

Bidirectional BIM–FEM Interoperability for Reinforced-Concrete Structures through Systematic Topology Refinement, Pattern-Based Shared Naming, and Change Propagation

Mohammad Amin Khayat Baharlouie¹, Claudio Mirarchi²

¹Department of Civil and Environmental Engineering, Politecnico di Milano, Milan, Italy ²Department of Architecture, Built Environment and Construction Engineering, Politecnico di Milano, Milan, Italy

Corresponding author: Mohammadamin.khayat@mail.polimi.it

Abstract—This paper examines the persistent gap between BIM and FEM models in structural design, where existing solutions remain unable to achieve complete, accurate, and fully automatic bidirectional data exchange for three primary reasons. First, correspondence between models typically depends on native identifiers, resulting in lost links when identifiers change in either environment. Second, transferred models often fail to comply with the modeling conventions of the receiving environment because systematic refinement operations are generally unavailable, and manual adjustments are required before and after each transfer. Third, changes introduced between iterations are not tracked or classified, necessitating manual identification, coordination, and reapplication of every revision. This study develops a dedicated Revit–SAP2000 plugin that refines the raw analytical model through a fixed sequence of automated operations, assigns each element a pattern-based shared name derived from its intrinsic properties and spatial position in both environments, and utilizes baseline snapshots to detect, classify, and propagate changes in both directions. The complete workflow was tested on a real reinforced-concrete building, where it generated the FEM model without manual remodeling, reduced the modeling time from approximately 15 hours to 2 hours, and maintained equivalence between the two models through successive design changes.

Keywords—BIM–FEM interoperability; bidirectional data exchange; analytical model refinement; model synchronization; Revit; SAP2000

I. INTRODUCTION

In structural design practice, a building that is modeled in a BIM environment needs also to be analyzed in an FEM environment. When the two models are built separately, the engineers re-enter much of the same data into different systems, thereby duplicating the modeling effort and introducing errors [1]. Deriving the analytical model from the BIM model avoids this duplication, yet the transformation remains tedious and time-consuming because the designer must make extensive modifications to prepare the model for analysis [2]. The analytical model generated from the geometrical model follows a BIM-oriented logic and rarely meets the topological requirements of FEM calculations, where nodes, members and panels must be precisely connected so that loads, stiffness and internal forces transfer correctly. Moreover, the analytical model usually needs geometrical simplifications that should not be included in the building information model. This study addresses these challenges by working on the following research questions: RQ1: Can the geometrical model be automatically refined into a clean, integrated analytical model for direct transfer to FEM tools? RQ2: Can each element be assigned a unique name based on its intrinsic properties and spatial position, shared between the BIM and FEM models, so that the elements remain traceable across design iterations? RQ3: Can changes and new elements be detected through baseline snapshots and propagated to the counterpart? The primary objective of the work is to develop and validate an integrated workflow that can automatically convert a BIM model for concrete structures into an analytical model ready for FEM analysis. By investigating several transfer needs we developed dedicated conversion methodologies guaranteeing a bidirectional information flow.

II. RELATED WORK

Several works have been developed on the BIM to FEM topic. The available exchange routes between BIM and FEM models are direct links based on native program files, direct links through application programming interfaces, and indirect links through the open IFC standard [3]. The indirect route has repeatedly shown the loss and the distortion of the transferred data. In the reported experiments, IFC caused data losses between Revit and the major analysis tools [4], the sections were not recognized, the material properties were set to zero [5], and the analytical relationships were not adequately preserved, leaving slabs unconnected and nodes disconnected [6]. In contrast, the API-based exchange was found to enable more reliable data integration, although the available links operate smoothly only between tools from the same vendor [4]. Therefore, a cross-vendor pair requires a dedicated link that integrates the two programs' APIs.

Even via the API route, the resulting analytical model is raw rather than ready for analysis, as its geometry and connectivity still require extensive modifications, such as manual realignment of the analytical axes [6, 7]. Salamak et al. developed a Dynamo program to correct the faulty analytical model derived from the BIM geometry [8]. Thus, the first requirement for a practical link is a systematic sequence of automated refinement operations applied to the analytical model.

The focus of the recent automated solutions has primarily remained on the one-directional transfer of information for the initial generation of the analysis model, through technologies such as an ontology [9], an open BIM script [10], an IFC-based extraction of the geometry and the materials [11], and a combination of the Revit API and the SAP2000 OAPI [12]. However, a bidirectional workflow requires two additional capabilities: an element identity that remains stable across iterations and the

detection and classification of changes, so that each change can be applied to the corresponding element via that identity. The native identifiers cannot provide the first capability for three reasons. First, when an element is deleted and recreated with the same properties, it receives a new GUID, which breaks comparisons between design versions [13]. Second, an object's GUID can change during the transfer itself, when the model moves between the BIM and the FEM software in either direction [14]. Third, many elements do not remain intact. In the FEM environment, a surface element may be manually divided to prepare it for analysis or fragmented by its openings, while a curved member of any type may be broken into smaller parts. Once an object is sectioned, its original structure is no longer preserved, and each resulting segment is assigned a new GUID [1]. Even CSiXRevit, the dedicated commercial link for this exchange, acknowledges this limitation [15]. A stable identification must therefore be built on the properties of the elements themselves. This principle has previously been applied for different purposes. Wu et al. use invariant signatures derived from geometric and material properties to recognize objects in an architectural model and generate structural-analysis inputs through a one-way transfer [16]. Hsu et al. compose functional element names to automate element naming and bill-of-materials generation within the BIM environment [17]. Neither method was developed to preserve element correspondence through a shared name across BIM and FEM environments, and each addresses a fundamentally different objective. The present work adopts only the property-based principle and develops it into a shared identity that is assigned identically in both environments and remains valid when elements are divided in the FEM model. The second capability has been addressed only within the BIM environment, where design changes are semantically classified and detected across model versions [18]. However, these methods remain within a single program and do not transfer the detected changes to a corresponding model. None of the reviewed solutions preserves the identity of an element when it is rebuilt or divided into panels in the FEM environment, and none of them detects, classifies, and propagates the changes in geometry, sections, or materials from one model to another. This gap defines the present work, which refines the analytical model through a fixed sequence of automated operations, transfers it between BIM and FEM environments maintaining a coherent link to guarantee changes detection and propagation.

III. METHODOLOGY

The proposed approach overcomes the existing limitations through a continuous workflow that keeps the BIM model and its FEM counterpart synchronized in parallel, from initial model generation through subsequent design changes. To build this workflow, the recurring issues that affect the exchange were analyzed and grouped into classes, so that each class could be handled by a specific automated operation. The workflow begins with the 3D physical BIM model, from which an analytical model is generated automatically. This raw analytical model follows a BIM-oriented logic and does not meet the topological requirements of the FEM environment, so a fixed sequence of automated operations corrects it, each step addressing one class of issue and leaving the physical model unchanged. The

refined analytical model is transferred directly into the FEM environment and finalized for analysis. In the next stage, a pattern-based shared name is assigned to every element in both models, based on its intrinsic properties and spatial position, so that corresponding elements remain matched across the two programs through successive iterations. The workflow then records a baseline snapshot in each environment as the reference state for change tracking. This snapshot stores data for each element, along with the complete section and material catalogs, allowing later modifications to be identified by comparing the current model with its baseline. During the synchronized iteration phase, any change introduced in either environment is detected relative to the baseline, classified by the synchronization it requires, and applied to the corresponding element in the other environment via its pattern-based shared name. After each propagation, the names are reassigned, and a new baseline is recorded, so subsequent changes are measured against the updated reference. Figure 1 gives an overview of the complete workflow developed.

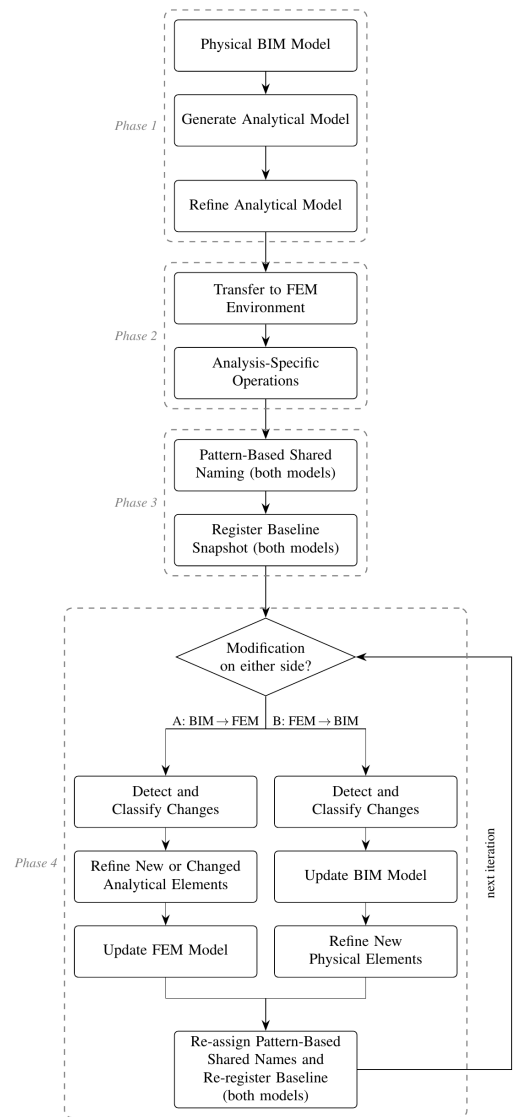


Figure 1. The proposed bidirectional BIM-FEM workflow.

The proposed approach was tested on a real case study, a

reinforced-concrete residential building under construction in the municipality of Monza, Italy. The structure comprises eight above-ground stories and one basement level, with a total height of approximately 30.9 m, and its load-bearing system combines reinforced-concrete columns, shear-wall cores, two-way slabs, and a continuous raft foundation enclosed by perimeter retaining walls. The building was selected because it reflects the geometric and modeling complexity of real structural projects rather than simplified benchmarks. Its shear walls and two-way slabs must be represented as area elements in the FEM environment and often require manual subdivision into smaller panels to ensure mesh quality and appropriate connectivity, as automatic meshing is frequently insufficient. The model also incorporates irregular slab and foundation layouts, openings, a complex retaining-wall perimeter, and eccentric columns, which collectively increase the complexity of the bidirectional exchange and render the building an appropriate case study for assessing the proposed workflow.

IV. IMPLEMENTATION

Selected software and developments

The proposed workflow was tested developing a dedicated Revit–SAP2000 Interoperability (RSI) Plugin, with a Revit side and a SAP2000 side. Both sides were developed entirely by the authors, including their user interfaces and the code that implements the methodology. The interfaces of the two sides are shown in Figure 2.

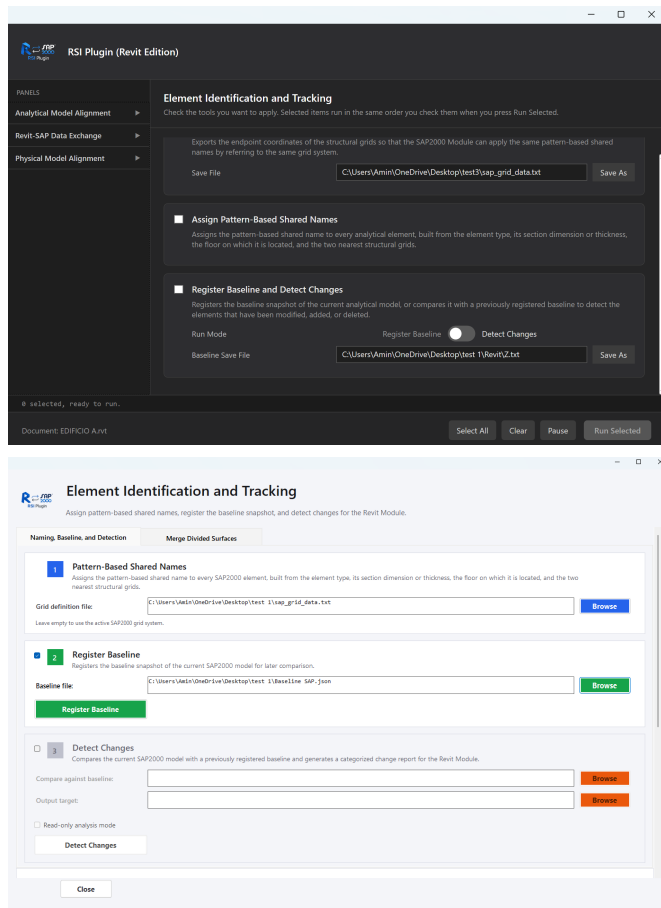


Figure 2. Interfaces of the RSI Plugin: the Revit side (top) and the SAP2000 side (bottom).

Autodesk Revit 2025 served as the BIM environment and SAP2000 2026 as the FEM environment. Both programs were selected for their widespread adoption and for their programming interfaces that enable operations to be automated. Revit offers a .NET application programming interface (API) accessible from Python, so the Revit side was developed in Python as a pyRevit extension. SAP2000 supports automation via the CSI Open Application Programming Interface (OAPI), so the SAP2000 side was developed in C# and runs within SAP2000, with direct access to the active finite element model.

The two sides are complementary, but the task allocation between them is asymmetric. The Revit side has direct access to the BIM model and can also control SAP2000 through the OAPI, whereas SAP2000 cannot modify the Revit model from within. Therefore, all transfer and update operations between the two environments are handled from the Revit side, while the SAP2000 side acts only on the FEM model.

Workflow operations

The implemented operations are organized into four sequential stages and each stage is described in the following subsections.

4.2.1 Refinement of the analytical model

The analytical model derived from the physical BIM model is not directly suitable for finite element analysis, in which nodes, members, and panels must be precisely connected to ensure the correct transfer of stiffness and internal forces. A set of refinement operations brings the raw analytical model to this state, modifying only the analytical model and leaving the physical BIM unchanged. These operations run in a fixed sequence, because each step depends on the corrections made by the previous ones. In the Revit plugin, they are organized into four panels: Vertical Alignment, Plan Alignment, Openings, and Curved Elements. Each operation is governed by user-defined parameters, such as a snap distance or a grouping tolerance, so that only the elements within the specified limits are adjusted.

The Vertical Alignment panel aligns the analytical elements to consistent reference planes. Its first operation consolidates the analytical elements of a selected level onto a target level and is applied when a floor contains minor local steps that have a negligible effect on the global response. Its second operation shifts the analytical slab panels from the top surface of the slab, where the BIM software places them, to the mid-plane of the physical thickness, where the shell idealizes the slab in the FEM model. Connected panels of different thicknesses are brought onto a common plane to preserve their connectivity. After these adjustments, the analytical member ends, and wall edges may fall slightly above or below the floor reference. The third operation snaps each analytical node onto the analytical slab panel of its floor, within the defined snap distance and edge tolerance.

The Plan Alignment panel corrects the horizontal position and connectivity of the analytical elements. These corrections are needed because each member is represented by its centerline or center plane, so physically connected elements do not always meet at a common node. The first operation treats physically eccentric columns whose center lines do not coin-

cide across levels. Since the analytical column members pass through the centers of the physical columns, these analytical members are also offset from one another. It groups the affected analytical column members within an assigned tolerance and aligns them to a common centerline. Each offset can be neglected or preserved as a geometric eccentricity, which is later reproduced in the FEM model through the insertion point of the corresponding frame objects. A second operation corrects the rotation of the analytical column sections, a mismatch found mainly in columns inclined in plan.

Thereafter, the analytical beam members are realigned in two stages. The first stage aligns the ends of the primary beams with the fixed column nodes, since these ends can detach when the corrected analytical column members move or when beams and columns of different widths meet at their faces. The second stage closes the small gaps between the primary and the secondary beams, so the floor framing remains a continuous analytical network. In the following step, the analytical wall panels are treated for three conditions: a physically connected wall and column whose analytical connection is not properly represented because the analytical wall panel and the analytical column member pass through their respective center planes and center lines, panels of different thicknesses that receive offset center planes along a shared face, and small gaps left along a continuous wall line by the earlier corrections. In each case, the panels are shifted and joined so that the wall system stays continuous. Finally, the edges of the analytical slab and foundation panels are reshaped. The physical slabs cover the column footprints and extend to the outer faces of the walls, so their analytical edges pass alongside the analytical column nodes and stop outside the analytical wall panels. These edges are corrected to pass through the nodes and meet the wall panels at their final positions.

The Openings panel carries the slab voids through the exchange. Its first operation handles the analytical openings bounded by walls, such as those around the staircase and the elevator shaft. The physical edge of such an opening follows the outer face of the wall, whereas the analytical wall panel lies on its centerline. Therefore, the affected edges of the analytical opening are repositioned onto the plane of the host analytical wall panel, so the opening connects to the wall on that side. The second operation simplifies the opening boundaries. Two openings facing each other at a small distance are merged, and the opening edge that ends near the top or bottom node of the host wall panel is moved to that node. These corrections remove the thin residual wall strips and the small irregularities that would complicate the meshing stage in the FEM environment.

The Curved Elements panel handles two cases: a fully curved wall, and an analytical slab or foundation panel with curved boundary portions. Because the FEM software approximates such geometry with straight segments only at import, the correspondence between the two models would be lost. To prevent this, the curved analytical wall is reconstructed as a sequence of straight analytical wall panels along its centerline, and only the curved portions of the analytical slab and foundation boundaries are replaced with straight segments. The number of segments is user-controlled, so the discretization is defined explicitly in the BIM environment before the transfer.

4.2.2 Transfer to the FEM environment and analysis preparation

After refinement, the analytical model can be transferred directly to SAP2000. The structural grids and story levels are recreated first, from the shared coordinate origin of both models. Next, the materials and the frame and shell sections are regenerated. The analytical elements are reconstructed as frame and area objects at their precise locations, with the appropriate sections and materials assigned. This direct programmatic transfer replaces file-based exchange and preserves the full geometric and semantic content of the analytical model, so the structure is not modeled twice and the resulting inconsistencies are avoided.

In the FEM environment, the model is finalized through the operations required for structural analysis. These include the division and meshing of the slab, wall, and raft foundation panels, the definition of load patterns, and the generation of load combinations in accordance with the design codes adopted for the specific case (for the developed case study we used the Italian NTC2018 and the relevant Eurocodes [19]). The seismic settings are also defined here, including the response spectrum functions and the mass source. Finally, the restraints, member end releases, floor diaphragms, and other analysis parameters are assigned to represent the building's structural behavior.

4.2.3 Pattern-based shared naming and baseline registration

To enable a bidirectional workflow, corresponding elements must retain identical names in both environments at every iteration, so that changes detected in one environment can be accurately applied to the same element in the other. As discussed in Section 2, the native identifiers cannot serve this role. To overcome existing limitations, a naming convention was developed in which each element is assigned a reproducible name based on its intrinsic properties and spatial position. The resulting name, referred to as the pattern-based shared name, does not replace the native identifier and serves as the stable reference for matching the corresponding elements. On both the BIM and FEM software sides, naming is performed using the proposed approach (integrated into the developed plugin), which reads these data from the analytical elements in the BIM environment and from the corresponding frame and area objects in the FEM environment. The name joins the element type, the section dimension for columns and beams, the thickness for slabs, or the width for walls, the floor of the element, and the two nearest structural grids, one from each of the two grid families commonly labeled with letters and numbers.

The grid coordinates used for naming are read from the endpoints of the Revit grids and exported to a grid file, because inclined grids cannot be drawn in SAP2000. The SAP2000 plugin loads this file before naming, so both environments derive the grid references from a single identical source. The grids are selected with respect to the center of a column, the midpoint of a beam, or the centroid of a wall panel. Slabs, raft foundations, and openings carry no grid reference and are identified by their type and floor, with the thickness added for slabs and raft foundations.

The floor field is derived in Revit from the positions of the

analytical slab panels. The analytical slab panel nearest to the ground level defines floor 0. The analytical slab panels of the floors below this reference receive consecutive negative numbers, while those of the floors above it receive consecutive positive numbers. Beam members, slab panels, and openings take the floor on which they lie, whereas columns and walls take the floor of their lowest end. In SAP2000, the floor field is read from the story levels of the grid system. As described in the transfer stage, these story levels are recreated from the elevations of the analytical slab panels of the Revit model, so both environments assign the same floor number to each element.

When several elements would receive the same name, a numeric counter ordered by the distance of each element from the model origin distinguishes them. When two of these elements also lie at the same distance from the origin, the counter is ordered by the Y-coordinate distance of their centroids from the origin. Every element therefore receives a distinct name, and repeated, symmetrical, or closely spaced elements are separated by the same rule in both environments. A column with a 450×800 section, located on the fifth floor near the intersection of grids C and 2, receives the name *C_450x800_5_C,2.

For the divided panels of the FEM environment, the naming first recovers the continuous element. The slab panels that lie on the same floor, share the same thickness, and are connected through common edges are treated as a single slab. In the same way, the wall panels with the same width, floor, and orientation that are connected through common edges are treated as a single wall, and the grid references of its pattern-based shared name are selected with respect to the centroid of the recovered region. Therefore, a set of divided panels and the corresponding continuous BIM element share the same name and remain traceable across all cycles.

Name storage follows the logic of each program. In the BIM environment, the native identifier of an analytical element cannot be edited, so the name is stored in a dedicated parameter of each analytical element. In the FEM environment, every object keeps a unique native label, so the name is stored as a named group that collects all divided panels of the same continuous element.

Once all elements have been named, a baseline snapshot is registered in each environment creating a baseline text file. For each element, the baseline stores the native identifier, the pattern-based shared name, the assigned section and material, and the geometry, described by the end-point coordinates and the rotation for linear elements and by the boundary vertices for surface elements. The complete section and material catalogs are stored with their full properties, and the baseline file is later used for comparison and change detection. Figures 3 and 4 show examples of the recorded baseline data in the two environments.

The order of the workflow keeps this naming system stable across the design iterations. Events such as a change in the grids or story levels, or the insertion of a new element close to the origin among similar elements distinguished by counters, cannot disturb the correspondence, because a change applied in either model does not update the pattern-based shared names at that moment. In every cycle, the changes are first detected and classified relative to the last registered baseline. The coun-

terpart model is then updated through the pattern-based shared names, which at this point are still identical for all elements in the two models, and the grids and story levels of the destination model are updated automatically from the source model when needed. Once the update is complete and the two models are identical again, the pattern-based shared names are reassigned in both models according to the new properties of the elements. Because the two models are in the same reference state, every element again receives the same name in both environments, and a new baseline is registered for the next iteration. The pattern-based shared names thus act as a bridge that carries each change correctly from one environment to the other and are themselves renewed automatically at the end of every cycle.

```
"c6d80aa5-e80d-4454-9be8-c177474f221a-00079441": {
  "unique_id": "c6d80aa5-e80d-4454-9be8-c177474f221a-00079441",
  "element_id": "496705",
  "comment": "*C_350x700_2_D,3",
  "analytical_id": "712813",

  "category": "Structural Columns",
  "family": "M_Concrete-Rectangular-Column",
  "type": "STR - COL- 350 X 700",
  "section_dimension": "350x700",
  "section_id": "372008",
  "material": "Concrete - Columns - C32/40",
  "material ID": "443265",

  "base_level": "PIANO 2_ STRU_ GE",
  "base_offset": 0,
  "top_level": "PIANO 3_ STRU_ GI",
  "top_offset": 0,
  "real_length_m": 3.05,

  "position": [33.63, -46.32],
  "rotation": 0,
}
```

Figure 3. Recorded baseline data of an element in Revit.

```
},
"Area:4944": {
  "name": "4944",
  "label": "4944",
  "guid": "c4effd72-14f4-4c95-9ff9-821c4a297a2b",
  "type": "Area",
  "groups": [{"W_400_1_D,4"}],
  "props": {"Section": "STR - PSHR - 400", "Material": "None", "NumPoints": "4",
  "X_P0": "15.7", "Y_P0": "-14.094", "Z_P0": "5.825", "X_P1": "14.45", "Y_P1": "-14.094", "Z_P1": "5.825", "X_P2": "14.45", "Y_P2": "-14.094", "Z_P2": "6.65",
  "X_P3": "15.7", "Y_P3": "-14.094", "Z_P3": "6.65", "Rotation": "0"}
},
"Area:4945": {
  "name": "4945",
  "label": "4945",
  "guid": "90917b65-909d-4910-87b9-2b00c6e4a8cf",
  "type": "Area",
  "groups": [{"W_400_1_D,4"}],
  "props": {"Section": "STR - PSHR - 400", "Material": "None", "NumPoints": "4",
  "X_P0": "19", "Y_P0": "-14.094", "Z_P0": "6.65", "X_P1": "19.825", "Y_P1": "-14.094", "Z_P1": "6.65", "X_P2": "19.825", "Y_P2": "-14.094", "Z_P2": "5.825",
  "X_P3": "19", "Y_P3": "-14.094", "Z_P3": "5.825", "Rotation": "0"}
}
```

Figure 4. Recorded baseline data of some elements in SAP2000.

4.2.4 Change detection and propagation of the changes to the corresponding model

Structural design is iterative, revising member properties through repeated cycles of analysis and redesign until the strength and serviceability requirements are met. During this process, the engineer may modify, delete, or create an element in the FEM model, while the BIM model may be revised in parallel to accommodate architectural changes. In both situations, the workflow must identify what has changed since the previous synchronization, so that only the affected elements are exchanged, and the two models remain exact counterparts. For this purpose, change detection is performed in the environment

where the modifications originate by comparing the current model with a manually selected baseline, which may be any earlier reference state of the workflow.

The detected differences are classified according to the synchronization behavior they require and written to a categorized text file, referred to as the change report. The report is organized into the following categories: (1) Modified Section and Material Definitions, (2) Elements Affected by a Modified Section Dimension, (3) Elements with a Reassigned Material, (4) Elements with a Reassigned Section, and (5) Deleted Elements or Elements with a Modified Position or Boundary. Each affected element is registered under its category via its pattern-based shared name, and the entry stored for it depends on how the change is applied later. For a modified section or material catalog, the report stores its name and the name of the changed parameter, and the exact new value is read from the same definition in the source model during the update. Renamed sections and materials are recognized by their identifiers as existing definitions rather than new ones, so their previous names are also recorded and each counterpart is located under its old name before it is updated with the new name and properties. For the reassignment categories, the newly assigned section or material is written next to each element's pattern-based shared name, so the receiving program selects the element and assigns the definition without rebuilding it. In contrast, a change in geometry, position, or boundary cannot be expressed as a parameter update, so the element is deleted and reconstructed from its updated source, whereas a deleted element requires only the removal of its counterpart. Furthermore, new elements are recognized when their identifiers are absent from the baseline, and their properties are not written to the report, because the update transfers them directly from the model.

While the change report classifies modifications, it does not include geometric data. Therefore, each detection run also generates a model-side container that stores the geometry of changed and new elements: a dedicated Revit view for the Revit-to-SAP2000 direction and a trimmed SAP2000 model for the SAP2000-to-Revit direction.

Manual baseline selection does not place either model at risk because change detection is non-destructive. It only compares the current model with the selected baseline and generates a change report together with the corresponding model-side container, without modifying either model. The user may therefore repeat the detection against different baselines and discard outdated results. The counterpart model is modified only during the separate update step. To ensure that it is updated correctly, the user must select the change report and its associated model-side container generated by the detection run against the most recent baseline.

In the Revit to SAP2000 direction, the detection run first regenerates the analytical model of the changed elements, because a modification to a physical member is not automatically reflected in its analytical counterpart. The analytical element of each modified physical member is deleted and rebuilt with the updated properties, and a new analytical element is created for each new physical member. All rebuilt and new analytical elements, together with their physical members, are collected in a dedicated view named Modified Elements (Revit to SAP). Since these analytical elements are generated in their raw form,

the refinement operations are reapplied before transfer, using the most recent input values stored by each tool, thereby avoiding the need to re-enter the parameters. In the report, each of these elements stores its previous pattern-based shared name together with the identifier of its rebuilt analytical element. When the update is run, the user picks the affected elements from this view. The previous name locates each target in SAP2000, whereas the rebuilt identifier matches every picked element against the report categories, so an element already handled by the report is not transferred and recreated. The definition-level entries are applied first, so every element that uses those definitions is revised automatically. After this, the elements with a reassigned section or material receive their new assignment directly through their pattern-based shared names. Once all updates have propagated, naming and baseline registration are repeated in the SAP2000 model, so both environments share an identical reference state again.

The SAP2000 to Revit direction follows the same category logic but differs in two respects. First, SAP2000 cannot modify the Revit model from within its environment, so it exports a trimmed SAP2000 model, created as a copy of the main model with all unchanged elements removed, which the Revit plugin reads together with the report. Second, the slab and wall panels are typically divided into several area objects during analysis preparation, where several panels carry different native labels under a single pattern-based shared name. For this reason, only the pattern-based shared names are written under the categories, which keeps the report compact. This choice introduces a filtering problem, because the report stores the previous pattern-based shared names that locate the corresponding Revit elements, whereas the trimmed model carries the new names assigned during the updating of pattern-based shared names that follows report generation. Without an additional rule, the plugin could not distinguish the imported elements that must be rebuilt in Revit from those whose changes are already handled as direct updates. To resolve this, the SAP2000 report includes an additional category, Elements Updated in Place (Not Rebuilt in Revit), which stores the new pattern-based shared names of all directly updated elements from the report. Every imported element whose current name appears in this category is filtered out, so only the truly new elements and those with a modified position or boundary are rebuilt from the trimmed model.

Before the transfer, the SAP2000 plugin merges the divided panels that share the same pattern-based shared name and belong to a single continuous slab or wall into one area object, so each changed surface arrives in Revit as a unified element. Opening regions, represented in SAP2000 as panels with the None section, are retained alongside the merged surfaces, so each void is later recreated in Revit as an opening in the host slab. The update to the Revit model is performed by the Revit plugin, which reads the trimmed model and the change report together, so that the model supplies the geometry and the report supplies the classification. Based on these two inputs, the direct updates are first applied to both the analytical elements and their associated physical members, which are identified by their recorded pattern-based shared names. Thereafter, each element to be rebuilt or created is reconstructed in two linked steps: the analytical element is generated from the imported

geometry, and the physical member is produced from it. In this reconstruction, each physical slab is created so that its analytical panel lies at the mid-plane, preserving consistency with the FEM representation. Once the update is complete, all modified and new elements are collected in a dedicated view named Modified Elements (SAP to Revit) for review.

Because these physical members originate from the FEM topology, they do not yet comply with BIM modeling conventions, and a reverse counterpart of the analytical refinement is applied to them. The vertical alignment snaps the endpoints of the reconstructed columns and walls to the appropriate top or bottom faces of the physical slabs and aligns each beam's top surface with the slab's top face, while the element types that receive this alignment remain selectable by the user. The plan alignment moves the reconstructed slab and foundation boundaries outward from the member centerline to the outer faces of the adjacent columns and walls, with an optional offset, so the slabs cover the supporting members. Adjustments that do not follow a recurring pattern are left for manual review in the dedicated view. As the final step of the cycle, naming and baseline registration are repeated in the updated Revit model, and the bidirectional exchange closes with both models sharing the same reference state.

V. RESULTS

The complete workflow was applied to the case study model. The refinement operations were executed in their defined sequence on the raw analytical model generated by Revit, and the entire analytical model was corrected without any manual editing. The result is a single, connected analytical model that satisfies the topological requirements of the FEM environment and was transferred directly by the developed tool into SAP2000. The transfer reproduced all 276 frame elements and 142 surface elements of the refined analytical model, together with the complete catalogs of 24 sections and 7 materials, without any loss. Figure 5 shows the three stages of this process, from the physical BIM model through the refined analytical model to the FEM model generated in SAP2000.

Thereafter, the analysis-specific operations of the workflow were applied to the generated model, and the structural analysis was executed in SAP2000. Table 1 reports the periods and participating mass ratios of the first three modes, and Figure 6 shows the corresponding modal shapes. The first mode is translational in the Y direction with a period of 1.01 s, the second mode is predominantly torsional, and the third mode is translational in the X direction. This regular sequence of two translational modes and one torsional mode, together with the absence of spurious local modes among the computed modes, indicates that the transferred elements form a single connected load-resisting system. With 15 computed modes, the cumulative participating mass reaches 88.7% in the X direction and 93.1% in the Y direction, satisfying the 85% requirement of NTC2018 for response spectrum analysis [19]. These results confirm that the analytical model was correctly and completely transferred to SAP2000 and that the analysis was executed without errors.

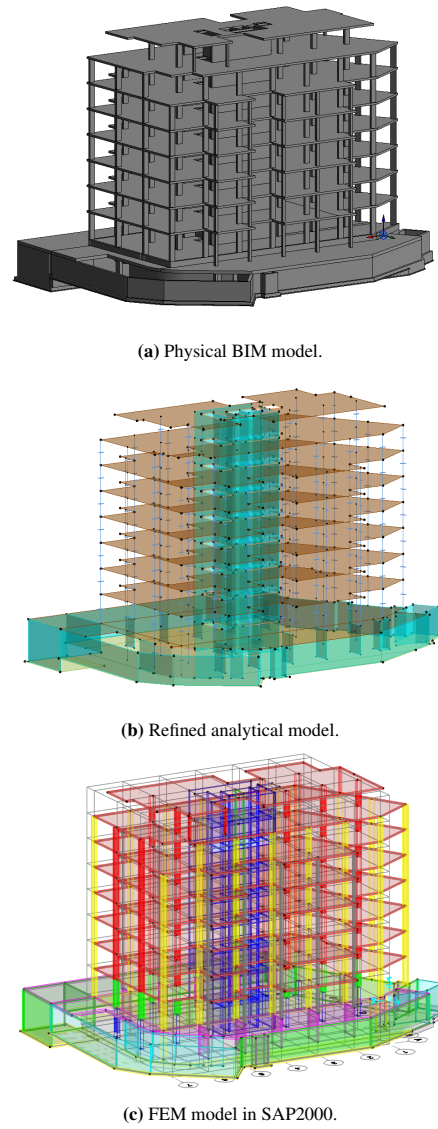


Figure 5. Stages from the physical BIM model to the FEM model generated in SAP2000.

Table 1. First three modal periods and participating mass ratios of the generated model.

Mode	Period (s)	UX	UY	RZ
1	1.010	0.001	0.447	0.120
2	0.895	0.037	0.108	0.285
3	0.763	0.542	0.002	0.026

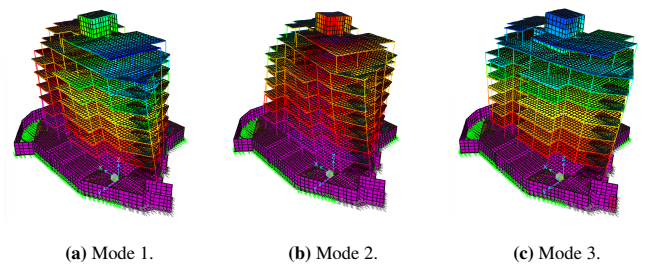


Figure 6. First three modal shapes of the generated model.

To evaluate change propagation, a first cycle was performed from Revit to SAP2000 by applying several modifications to floor 5 of the model. The section of all 350×350 columns was changed to 450×450 , the thickness of the slab panel of the same floor was modified, the length of a core wall was changed, two columns were deleted, and a new column was added, as shown in Figures 7 and 8. The detection run produced the categorized change report, an excerpt of which is summarized in Table 2. The report registered 65 changed elements: 60 columns affected by the modified section dimension, 1 slab with a reassigned section, 3 elements deleted or geometrically modified, and 1 new column. Subsequently, the rebuilt and new analytical elements were reviewed in the Modified Elements (Revit to SAP) view and refined again. The SAP2000 model was automatically updated by the plugin, with each change applied to the corresponding object via its pattern-based shared name. As shown in Figure 9, all the modified elements were correctly updated, and the two models again became equivalent counterparts.

Table 2. Excerpt of the categorized change report produced by the RSI Plugin (Revit side).

Modified Section and Material Definitions			
STR-COL-350×350 → STR-COL-450×450: renamed, dimension changed			
Elements Affected by a Modified Section Dimension			
Element	Phys ID	Old ID	New ID
*C_350x350_0_A,2	684159	713159	747334
*C_350x350_0_A,3	684171	713165	747323
*C_350x350_0_A,4	684177	713171	747288
		⋮	
*C_350x350_5_F,6	683817	713104	747310
Elements with a Reassigned Material			
None			
Elements with a Reassigned Section			
*SLB_250_5: STR-SLB-350 (Phys ID 497142, Old ID 712909, New ID 747324)			
Deleted Elements or Elements with a Modified Position or Boundary			
*W_300_5_C,4(3)	*C_300x700_5_E,2(1)	*C_300x700_4_E,2(1)	

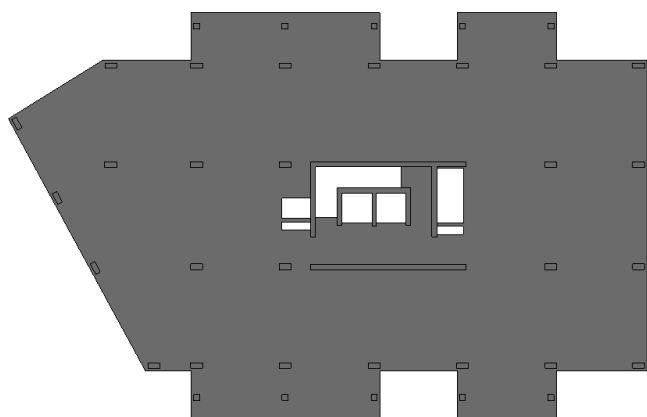


Figure 7. Floor 5 of the Revit model before the applied changes.

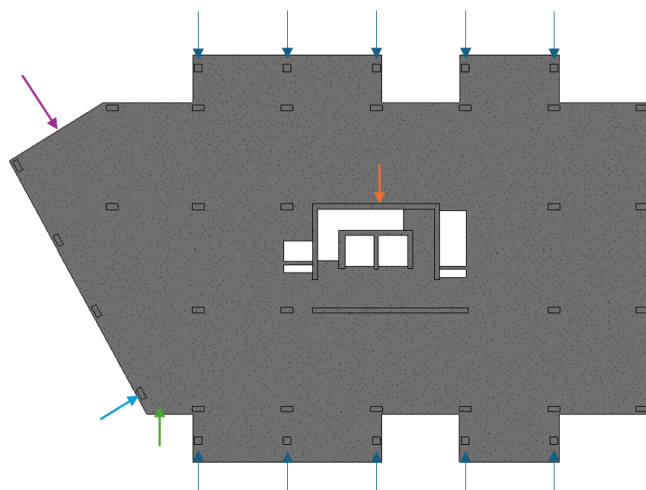


Figure 8. Floor 5 of the Revit model after the applied changes, with the columns resized to 450×450 , the slab section reassigned, the core wall shortened, two columns deleted, and a new column added.

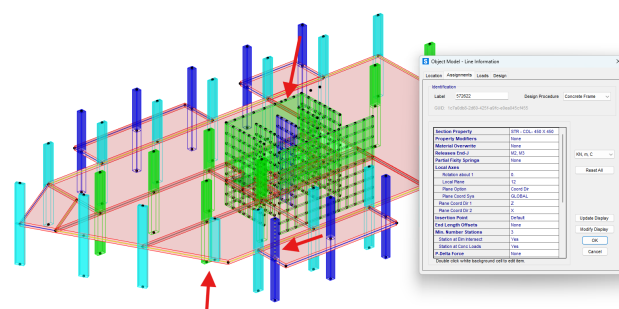


Figure 9. SAP2000 model after the propagated changes.

The reverse direction was evaluated in a second cycle by modifying the SAP2000 model, as shown in Figures 10 and 11. The 350×700 column section was resized to 450×800 and renamed, the material assigned to the columns was renamed, the separate balcony panels of the second-floor slab were merged into a continuous panel on each side, and the two central shear walls were reassigned the STR-PSHR-400 section. The detection run generated the categorized change report, summarized in Table 3, together with the trimmed model of the changed and new elements. The report registered 65 changed elements: 62 columns affected by the modified section dimension, 2 shear walls with a reassigned section, and 1 slab with a modified boundary. Before the transfer, the Merge Divided Surfaces tool recombined the divided panels of the modified slab and walls into continuous surfaces while retaining the opening panels (Figure 12). The update was applied to the Revit model, where the existing elements were updated in place, and the rebuilt physical members were corrected using the physical elements refinement tools, as shown in Figure 13.

Table 3. Excerpt of the categorized change report produced by the RSI Plugin (SAP2000 side).

Modified Section and Material Definitions	
Concrete-Columns-C32/40 → Concrete-C32/40: renamed	
STR-COL-350×700 → STR-COL-450×800: renamed, t_3 and t_2 changed	
Elements Affected by a Modified Section Dimension	
*C_350x700_0_C,2	*C_350x700_1_C,1
*C_350x700_1_C,2	*C_350x700_1_C,3
⋮	⋮
*C_350x700_7_D,3	*C_350x700_7_D,6
Elements with a Reassigned Material	
None	
Elements with a Reassigned Section	
*W_300_1_D,4: STR-PSHR-400	*W_300_2_D,4: STR-PSHR-400
Deleted Elements or Elements with a Modified Position or Boundary	
*SLB_250_2	
Elements Updated in Place (Not Rebuilt in Revit)	
*C_450x800_0_C,2	*W_400_1_D,4
*C_450x800_1_C,1	*W_400_2_D,4
⋮	⋮

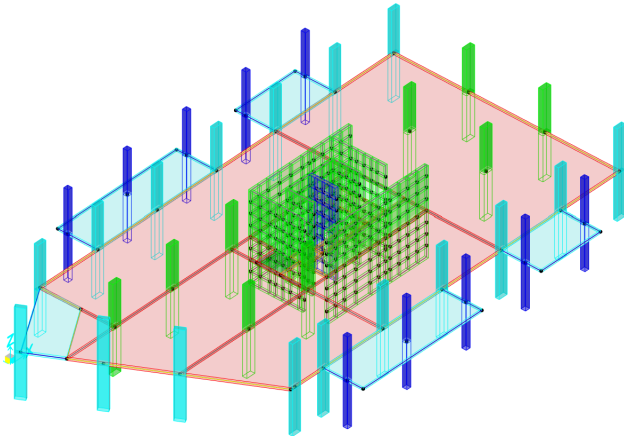


Figure 10. SAP2000 model before the applied changes.

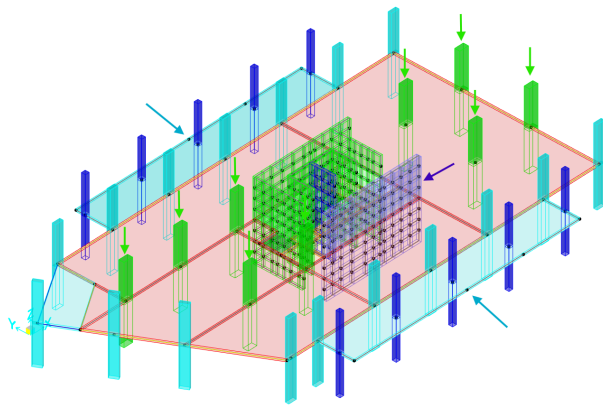


Figure 11. SAP2000 model after the applied changes, with the columns resized to 450×800 , the shear walls reassigned the STR-PSHR-400 section, and the balcony panels merged into a continuous slab boundary.

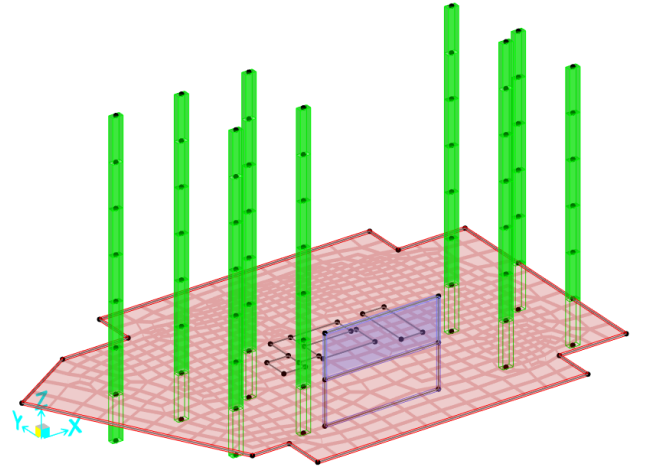


Figure 12. Trimmed SAP2000 model containing only the changed and new elements, after merging the divided panels into continuous surfaces with the opening panels retained.

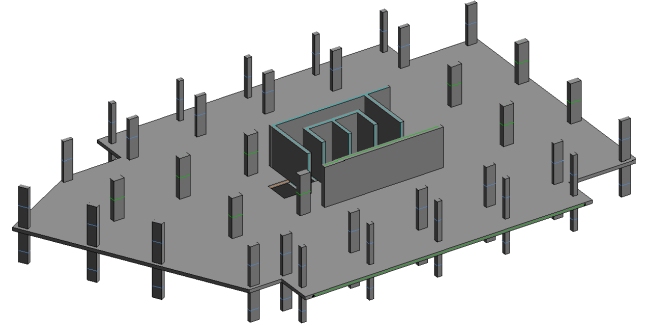


Figure 13. Updated Revit model after SAP2000 change propagation.

In both directions, every detected change was propagated and applied without manual remodeling. After each update, the pattern-based shared names were reassigned, and a new baseline was registered in both models. The two models thereby returned to an equivalent state and were ready for the next change cycle.

VI. DISCUSSION

The results confirm that the proposed workflow fulfills its central objective within the scope of the tested case study. The BIM model remained a correct representation of the building as it will be constructed, while the calculation model was generated from it automatically and kept coherent across the design iterations. To quantify the modeling effort, the manually created reference model was developed in parallel up to the stage preceding the analysis-specific operations, corresponding to the state shown in Figure 5. Reproducing the geometry of the structure in SAP2000 required approximately 15 hours for an engineer with intermediate experience, whereas the same state was reached from the Revit model in approximately 2 hours with the developed workflow, including the verification of the element connections and the selection of the appropriate input values for each refinement tool. In addition, the complex geometries of the case study were reproduced in the FEM environment exactly as defined in the BIM model, so the geometric simplifications traditionally introduced in manual modeling to save time are not necessary. An attempt was also made to

generate the same model using CSiXRevit, the dedicated commercial link between Revit and SAP2000. However, the tool is limited to data transfer and does not refine the raw analytical model created in Revit. Consequently, each transferred analytical element would require manual verification and correction in SAP2000, a process rendered impractical by the geometric complexity of the case study. The IFC-based route was likewise excluded because the reported losses of geometric and connectivity information result in an incomplete model that still requires substantial manual reconstruction [4–6].

The strongest implications of this proposed workflow concern updating the models. In current practice, every design revision requires direct coordination between the structural analysis team and the BIM modeling team, and each change must be manually implemented again in the other environment, increasing both the time required and the likelihood of errors. Moreover, minor geometric modifications of a few centimeters are often not carried over to the FEM model because their effect on internal forces is negligible, causing the two representations to diverge progressively during design iterations. The proposed method reverses this condition. At every stage, the combination of the registered baseline and the change detection identifies, classifies, and manages all the modifications introduced since the previous synchronization, so no change is forgotten. As demonstrated by the test cycles, the corresponding model is updated automatically and without any error, with an effort reduced to running the tools and reviewing the categorized report. Consequently, the two models remain aligned through all the iterations, and the manual effort required to maintain this alignment is reduced substantially.

Finally, the proposed concepts are not bound to the selected programs. The method can be re-implemented for other BIM and FEM platforms that expose programming interfaces by adapting the operations to the representation logic of each environment, so that changes at each stage are detected relative to the previous baseline, classified, and applied to the corresponding elements via their pattern-based shared names.

VII. CONCLUSION

This paper presented a bidirectional workflow that maintains coherence between a BIM model and its FEM counterpart throughout the structural design process, implemented via the RSI Plugin for Revit and SAP2000. The workflow refines the raw analytical model through a fixed sequence of automated operations and transfers it directly into the FEM environment. Every element receives the same pattern-based shared name in both programs, and a baseline lets any later change be detected, classified, and applied to the corresponding element through that name. In the SAP2000-to-Revit direction, the rebuilt physical members are corrected using automated operations to comply with BIM modeling conventions. When tested on a real reinforced-concrete building, the workflow generated the calculation model without manual remodeling, reduced the modeling time from approximately 15 hours to approximately 2 hours, less than one-seventh of the manual reference, and kept the two models equivalent as changes were propagated in each direction.

The scope of this validation defines the current limitations. The case study contains two-way flat slabs without beams,

so the beam support included in every tool has not yet been verified on a real structure. The method was also developed for reinforced-concrete structures; its foundation operations were verified only on a raft foundation, and the implemented refinement operations address the recurring issues identified so far, thereby fully preparing the case study model.

Future work follows directly from these boundaries. The beam support and performance on other foundation types will be tested in additional case studies, as needed. Applying the workflow to buildings with different structural systems may also reveal new refinement patterns, whether in the analytical model in Revit or in the physical members rebuilt from SAP2000, and these patterns can be automated and added in the same manner. Since the scope of this study covered reinforced-concrete structures, the same methodology can be adapted to steel buildings to meet their specific requirements, thereby progressively making the workflow applicable to all structural types.

REFERENCES

- [1] M. F. Muller, A. Garbers, F. Esmanioto, N. Huber, E. R. Loures, and O. Canciglieri, “Data interoperability assessment though IFC for BIM in structural design—a five-year gap analysis,” *Journal of Civil Engineering and Management*, vol. 23, no. 7, pp. 943–954, 10 2017.
- [2] I. J. Ramaji, S. M. Asce, A. M. Memari, and F. Asce, “Interpreted Information Exchange: Systematic Approach for BIM to Engineering Analysis Information Transformations,” 2016.
- [3] M. Aldegeily, “From Architectural Design to Structural Analysis: A Data-Driven Approach to Study Building Information Modeling (BIM) Interoperability,” Tech. Rep., 2018. [Online]. Available: https://scholarworks.wmich.edu/masters_theses
- [4] D. İlipinar, B. O. Ay, and M. K. Pekerçli, “Comparison of Data Integration Methods in BIM Tools for Structural Engineering: An Evaluation through Structural Analysis Tools,” *Gazi University Journal of Science*, vol. 38, no. 3, pp. 1062–1078, 9 2025.
- [5] T. Drávai, H. Khalyar, and G. Nagy, “The effect of interoperability between bim and fem tools on structural modeling and analysis,” Master’s thesis, Aalborg University, Aalborg, Denmark, 2016. [Online]. Available: https://projekter.aau.dk/projekter/files/239457557/The_effect_of_interoperability_between_BIM_and_FEM_tools_on_structural_modeling_and_analysis.pdf
- [6] P. Colajanni, L. Inzerillo, A. Pisciotto, F. Acuto, K. Mantalovas, and G. Di Mino, “Architectural and Structural Interoperability in the BIM Design Workflow,” *Buildings*, vol. 15, no. 24, 12 2025.
- [7] I. J. Ramaji and A. M. Memari, “Interpretation of structural analytical models from the coordination view in building information models,” *Automation in Construction*, vol. 90, pp. 117–133, 6 2018.
- [8] M. Salamak, M. Jasinski, T. Plaszczyk, and M. Zarski, “Analytical Modelling in Dynamo,” *Transactions of the VŠB – Technical University of Ostrava, Civil Engineering Series*, vol. 18, no. 2, 2018.
- [9] J. Jia, J. Gao, W. Wang, L. Ma, J. Li, and Z. Zhang, “An Automatic Generation Method of Finite Element Model Based on BIM and Ontology,” *Buildings*, vol. 12, no. 11, 11 2022.
- [10] T. Singh, M. Mahmoodian, and S. Wang, “Enhancing Open BIM Interoperability: Automated Generation of a Structural Model from an Architectural Model,” *Buildings*, vol. 14, no. 8, 8 2024.
- [11] I. Rudenko and Y. Petryna, “An Approach to Automatic Building Information Modeling-Based Generation of Finite Element Models of Different Complexity and Finite Element Dimensionality †,” *Buildings*, vol. 15, no. 2, 1 2025.
- [12] D. Wang and H. Lu, “Development of a BIM Platform for the Design of Single-Story Steel Structure Factories,” *Buildings*, vol. 14, no. 3, 3 2024.
- [13] Z. Wang, C. Du, Q. Liu, and A. Borrmann, “BIMdiff: Object-Level BIM Change Detection with Identifier Reconciliation,” Tech. Rep. [Online]. Available: <https://github.com/ZijianWang-ZW/IFC2StructuredData>

- [14] B. Habte and E. Guyo, "APPLICATION of BIM for STRUCTURAL ENGINEERING: A CASE STUDY USING REVIT and CUSTOMARY STRUCTURAL ANALYSIS and DESIGN SOFTWARE," *Journal of Information Technology in Construction*, vol. 26, pp. 1009–1022, 2021.
- [15] Computers and Structures, Inc., *CSiXRevit 2020: SAP2000, ETABS, SAFE and Revit 2020 Data Exchange Documentation*, https://www.csiamerica.com/sites/default/files/CSiXRevit_2020_Manual.pdf, Computers and Structures, Inc., Walnut Creek, CA, 2019.
- [16] J. Wu, H. L. Sadraddin, R. Ren, J. Zhang, and X. Shao, "Invariant Signatures of Architecture, Engineering, and Construction Objects to Support BIM Interoperability between Architectural Design and Structural Analysis," *Journal of Construction Engineering and Management*, vol. 147, no. 1, 1 2021.
- [17] C.-L. Hsu, J.-T. Wang, and Y.-H. Huang, "A Functional Element-Naming Approach for BIM Elements in Building Envelope Systems," *Journal of Construction Engineering and Management*, vol. 149, no. 6, 6 2023.
- [18] J.-R. Lin and Y.-C. Zhou, "Semantic classification and hash code accelerated detection of design changes in BIM models," *Automation in Construction*, vol. 115, p. 103212, 7 2020.
- [19] Ministero delle Infrastrutture e dei Trasporti, "Aggiornamento delle norme tecniche per le costruzioni (NTC 2018), D.M. 17 gennaio 2018," *Gazzetta Ufficiale della Repubblica Italiana*, n. 42, Suppl. Ordinario n. 8, 2018.