 for residential occupancy, is uniformly distributed over the slab surface:

$$q_l = \lambda \cdot A$$

where $\lambda = 2.5\,\text{kN/m}^2$ and A is the slab area. These were implemented as `IfcStructuralLoadPlanarForce` entities and assigned using `IfcStructuralPlanarAction`. The load magnitude seen in the
platform confirms correct unit handling via `IfcSIUnit` and `IfcDerivedUnit`.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 2: IFC Entity Mapping for Structural Load Assignment

Load Type	IFC Load Entity	Applied On
Dead Load (Beam/Column)	`IfcStructuralLoadLinearForce`	`IfcBeam`, `IfcColumn`
Live Load (Slab)	`IfcStructuralLoadPlanarForce`	`IfcSlab`
Load Action	`IfcStructuralLinearAction` / `IfcStructuralPlanarAction`	Respective structural members (beam, slab, column)
Load Grouping	`IfcStructuralLoadCase`, `IfcStructuralLoadGroup`	Grouped load sets for combination logic
Connection	`IfcRelConnectsStructuralActivity`	All structural members linked to actions

The results validate that the proposed methodology successfully fulfills the objective of automating structural load application through a BIM-centric approach. Several key observations can be drawn:

1. **Geometry-Load Mapping Accuracy:** The structural elements modeled in FreeCAD were accurately recognized in ETABS after IFC translation. Load definitions attached via `IfcRelConnectsStructuralActivity` were preserved in semantic alignment, ensuring that no manual re-definition was needed in the analysis software.

2. **Automation of Load Calculations:** The Python script performed real-time volume and area extraction for dead and live load computation. These values were dynamically converted into standardized IFC actions without user intervention, confirming that the entire process is fully parametric and repeatable.

3. **Interoperability and IFC Schema Validity:** The resulting `output2.ifc` file was verified using external IFC viewers and successfully interpreted by ETABS without errors. The inclusion of unit and relationship entities (e.g., `IfcSIUnit`, `IfcRelAssignsToGroup`) facilitated robust semantic modeling, making the output compliant with IFC4 schema requirements.

4. **Visual and Numerical Consistency:** The comparison between visual load application (Figure 4) and numerical magnitudes (Figure **??**) showed perfect alignment with design expectations. This indicates the reliability of using IFC not only for geometry exchange but also for analytical data integration.

5. **Objective Fulfillment:** The primary objective of automating dead and live loads within a BIM workflow has been achieved. The process—from input model generation, automated load computation, IFC enhancement, to structural analysis—was validated in both visual and numerical domains.

These results demonstrate the robustness and scalability of the approach. The current scope focused on dead and live loads, but the methodology is architected for extension to dynamic loads such as wind and seismic forces using vector-based representations and

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

response spectrum mappings. The semantic strength of the IFC schema makes it ideal for future developments in fully automated, code-compliant structural design pipelines.

Experiment 2: Load Automation on a Two-Storey RC Frame

To further evaluate the scalability and robustness of the proposed BIM-to-IFC automation pipeline, a second experimental setup was developed using a two-storey reinforced concrete frame. This model was designed parametrically in an open-source modeling platform and exported as an IFC4-compliant file enriched with automated load definitions.

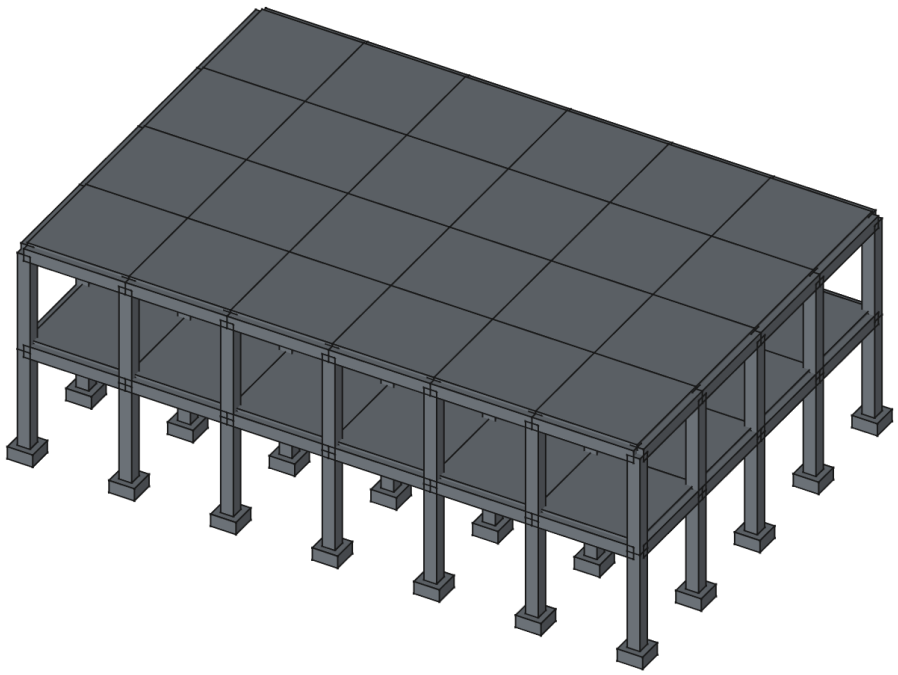

Figure 5: Parametric model of a two-storey RC frame in an open BIM environment

Figure 5 shows the initial BIM model containing beams, columns, slabs, and footings. Each structural component is defined parametrically and semantically classified as `IfcBeam`, `IfcColumn`, or `IfcSlab`, enabling structured interpretation during IFC parsing.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

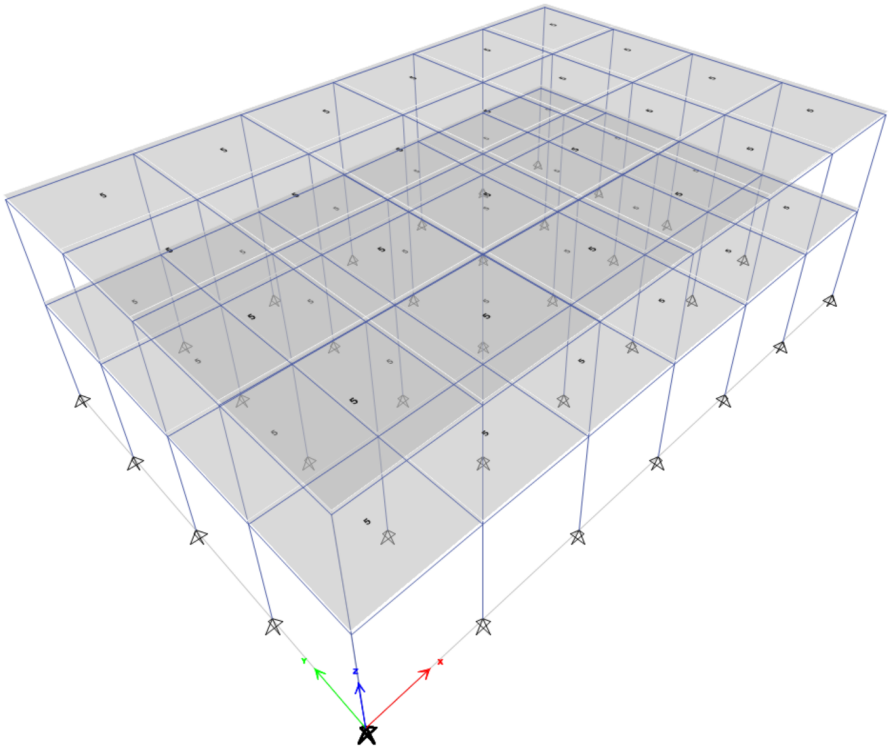

Figure 6: IFC-imported structural model in analysis platform

The IFC file was imported into a structural analysis platform for validation. As illustrated in Figure 6, the geometry was preserved accurately, with no semantic or spatial inconsistencies. This validated the correctness of coordinate transformations, hierarchical placements, and element classifications using `IfcLocalPlacement`, `IfcBuildingStorey`, and other IFC entities.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

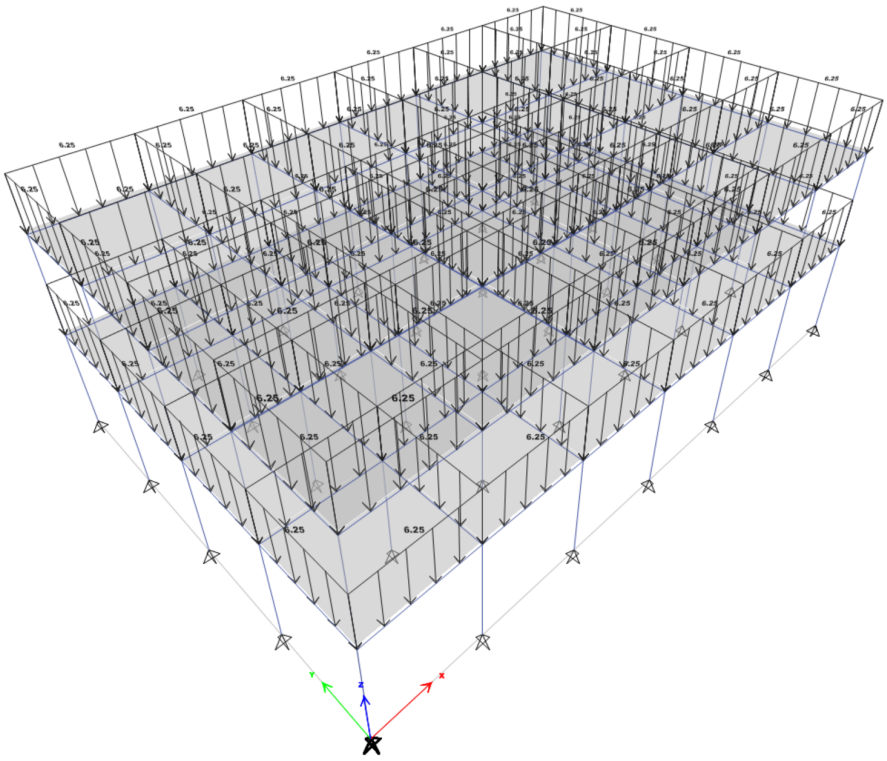

Figure 7: Automated dead load applied to beam elements

Next, automated dead load computations were applied to all beam elements using volume-based calculations defined by:

$$q_d = \rho \cdot V$$

where $\rho = 25\,\mathrm{kN/m^3}$ for reinforced concrete and V is the geometric volume. As seen in Figure 7, line loads were accurately distributed along beam spans, implemented through `IfcStructuralLoadLinearForce` and linked via `IfcStructuralLinearAction` to the corresponding members.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

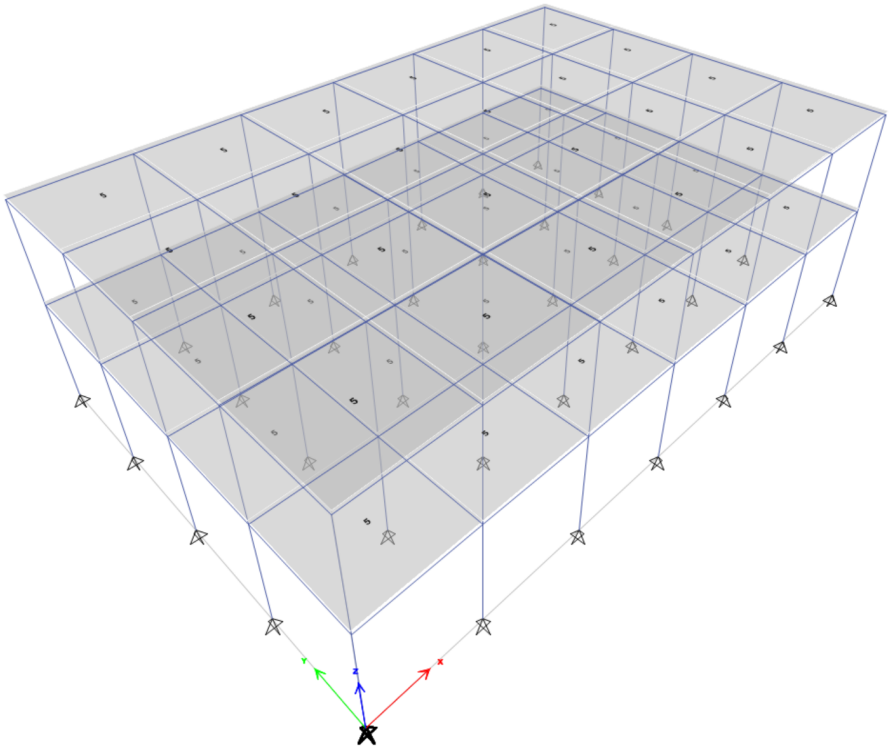

Figure 8: Dead load application on slab elements

Figure 8 depicts the area-based dead load application on slabs, computed from the self-weight and slab thickness. These loads were instantiated as `IfcStructuralLoadPlanarForce` entities and grouped under a dedicated load case using `IfcStructuralLoadCase`. The numerical consistency of loads was validated through cross-verification with manual calculations.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

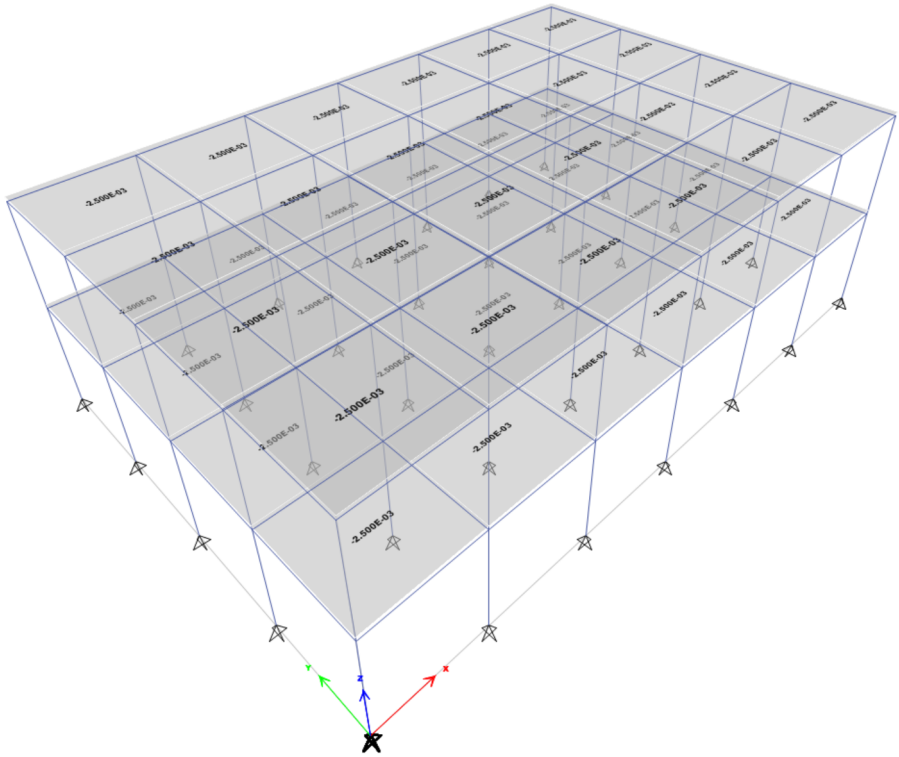

Figure 9: Automated live load is 2.5 kN/sq m on first floor slab and 1.5 kN/sq m on top slab as per IS code.

In Figure 9, the live load of 2.5 kN/m^2, as specified by IS 875 Part 2 for residential occupancy, is shown applied over the slab areas. The Python script referenced usage-type mappings to assign the correct value programmatically. Loads were represented as `IfcStructuralPlanarAction` entities and connected via `IfcRelConnectsStructuralActivity`, ensuring traceable and reusable load definitions within the IFC structure.

Table 3: Summary of Load Assignments and IFC Mappings for Two-Storey Model

Structural Element	**Load Type and Value**	**IFC Entity Used**
Beams (all storeys)	Dead Load: Self-weight (calculated via $q_d = \rho \cdot V$)	`IfcStructuralLoadLinearForce`, `IfcStructuralLinearAction`
Slabs (all storeys)	Dead Load: Self-weight (area-based planar force)	`IfcStructuralLoadPlanarForce`, `IfcStructuralPlanarAction`
Slabs (residential)	Live Load: 2.5 kN/m^2 (IS 875 Part 2)	`IfcStructuralLoadPlanarForce`, `IfcStructuralPlanarAction`
Load Grouping	Dead + Live Load Cases	`IfcStructuralLoadCase`, `IfcStructuralLoadGroup`
Load-Element Linking	All structural elements	`IfcRelConnectsStructuralActivity`, `IfcRelAssignsToGroup`

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

The results of this extended validation shows:

- The scalability of the pipeline to multi-storey systems without additional manual intervention.
- Accurate load generation and assignment for both linear and planar structural members.
- Full semantic enrichment of the IFC file using open standards and IS code-driven load intensities.
- Successful interpretation of the model in the analysis platform with correct load magnitudes and placements.

This experiment reinforces the generalizability of the BIM-driven automation approach across varying structural complexities and highlights the viability of using IFC as both a geometric and analytical data carrier in model-based engineering workflows.

5 Conclusion

This study introduced a platform-independent, BIM-based automation framework for structural load assignment, focusing on dead and live loads. By leveraging parametric modeling and programmable IFC augmentation, the workflow eliminates the need for manual load input by extracting geometric and material information directly from the building model and generating standardized load definitions based on national design codes.

The results from two experimental implementations — a single-storey frame and a two-storey building — confirmed the robustness and accuracy of the method. Graphical outputs validated correct geometric-to-load mapping, and the numerical magnitudes matched expectations derived from IS 875 (Part 1 and 2) standards. Semantic entities such as `IfcStructuralLoadLinearForce`, `IfcStructuralPlanarForce`, and `IfcRelConnectsStructuralActivity` were programmatically generated to support structural analysis-ready models. The approach ensured compliance with the IFC4 schema, preserving unit integrity and data traceability throughout the process.

Compared to traditional workflows that are often proprietary or semi-automated, this methodology presents a transparent, repeatable, and fully scriptable alternative. The use of open standards ensures that the solution is both interoperable and extensible, offering flexibility to adapt across different structural systems and project scales.

5.1 Future Work

Future development of this framework will focus on expanding the scope and adaptability of automated load modeling:

- **Lateral Load Integration:** Inclusion of wind and seismic loads using IS 875 Part 3 and IS 1893, with directionally resolved force vectors and load combinations.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

- **Digital Twin Connectivity:** Integration with real-time monitoring systems through IoT sensor data streams, enabling dynamic load updates and health assessment workflows.
- **AI-Based Load Prediction:** Implementation of intelligent agents to infer occupancy-based load patterns and recommend optimal structural configurations.
- **National Code Libraries:** Embedding automated lookup logic for IS codes and other regional standards to assign context-sensitive load templates.

The proposed methodology lays a strong foundation for the advancement of open, intelligent, and code-compliant structural BIM workflows.

References

[1] S. Lakušić, "Automated preprocessing of building models for structural analysis," *Journal of the Croatian Association of Civil Engineers*, 2022.

[2] T. Singh, M. Mahmoodian, and S. Wang, "Enhancing Open BIM Interoperability: Automated Generation of a Structural Model from an Architectural Model," *Buildings*, 2024. .

[3] M. Fawad et al., "Automation of structural health monitoring (SHM) system of a bridge using BIMification," *Scientific Reports*, vol. 13, 2023.

[4] V. Adamenko and O. Romanyshen, "Implementation of BIM and IT for structural analysis in RC frame design," *Building Constructions: Theory and Practice*, 2023.

[5] Y. Zheng, Y. Zhang, and J. Lin, "BIM–based time-varying system reliability analysis," *Journal of Building Engineering*, 2023.

[6] P. H. G. Alfaro et al., "BIM methods and CDE in professional education," *IOP Conf. Ser.: Earth Environ. Sci.*, vol. 1123, 2022.

[7] G. A. Ferreira, "BIM methodology implementation in structural design," *J. Civil Engineering and Environmental Sciences*, 2022.

[8] A. Gomes et al., "BIM in structural project: Interoperability and data management," *Applied Sciences*, 2022.

[9] A. P. Mello, R. C. de Freitas, A. P. de Almeida, and M. I. M. Paes, "AI-based automation for structural design of buildings," in *CILAMCE*, 2024.

[10] N. Bourahla, R. Tafraout, and A. A. Elbaz, "GA-based optimization of earthquake-resisting CFS structures in BIM," *Structures*, vol. 38, pp. 1309–1324, 2022.

[11] Y. Bashynskyi, Y. Barabash, and L. Tashlykova, "Metro loads on construction via BIM," *Journal of Civil Engineering and Management*, 2023.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

[12] D. Wang and H. Lu, "Development of a BIM Platform for the Design of Single-Story Steel Structure Factories," *Buildings*, 2024.

[13] X. Li, Y.-X. Dong, and W. Xiang, "Digital twin SHM using BIM," *Measurement Science and Technology*, 2024.

[14] G. R. Paul, "BIM in structural engineering: A study of interoperability," 2023.

[15] P. Palacz and M. Major, "Numerical-BIM integration for steel design," *BIM Modeling in Construction*, 2022.

[16] G. M. Azanaw, "Revolutionizing Structural Engineering: A Review of Digital Twins, BIM, and AI Applications," *Indian Journal of Structure Engineering*, 2024.

[17] G. Sibenik, I. Kovacic, and V. Petrinas, "Automated model preprocessing for structural analysis," *ISARC: International Symposium on Automation and Robotics in Construction*, 2021.

[18] X. Hu, G. Olgun, and R. H. Assaad, "An intelligent BIM-enabled digital twin framework for real-time structural health monitoring using wireless IoT sensing, digital signal processing, and structural analysis," *Expert Systems with Applications*, vol. 252, 2024.

[19] Z. Liu, H. Huang, and Y. Wang, "Automated framework for asphalt pavement design and analysis by integrating BIM and FEM," *Automation in Construction*, 2025.

[20] J. Ramonell and E. Chacón, "Automated load test processing using digital twins in railway bridges," 2022.

[21] Z. Liu et al., "Digital twins for marine SHM using BIM and fatigue models," *Engineering Structures*, 2020.

[22] S. Panagiotopoulos and K. Anyfantis, "Load identification for fatigue life assessment in ship hull structural elements," *e-Journal of Nondestructive Testing*, 2024.

[23] X. Lai et al., "Building a lightweight digital twin of a crane boom for structural safety monitoring based on a multi-fidelity surrogate model," *Journal of Mechanical Design*, 2022.

[24] D. Milanoski, G. Galanopoulos, and T. Loutas, "Digital-twins of composite aerostructures towards structural health monitoring," *2021 IEEE International Workshop on Metrology for Aerospace*, 2021.

[25] H. Lei et al., "FBG-based monitoring in bridges with digital twin integration," 2022.

[26] Y. Cheng et al., "Digital twin for hull monitoring in ocean vessels," *Marine Structures*, 2022.

[27] J. Zhang et al., "Finite element digital twin for structural behavior prediction," *Wind Energy Science*, 2024.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

[28] Y. Ye et al., "A dynamic data driven reliability prognosis method for structural digital twin and experimental validation," *Reliability Engineering and System Safety*, vol. 240, 2023.

[29] J. Ramonell et al., "Digital twin-based testing of steel cable net structures," 2022.

[30] Y. Ye et al., "Digital twin for the structural health management of reusable spacecraft: A case study," *Engineering Fracture Mechanics*, vol. 234, 2020.

[31] T. Rølvåg and T. Stranden, "Digital twin-based real-time structural load tracking for offshore cranes," 2022.