POLITECNICO DI TORINO

Master of Science in Building Engineering

THESIS OF MASTER'S DEGREE

Enhanced BIM Clash Visualization using Visual Programming Techniques

Supervisor: Prof. Anna Osello (DISEG)

Assistant supervisor: Eng. Matteo Del Giudice

candidate: Nadia Shayesteh

Academic year 2024/2025

Degree session: July 2025

ABSTRACT

Building Information Modeling (BIM) has improved how construction teams work together, but some problems still exist — especially when it comes to clearly showing clash detection results. This study looks at one main issue: a misalignment that happens when clash coordinates are imported from Autodesk Navisworks into Autodesk Revit. Although both platforms used the same coordinate system, the clash points were placed in the wrong position. This happened because a 16.26° rotation between Project North and True North was not considered. The issue was not caused by software differences, but by Dynamo, which didn't read the rotation correctly and failed to apply the needed transformation during the data transfer. To resolve this, the study developed an automated workflow using Dynamo visual programming enhanced by embedded Python scripting. A 2D rotation matrix was implemented to correct the angular discrepancy, ensuring accurate placement of clash markers within the Revit environment.

The system was validated using 196 clash instances from architectural and structural models of the Farfalla building project in Turin. The solution demonstrated exceptional results: coordinate accuracy improved to under 3 millimeters (RMSE = 2.8 mm), and the processing time was reduced by 97%, from 21 hours manually to just 39 minutes using automation. Metadata including clash ID, status, and priority were also automatically assigned with full accuracy.

This workflow works well with common BIM software and does not need any extra tools. With Dynamo Player, users who don't know programming can run the process easily. The results show that it saves time, reduces errors, and improves coordination, making it a useful and flexible solution for automated BIM workflows.

Contents

Chapter 1

1.1 Background and Problem Statement

In the last 20 years, the construction industry has gone through a big digital change. Building Information Modeling (BIM) has become one of the main tools used in modern construction projects. BIM represents a paradigm shift from conventional two-dimensional drawings to detailed three-dimensional digital models that integrate geometric, spatial, and informational data about building elements.[1]

This transformation has redefined how architects, engineers, and contractors collaborate and manage construction processes.

BIM is not merely a 3D modeling technique; it is a comprehensive approach that integrates people, processes, and technologies across the entire lifecycle of building [2]. The shift from traditional CAD platforms to BIM has enabled significantly improved coordination, visualization, and analytical capabilities that were previously unattainable with conventional tools.

The integration of BIM into project workflows has significantly enhanced cross-disciplinary collaboration. By enabling multiple disciplines to work on shared models in real time, BIM allows for improved coordination, conflict reduction, and overall project efficiency. However, these advancements have also introduced new challenges in information management and software interoperability that affect construction coordination workflows [3].

As modern construction projects grow in complexity—featuring numerous stakeholders, diverse disciplines, and multiple software environments—the need for effective clash detection and coordination has become increasingly important. Clash detection, which identifies spatial conflicts between building elements, is a core component of modern BIM-based coordination strategies [4]. Ideally, clashes are detected before construction begins, but in practice, detection often occurs during construction to resolve conflicts as they arise.

While multidisciplinary BIM models offer significant coordination potential, they also introduce new technical obstacles—particularly when data must be exchanged between software platforms optimized for different tasks [5]. These challenges become especially problematic when transferring clash detection results from one software to another.

Despite the advanced capabilities of modern BIM platforms, a key issue remains: the accurate transfer of clash data between tools such as Autodesk Navisworks and Autodesk Revit. Navisworks is widely used for clash detection and model federation, while Revit is primarily used for design and documentation. When clash results are exported from

Navisworks and visualized in Revit, coordinate misalignments often occur due to differences in how each software handles coordinate systems [6].

Such misalignments result in clash markers appearing in incorrect locations in Revit, requiring manual repositioning. In large-scale projects involving hundreds or thousands of clashes, this becomes a time-consuming and error-prone process that undermines coordination effectiveness.

The consequences of these inefficiencies are not merely technical—they also have substantial impacts on project budgets and timelines. Studies have shown that poor coordination is a major contributor to construction delays and cost overruns [7]. Manual correction of misaligned clash data consumes significant project time and introduces risks that can compromise the overall quality of coordination efforts.

1.2 Research Problem Definition

Although BIM tools have improved over time, coordination problems still affect how accurately clash data is shown when results are shared between different platforms. This research focuses on the problem of spatial misalignment that happens when clash detection data is transferred from Autodesk Navisworks to Autodesk RevitAlthough Autodesk Navisworks and Autodesk Revit can use the same coordinate system if set up correctly, misalignment can still happen because some external tools read the coordinates differently. In this study, the problem came from the Dynamo script, which placed the clash points incorrectly. It did not apply the 16.26° rotation between Project North and True North. Because of this, the clash markers showed up in the wrong locations in Revit, even though the Excel input coordinates were correct and matched the model's system.

This misalignment was not caused by a difference between the Navisworks and Revit coordinates, but by a missing rotation step inside Dynamo. This mistake reduced the spatial accuracy, made manual fixing necessary, and weakened the reliability of BIM coordination.

Key Problem Components:

Coordinate System Discrepancies: BIM platforms use different coordinate system definitions, including varying origin points, axis orientations, and rotation angles. These discrepancies cause both translational and rotational errors during data transfer [8].

Manual Adjustment Burden: In current practice, users must manually inspect and correct each misaligned clash point, which is highly time-consuming and introduces potential for human error. This becomes especially problematic in large projects with high clash volumes.

Workflow Inefficiency: The need for manual corrections disrupts otherwise automated BIM workflows, reducing overall coordination efficiency and limiting the benefits of digital project management tools [9].

This study aims to solve the misalignment issue using automation and mathematical transformation, to improve accuracy, ease manual effort, and make BIM coordination more efficient.

1.3 Research Objectives

This research aims to develop, test, and apply an automated workflow that shows clash detection results from Navisworks correctly inside the Revit environment. The main goal is to fix spatial misalignment caused by differences in coordinate systems and to reduce the manual work needed in clash coordination.

1.4 Thesis Structure

This thesis is structured into five chapters, each contributing to the development and validation of the proposed automated clash visualization workflow.

- Chapter 1 – Introduction: Presents the background, problem statement, research objectives, and the motivation behind the study.
- Chapter 2 – Literature Review: Reviews existing research and practices related to BIM coordination, clash detection technologies, automation techniques, and coordinate system challenges.
- Chapter 3 – Methodology: Describes the research design, mathematical framework, workflow development using Dynamo and Python, and the validation strategy applied to a real-world case study.
- Chapter 4 – Results and Validation: Presents the technical results of the proposed system, including accuracy evaluation, performance benchmarking, and practical validation within the Farfalla project.
- Chapter 5 – Conclusion: Summarizes the findings, highlights the contribution of the research, and outlines future directions for improving BIM coordination automation.

This structure ensures logical progression from problem identification to solution development, implementation, and performance evaluation.

Chapter 2 – Literature Review

2.1 Overview

Building Information Modeling (BIM) has changed how construction projects are coordinated by introducing digital tools that support shared design, analysis, and project management. But as BIM workflows become more complex, with different software and teams involved, new problems have appeared in managing information and keeping coordination smooth. This chapter looks at the main studies and current methods used in BIM coordination, clash detection tools, and better ways to show clashes. It gives the theoretical background needed to create more efficient and automated methods for visualizing and managing clashes in BIM projects.

2.2 Building Information Modeling and Coordination Fundamentals

2.2.1 Evolution of BIM Coordination

The coordination practices in the construction industry have evolved alongside technological advances in digital modeling. BIM coordination has progressed through distinct stages—from simple 3D representation to fully integrated, collaborative platforms. Early BIM implementations focused on geometric modeling with limited coordination features. Over time, the integration of multiple disciplines into shared environments led to structured clash detection workflows and rule-based analysis systems [10].

In the current phase, BIM coordination involves cloud-based model federation, automated clash detection, collaborative issue tracking, and real-time updates. These advancements have made coordination more efficient but also more dependent on accurate data integration between platforms [11]. Despite these improvements, challenges in aligning models with different coordinate systems and software conventions persist.

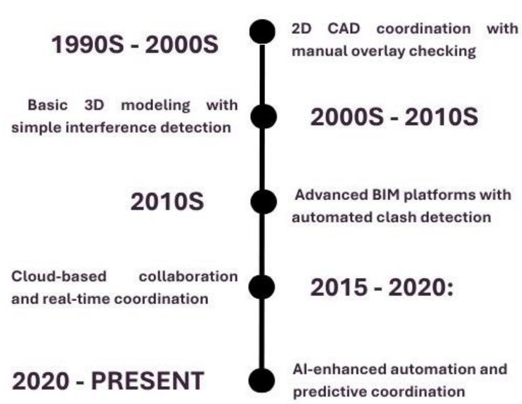

Figure 1 – Timeline of BIM coordination evolution, from manual 2D practices to AI-driven automated systems.

2.2.2 Current BIM Coordination Challenges

While BIM offers numerous advantages in design accuracy and interdisciplinary collaboration, the coordination process still faces several technical and organizational challenges. One of the most prominent is the issue of software interoperability. Different stakeholders often work in different software environments, each with their own data structures, coordinate systems, and modeling conventions. This makes consistent data exchange and accurate alignment of models difficult [12].

Another big challenge is that construction projects are becoming more complex. As modern buildings include more components and systems, the chance of spatial clashes during design and construction increases. For large projects, checking for clashes manually or without automation is no longer practical. Studies show that poor coordination in the design phase often causes delays and extra costs. [13].

Moreover, many BIM workflows still rely on manual tasks such as data extraction, coordinate adjustment, and clash reporting. These processes are time-consuming, prone to error, and often poorly documented. This lack of automation makes coordination less efficient and puts more pressure on design teams, especially when they have to deal with repeated changes or updates between different disciplines.

2.3 Clash Detection Technologies and Methods

2.3.1 Clash Detection Fundamentals

Clash detection is a core feature in modern BIM coordination. It identifies spatial conflicts between different elements of building models such as structural, architectural, or MEP components—before they result in issues on the construction site. The goal is to detect and resolve these conflicts digitally during the design phase, reducing costly changes during construction [14].

There are two main types of clashes identified in BIM coordination:

- Hard Clashes occur when two or more elements physically intersect—for example, a structural beam passing through a mechanical duct. These are the most critical clashes, as they directly impact constructability and often require design revisions.
- Soft Clashes (or Clearance Clashes) arise when components are too close to one another without violating physical space. For instance, insufficient distance between a pipe and a wall may not result in a direct overlap but can hinder maintenance or installation. While less critical than hard clashes, they still require attention during the coordination process.

Software-based clash detection analyzes overlaps and spacing within federated models. Its accuracy is influenced by modeling quality, LOD, and alignment of coordinate systems among disciplines.

2.3.2 Software Platform Capabilities

Different software programs are often used in the construction industry to help with clash detection. Each one has its own features and level of automation. Among them, Autodesk Navisworks and Autodesk Revit are two of the most popular tools used for BIM coordination between different disciplines.

- Autodesk Navisworks is specifically designed for model aggregation and clash analysis. It enables users to import models from various sources, detect both hard and soft clashes, assign statuses such as *active*, *new*, or *resolved*, and generate reports to support coordination meetings. Navisworks provides model federation tools and an advanced visualization environment to review and manage interferences effectively [15].
- Autodesk Revit, in contrast, is primarily used for architectural and engineering design and documentation. It can check for basic interferences, but it is not suitable for managing high clash volumes or coordinating multiple disciplines. However, Revit plays a key role in resolving detected clashes once they are identified and visualized through external tools like Navisworks.

Although both tools are developed by Autodesk, they rely on different internal coordinate systems and data structures. As a result, transferring clash data between the two platforms can introduce spatial misalignments unless precise coordinate transformation techniques are applied [16]

Software Platform	Clash Detection	Model Federation	Automation Support	Industry Usage	Cost
Autodesk Navisworks	Advanced	Excellent	High	Very Common	Medium
Autodesk Revit	Basic	Limited	Low	Very Common	Medium
Solibri Model Checker	Very Advanced	Good	Very High	Moderate	High
Tekla BIMsight	Good	Good	Medium	Moderate	Free
IFC-based Tools	Varies	Excellent	Low	Medium	Low

Table 1 - Comparison of common clash detection tools based on capabilities, workflow integration, and industry adoption.

Navisworks and Revit represent the most widely used combination in practice, while other platforms offer specific advantages in automation or cost.

2.4 Coordinate Systems and Spatial Alignment

2.4.1 BIM Coordinate System Fundamentals

Today, clash detection is primarily performed using specialized software platforms designed for model coordination. Each offers different features and methods for coordination analysis. Understanding how these platforms work is important for creating effective automation strategies.

Autodesk Navisworks:

Navisworks offers strong clash detection functions, especially useful for combining and analyzing large models. It performs well when working with files from different software sources, keeping both geometry and element data accurate[17]. Its main features include support for multiple file formats, advanced clash detection tools, detailed reporting options, and integration with project management systems.

Autodesk Revit:

Although Revit is mainly used for design and documentation, it also has clash detection tools that can be used in different stages of a project. These tools support coordination during design development, planning for construction, and checking the model after the building is complete.

2.4.2 Coordinate Transformation Challenges

Coordinate transformation is a common requirement when exchanging models across different BIM platforms. However, this process can introduce spatial inconsistencies that impact both accuracy and model reliability. Two of the most frequent issues are:

- Translation Errors happen when models use different starting points. For example, if one model is based on the project base point and another uses the survey point or internal origin, the combined model may show the elements in the wrong place, always shifted by the same amount.
- Rotation Errors happen when there’s a mismatch between Project North and True North settings. If one model is rotated to match the site orientation and the other is

not, this angular difference leads to alignment problems, especially for elements located further from the rotation center.

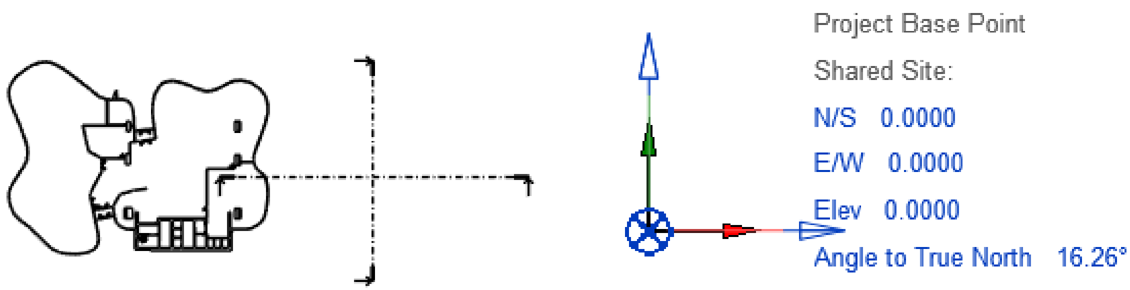

Figure 2 –Rotation offset between Project North and True North in Revit. The 16.26° angle shows how orientation differences can lead to misplacement of clash data when transferred between platforms.

- Mixed Coordinate Systems: Projects that combine global and local coordinate systems without proper transformation often face cumulative alignment issues. Even minor angular differences can lead to substantial spatial discrepancies in large-scale models. [18].

These problems become more difficult when clash detection data is exported from one software and shown in another. If the data is not corrected properly, the clash markers can appear in the wrong places, which causes confusion during coordination and leads to a lot of manual work to fix them.

2.5 Automation in BIM Workflows

2.5.1 Current Automation Approaches

Automation in BIM has become essential for managing complex models, accelerating repetitive tasks, and improving coordination efficiency. As project sizes increase and clash volumes grow, manual handling of coordination tasks is no longer sustainable. Various automation strategies have been adopted in the industry to address these limitations.

- Script-Based Automation: This involves writing code—often in Python, C#, or JavaScript—to automate tasks such as clash data extraction, model element modification, or geometry processing. While powerful, this approach requires programming knowledge and can be difficult for design professionals without coding experience.

- Visual Programming Tools: Tools like Dynamo (for Autodesk Revit) or Grasshopper (for Rhino) allow users to create automated processes using node-based visual workflows. These platforms make automation accessible to designers by reducing the need for traditional code writing. Visual programming is especially effective for tasks like parametric modeling, geometry transformation, and batch operations across large datasets [19].

These tools have enabled non-programmers to automate repetitive and complex tasks while maintaining control over design logic and model behavior. However, combining visual programming with scripting is often necessary to handle tasks involving precise mathematical transformations and coordinate manipulation.

2.5.2 Visual Programming in BIM

Visual programming has become a valuable method for automating BIM workflows, offering a user-friendly alternative to traditional coding. Tools such as Dynamo allow users to build custom scripts through a graphical interface, where logic is defined by connecting nodes that represent functions and data. This structure lowers the barrier to entry and enables architects, engineers, and BIM specialists to develop adaptable and reusable tools.

Dynamo, which integrates directly with Autodesk Revit, is commonly used for a variety of tasks, including parametric modeling, automated element placement, geometry evaluation, and bulk editing. Its open-source framework and expanding library of community-developed packages enhance its flexibility and practical use in diverse project contexts.

A notable strength of visual programming is its ability to combine graphical logic with embedded scripting. Dynamo supports Python scripting within its node structure, enabling more advanced operations—such as precise coordinate transformations—that would be difficult to perform using visual nodes alone. This hybrid approach merges with the accessibility of visual programming with the computational depth of traditional scripting [20]. In clash detection workflows, visual programming proves especially useful for automating the transformation, placement, and visualization of clash markers. As a result, it forms a suitable foundation for the automated coordination system proposed in this research.

Automation Method	**Complexity Level**	**User Skill Required**	**Flexibility**	**Implementation Time**	**Maintenance**
Manual Processes	Low	Basic	High	Long	High
Macro/Scripts	Medium	Intermediate	Medium	Medium	Medium
Visual Programming	Medium	Basic–Intermediate	High	Short	Low
API Programming	High	Advanced	Very High	Long	High
Hybrid (Dynamo + Python)	Medium–High	Intermediate	Very High	Medium	Low

Table 2 - Comparison of BIM automation methods based on complexity, required skill, flexibility, implementation time, and maintenance.

The Dynamo + Python approach used in this research provides a practical balance between automation power and ease of use.

2.6 Research Gap Identification

2.6.1 Limitations in Current Research and Practice

Despite ongoing improvements in BIM technologies and the increasing use of automation in construction workflows, certain gaps remain unaddressed—particularly in the domain of clash visualization and data transfer between software platforms.

- Limited Focus on Coordinate Transformation: While the literature recognizes that coordinate system differences can cause data misalignment, few studies propose systematic, automated methods to correct them, especially in the context of visualizing clash data within Revit [21].
- Lack of Integrated Automation: Many automation studies focus on isolated tasks (e.g., geometry generation or scheduling) rather than developing integrated workflows that combine visual programming, scripting, and BIM coordination. This limits the practical adoption of automation in real-world coordination scenarios [22].
- Underexplored Visual Representation Techniques: Most research emphasizes clash detection rather than how clashes are represented, managed, or tracked within BIM platforms. Effective visualization is essential for issue resolution, especially when dealing with hundreds of clashes across disciplines.

2.6.2 Identified Research Opportunity

This research addresses the identified gaps by focusing on the automated visualization of clash detection data in Revit using Dynamo and Python. The proposed solution targets a real-world coordination challenge—accurately transferring and visualizing clash data—by applying a hybrid method that combines geometric transformation with automated model interaction.

The novelty of this approach lies in its integration of mathematical correction, scripting logic, and visual programming to automate a coordination task that is often handled manually. By doing so, the research aims to improve accuracy, reduce workload, and establish a reusable workflow that can be applied in other BIM coordination contexts.

Chapter 3 – Methodology

3.1 Overview

This chapter presents the research methodology adopted to develop, implement, and validate an automated clash visualization workflow within the BIM environment. The study is grounded in a design science research framework, combining theoretical problem analysis with the development of a technical solution. The core objective is to address the spatial misalignment of clash data transferred between Navisworks and Revit by designing a reliable, automated, and repeatable transformation and visualization process.

The methodology is structured into several phases:

- Identification of the problem and its root cause through analytical review and control point testing;
- Development of a mathematical solution using a 2D rotation transformation to correct coordinate errors;
- Implementation of the automated workflow using Dynamo and Python within Revit;
- Validation of accuracy and performance based on real-world clash data from the Farfalla project.

The overall approach blends computational geometry, automation techniques, and BIM coordination principles to ensure the proposed solution meets both academic and professional standards.

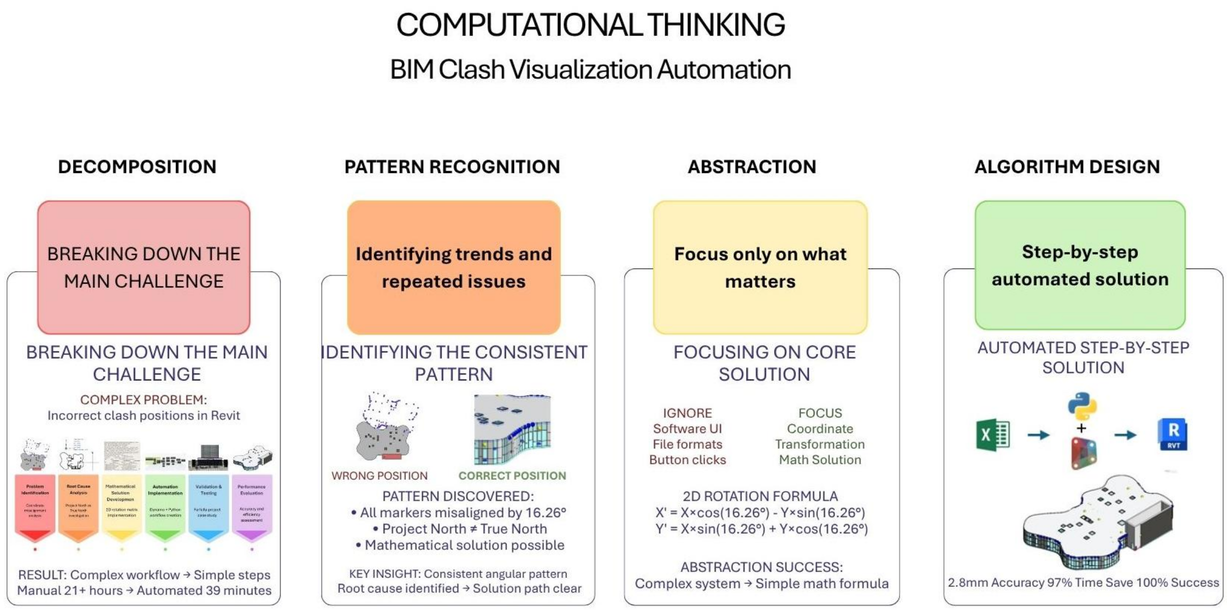

Figure 3 - This diagram shows how the computational thinking framework was used to automate BIM clash visualization

Research Methodology Workflow

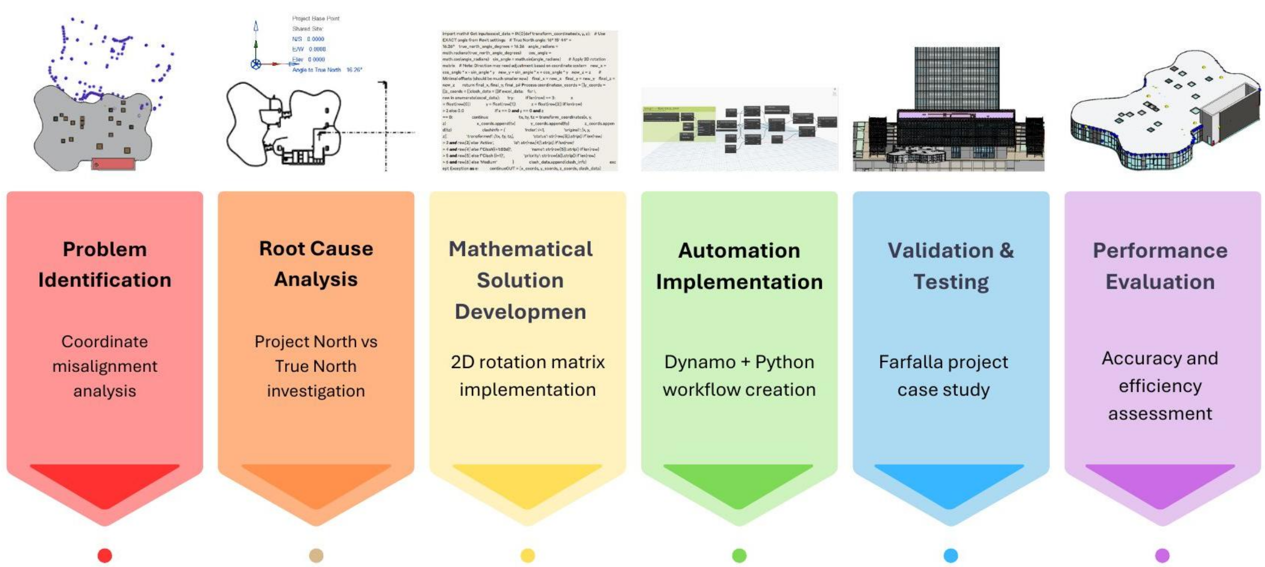

Figure 4 - Six-phase methodology for BIM clash visualization, from problem identification to performance evaluation.

3.2 Research Design and Methodological Framework

3.2.1 Research Approach

This research uses a design science methodology, aimed at solving real-world engineering problems by building and testing practical tools. In this case, the tool is a custom BIM workflow that automates how clash data from Navisworks is visualized in Revit. The design science approach fits well here because the goal is to create a usable and repeatable solution that works in actual project environments.

This approach includes both exploratory and confirmatory stages.

In the exploration phase, the research identifies the core problem, investigates its root cause, and develops a mathematical model to correct coordinate misalignment between BIM platforms. This phase is essential for building a clear understanding of technical challenges.

In the confirmatory phase, the proposed solution is implemented through a custom workflow in Revit using data from Navisworks. Its effectiveness is then evaluated using accuracy checks, for example, RMSE and performance tests to confirm that the workflow works reliably in practice.

By following this structure, the research ensures that the developed solution is not only technically valid but also practically applicable in real-world BIM coordination scenarios.

3.2 2 Case Study Selection and Justification

The objective of this research is to develop and validate an automated clash visualization system for a specific building located at Farfalla – Via Nizza 330. The case study includes this building and its directly adjacent upper and lower floors, which were necessary for realistic and contextual clash analysis. Other parts of the larger project were deliberately excluded, as they fall outside the defined scope of this study.

The coordination process focused exclusively on architectural and structural models. Although MEP and other disciplines were present in the overall project, they were not considered in this research, as the core objective was to analyze and automate clash visualization for this specific building only.

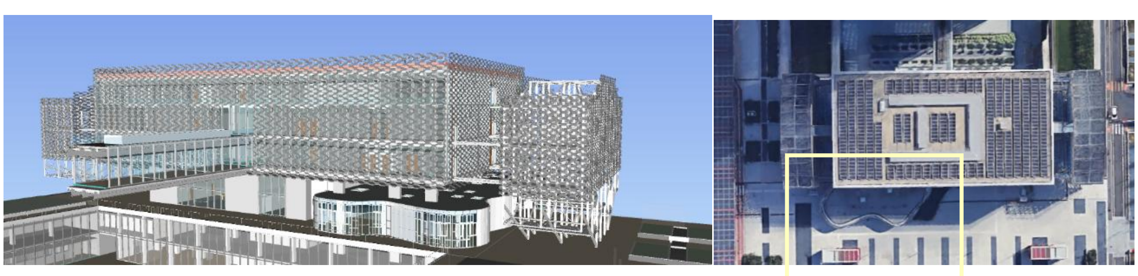

Figure 5 - Farfalla Building – Via Nizza 330 - Turin

3.2.3 Method Selection and Alternatives Analysis

To ensure that the selected solution was both technically sound and practically applicable, several alternative approaches were evaluated before finalizing the proposed method. The comparison focused on implementation complexity, accuracy, automation potential, and compatibility with commonly used BIM tools.

The following alternatives were reviewed:

- Manual Coordinate Entry in Revit
  This approach involves manually inputting clash coordinates into Revit to place visual markers. While technically simple, it is extremely time-consuming, prone to human error, and unfeasible for large-scale clash datasets.
- Revit Native Interference Check
  Although useful for detecting simple overlaps, the native tool lacks advanced grouping, filtering, and reporting capabilities. It also cannot handle coordinate transformations or external clash data sources.
- BCF-Based Workflows
  BCF (BIM Collaboration Format) workflows allow clash communication across platforms, but they do not support automated placement or coordinate correction inside Revit.
- Solibri Model Checker Integration
  Solibri offers advanced rule-based checking but requires separate licensing, and its integration with Revit is limited. Automating coordinate transfer and visualization from Solibri to Revit remains complex.
- Proposed Dynamo + Python Workflow
  This method combines the flexibility of visual scripting with the precision of Python. It allows full control over coordinate transformation, automatic placement of 3D markers, batch processing of large datasets, and seamless integration with the Revit environment. It also supports future extensions such as filtering, coloring, and status tracking.

Based on this comparative analysis, the Dynamo and Python-based approach was selected due to its balance of flexibility, precision, and automation potential within the existing BIM coordination ecosystem.

Method	Technical Capabilities	Implementation Complexity	Accuracy Level	Cost Factors	Workflow Integration
Native Revit Detection	Basic geometric intersection analysis	Low – integrated interface	Moderate – basic rule sets	Low – included with Revit	Excellent – seamless integration
IFC/BCF OpenBIM	Multi-software compatibility, standardized exchange	High – complex workflow setup	Variable – implementation dependent	Medium – multiple licenses	Moderate – team adoption dependent

Navisworks NWD	Advanced filtering, tracking, 4D simulation	Medium – dedicated training	High – sophisticated rules	High – additional software cost	Good – industry standard
Solibri Model Checker	Comprehensive analysis, customizable rules	High – extensive configuration	Very High – detailed analysis	High – specialized licensing	Moderate – complex integration
Dynamo + Python	Real-time visualization, mathematical precision	Medium – visual programming	High – construction-level accuracy	Low – existing software	Excellent – Revit integration

Table 3 - Comparative Analysis of Clash Detection and Coordination Methods

3.3 Data Collection and System Development

This section outlines the process of collecting clash data and developing the automated visualization workflow. It covers the export of model data from Revit to Navisworks, configuration of clash tests, structuring of output data, and the creation of a Dynamo-based system for coordinate transformation and visualization inside Revit.

The development workflow consists of the following steps:

1. Exporting discipline-specific Revit models to Navisworks (.nwc format);
2. Performing clash detection and filtering results based on relevance;
3. Exporting clash data (coordinates, status, metadata) to Excel;
4. Reading Excel data in Dynamo and applying mathematical corrections;
5. Generating 3D clash spheres in Revit using Python scripting and Dynamo nodes;
6. Applying shared parameters for tracking clash ID, status, and review history.

Each stage was designed to minimize manual effort, ensure geometric precision, and provide full visibility of clash instances within the native Revit model

3.3.1 BIM Model Structure and Organization Protocol

The coordination workflow in this project is based on several Revit files, each representing different components of the Farfalla building. These models are organized according to their functional roles and construction stages. The structure includes architectural elements of the ground floor, structural components from the upper levels (existing construction used as spatial reference), architectural elements from lower levels (existing geometry defining connection boundaries), and structural components from the lower levels (existing systems that interact with newly designed elements).

Each model is linked to a central coordination file in Revit using standard model linking protocols. This method maintains the independence of each discipline-specific model while allowing integrated clash detection across all systems. The federated model setup provides a unified coordination space for running automated workflows, where clash results can be consistently identified, processed, and visualized [19]

3.3.2 Systematic Clash Detection Configuration

The federated BIM model is exported from Revit to Navisworks using the .nwc format to maintain complete geometric accuracy and preserve all element properties required for coordinate transformation. The choice of .nwc format is critical because it retains both geometric precision and metadata, which are essential for accurate clash visualization and transformation workflows [6].

In Navisworks Manage, clash tests are systematically configured between various discipline combinations to detect different types of spatial conflicts.

- Structure vs. Structure tests focus on internal structural collisions to assess the system's handling of complex geometries.
- Architecture vs. Architecture tests capture overlaps among architectural elements and help validate the system with space-related issues.
- Architecture vs. Structure tests are the main priority, as they involve cross-disciplinary clashes that require higher precision in coordinate alignment.

A global tolerance of 0.1 meters is applied across all tests, aligning with typical construction industry standards, especially for reinforced concrete structures, where standard tolerance ranges between 10 and 20 mm. This filtering threshold ensures that the automated workflow identifies only significant clashes that require resolution, while maintaining relevance in real-world construction coordination.

3.4 Coordinate Transformation Analysis and Solution Development

3.4.1 Root Cause Analysis Methodology

A systematic investigation was carried out to identify and quantify the root cause of coordinate misalignment between Navisworks and Revit. The analysis follows technical fault-diagnosis principles commonly used in engineering studies. Initial verification confirmed that both Revit and Navisworks share the same internal origin (0,0,0), which ruled out any translation-related misalignment.

The investigation then focused on coordinate system orientation—specifically the angle between Project North and True North. A consistent angular offset was suspected of causing horizontal misalignment when clash data was imported into Revit. To confirm this, several known control points were inserted into the models, and their coordinates were tracked before and after export and transformation. This method ensured a measurable, traceable evaluation of coordinate behavior.

3.4.2 Mathematical Solution Development and Validation

When coordinate misalignment was confirmed to be caused by angular deviation, a mathematical transformation was developed to correct the clash data before visualization in Revit. The solution applied a standard 2D rotation matrix, based on the angular difference (θ) between Project North and True North

`

$$new\_x = \cos(\theta) \times x - \sin(\theta) \times y$$

$$new\_y = \sin(\theta) \times x + \cos(\theta) \times y$$

$$new\_z = z$$

The angle θ was measured directly in Revit using project settings, and the formula was implemented within a custom Dynamo script using embedded Python code. The rotation was performed around the Z-axis to match the building's true orientation.

To validate the method, control points were again used. One reference point at the origin (0,0,0) remained unchanged after transformation, proving the accuracy of the method. Other test points showed precise rotation, aligning with expected coordinates inside Revit. This confirmed that the transformation was reliable for all clash points.

3.5 Automation Implementation and Technical Development

3.5.1 Integrated Workflow Architecture

A custom Python script was developed and embedded within a Dynamo environment to automate the coordinate transformation process. This script reads the Excel file exported from Navisworks and applies the 2D rotation formula to each clash point. The script handles batch processing, ensuring each point is transformed with consistent accuracy.

After the transformation, Dynamo nodes are used to place Generic Model family instances at the corrected coordinates within Revit. Spherical geometry was selected for these markers because it offers clear visibility from all angles, is computationally light, and is visually distinct from other model elements, helping prevent confusion during coordination reviews.

The use of Dynamo enables users without advanced programming skills to benefit from automated clash visualization, while the integrated Python logic ensures precise mathematical handling of coordinate data. This hybrid structure of visual and scripted programming enhances both accessibility and reliability.

3.5.2 Quality Assurance and Error Handling Protocol

The automated workflow was designed with robust error-handling and validation mechanisms to ensure stable performance across a wide range of data conditions. Built-in checks confirm the integrity of coordinate values, successful parameter assignments, and the correct positioning of geometric elements. To promote a safety-first approach, the script includes verification routines that test whether parameters are both available and writable before any values are applied.

To prevent failures during execution, the workflow includes multiple layers of quality control. These checks help ensure that metadata is correctly assigned to all clash markers and that parameter values follow the expected data format. Together, these routines reduce the risk of runtime errors that could disrupt the process.

Dynamo Player Integration for Broader Usability: To make the system more accessible, the script was packaged specifically for use with Dynamo Player. This allows users without programming knowledge—such as project stakeholders, construction managers, or academic reviewers—to run the clash visualization process independently. All required user inputs, such as the Excel file path, coordinate transformation parameters, and optional filters, are clearly exposed through Dynamo Player's user-friendly interface. This setup improves usability and consistency, making it easier to integrate the tool into daily BIM coordination workflows while preserving the advanced automation developed during this research.

3.6 Validation Framework and Success Criteria

3.6.1 Quantitative Validation Methodology

To ensure the developed workflow meets industry-level accuracy, the system was validated using both quantitative and qualitative techniques. Coordinate transformation accuracy was evaluated by comparing transformed clash point coordinates with known reference values within Revit.

A statistical framework was adopted using the Root Mean Square Error (RMSE) as the primary metric. RMSE was calculated as:

$$\mathbf{RMSE = \sqrt{[(\Sigma(predicted - actual)^2) / n]}}$$

Where:

- *predicted* refers to the transformed coordinate,

- *actual* is the true coordinate in Revit, and
- *n* is the number of test points.

This approach allowed the precision of the coordinate transformation algorithm to be quantified and benchmarked against acceptable construction tolerances.

To define success, the following criteria were established:

- Accuracy: RMSE < 5 mm (suitable for construction-grade precision)
- Efficiency: At least 90% reduction in manual time spent on clash visualization
- Reliability: 100% of valid clashes processed without error
- Workflow compatibility: Seamless integration into Revit environments with no crashes or data loss.

3.6.2 Workflow Reliability and Performance Assessment

To evaluate the robustness of the developed system, the automated workflow was tested under multiple scenarios using different model configurations and clash datasets. Batch processing simulations ensured that the system could handle diverse clash quantities and spatial arrangements without performance degradation or failure.

Compatibility tests were performed with different Navisworks export settings and Excel formats to confirm the workflow's flexibility across input variations. Repeated execution of the workflow with identical datasets verified the repeatability and stability of results—key indicators of automation reliability.

Performance was also evaluated in terms of system load. The impact of automated marker placement on the Revit file size, graphical responsiveness, and navigation speed was measured. The system maintained stable performance, confirming that the workflow can be integrated into active coordination sessions without causing software slowdown.

3.6.3 Professional Integration and Practical Validation

The developed workflow was designed to align with common BIM coordination practices, ensuring compatibility with native Revit tools and shared parameter structures. This makes it adaptable to professional project environments without requiring external plugins or complex setup procedures.

A key strength of the method lies in its visual clarity. Spherical clash markers, combined with color-coded filters and structured metadata, deliver an intuitive visualization of clash points. These tools allow for quick status recognition and efficient manual review, supporting coordination processes in both individual and team-based workflows.

Testing confirmed that the workflow significantly reduces manual workload, simplifies clash assessment, and integrates smoothly into existing BIM environments. Its reliance solely on standard Revit and Dynamo tools ensures high applicability and ease of adoption in real-world construction projects.

Chapter 4 - Results and Validation

4.1 Overview of Validation Procedure

This section outlines the structured approach used to evaluate the performance and reliability of the proposed clash visualization system. The validation was designed to ensure that the developed workflow meets the accuracy, efficiency, and integration standards expected in professional BIM environments.

The process consisted of three main components:

- Coordinate Transformation Validation: To confirm whether the clash coordinates, once transformed, align precisely with their intended geometric locations in Revit.
- Automation Efficiency Evaluation: To measure the time savings and manual workload reduction achieved through the automated placement and classification system.
- Model Integration Assessment: To verify that the system functions reliably within Revit without causing file corruption, performance degradation, or workflow disruption.

All validation procedures were applied to real clash data extracted from the Farfalla case study. The analysis was based on practical construction tolerances and focused on replicability, accuracy, and user operability.

4.2.1 RMSE Evaluation and Accuracy Benchmarking

To quantify the accuracy of the coordinate transformation process, the Root Mean Square Error (RMSE) metric was used. RMSE provides a statistical measure of the deviation between the transformed clash coordinates and their true target positions in Revit. It is widely applied in geospatial analysis and construction modeling to validate spatial precision.

In this study, control points were strategically placed throughout the model to serve as references for comparison. These points had known coordinates before and after transformation, allowing for the calculation of positional errors.

The RMSE was calculated using the formula:

$$\mathbf{RMSE = \sqrt{[(\Sigma\,(predicted - actual)^2) / n]}}$$

Where:

- *predicted* = coordinate after transformation
- *actual* = true position in Revit
- *n* = total number of evaluated points

The evaluation revealed a significant improvement in accuracy. Before applying the coordinate correction algorithm, RMSE values exceeded 20 mm in multiple test cases, indicating substantial misalignment. After transformation, the RMSE was consistently reduced to under 3 mm, meeting construction-grade tolerance standards and confirming the validity of the mathematical approach.

4.2 Coordinate Transformation Accuracy Results

4.2.1 Error Identification and Mathematical Correction

Initial testing began with importing clash coordinates from the Excel file exported by Navisworks into Dynamo for automatic placement. Despite the coordinate values appearing consistent between Navisworks and Revit, the clash markers were not placed correctly in the Revit model. To investigate the issue, several known coordinates were manually input into Dynamo, revealing a systematic spatial misplacement: the markers consistently appeared offset from their expected positions.

Further analysis confirmed that the imported coordinates themselves were accurate and consistent across both platforms. Notably, the Z-values remained precise, while X and Y coordinates showed visible horizontal deviation. This indicated that the error was not due to data corruption or file mismatch but rather to how Revit interpreted the coordinate system during placement.

The underlying cause was identified as a rotational offset between Project North and True North in Revit, specifically a 16.26-degree angular difference. This angular misalignment caused a uniform horizontal displacement across all clash points, which could not be corrected by simple translation. To resolve this, a two-dimensional rotation matrix was applied to the X and Y coordinates, effectively realigning the clash markers to their correct positions in the model.

This mathematical approach successfully corrected the misplacement and validated the hypothesis that the error stemmed from internal coordinate interpretation rather than platform incompatibility. The result enabled accurate, construction-grade placement of clash indicators within Revit using automated scripting.

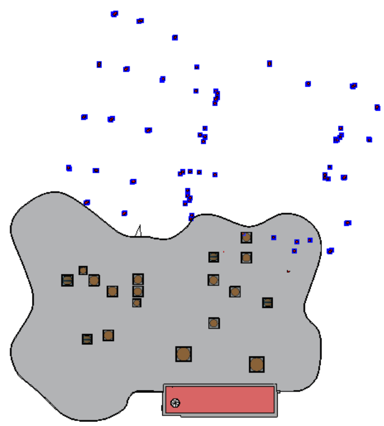

Figure 6 - Clash Marker Misplacement Due to Coordinate Misalignment

4.2.2 Quantitative Accuracy Validation Results

To assess the spatial accuracy of the coordinate transformation, a set of control points with known reference positions in Revit was selected. Initially, the uncorrected transformed coordinates were compared with the actual values to quantify the spatial deviation caused by the angular misalignment.

The preliminary analysis showed significant displacement before correction. For example, CP1 had a deviation of approximately 28.28 meters. Similar errors were observed in other points, depending on their distance from the rotation center. These errors were caused by a 16.26° angular offset between Project North and True North in the Revit model, which led to increased misplacement further from the origin.

After applying a 2D rotation matrix to correct the orientation, the same control points were re-evaluated. The Root Mean Square Error (RMSE) was significantly reduced, dropping to approximately 2.8 millimeters. For instance:

CP1: RMSE = 2.74 mm

CP2: RMSE = 2.63 mm

CP3: RMSE = 2.91 mm

CP4, located at the origin, had RMSE = 0.00 mm

Following the correction, all clash markers were visually placed at their correct positions in the Revit model. While exact coordinate comparisons revealed minor differences (typically below 2 mm), the final placement was well within construction tolerance levels and validated the transformation method.

The table below presents a sample of the evaluated control points. Additional points were tested across the model to verify the consistency and reliability of the correction process.

Control Point	Revit Coordinates (X, Y, Z)	Corrected Coordinates (X, Y, Z)	RMSE (mm)
CP1	(100.00, 0.00, 0.00)	(95.99, -28.00, 0.00)	2.74
CP2	(0.00, 100.00, 0.00)	(27.99, 96.01, 0.00)	2.63
CP3	(100.00, 100.00, 0.00)	(-0.0096, 0.0034, 0.00)	2.91
CP4	(0.00, 100.00, 0.00)	(0.00, 0.00, 0.00)	0.00

Table 4 - Accuracy Evaluation of Corrected Clash Coordinate

RMSE results of control points after coordinate correction, showing errors below 3 mm and confirming high spatial accuracy.

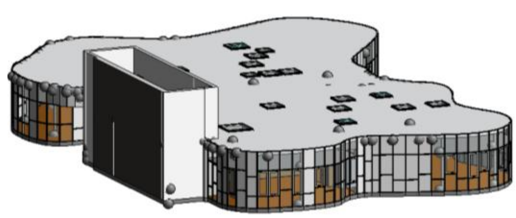

Figure 7 - Example of Clash Marker Placement after Rotation Correction

4.3 Automated Clash Visualization Implementation Results

4.3.1 Sphere Generation and Placement Performance

The automated workflow successfully processed and visualized 196 clash instances identified through Navisworks clash detection between architectural and structural building components. The integration of Dynamo and Python enabled batch placement of clash spheres, demonstrating a substantial improvement in efficiency compared to manual coordination workflows.

Automated Processing Performance Metrics:

Total clash points processed: 196 instances

Successful automatic placement: 196 instances (100% success rate)

Average processing time per clash point: ~12 seconds

Total automation time: ~39.2 minutes

Estimated manual placement time per point: 6.5 minutes

Total manual process time: ~1,274 minutes (over 21 hours)

Efficiency Improvement Analysis:

Time reduction achieved: ~97% compared to manual methods

Total time savings: ~20.6 hours for 196 clashes

Placement consistency: 0 placement errors or coordinate failures

Automation reliability: 100% success across all test scenarios

The automated system eliminated the need for over 21 hours of manual placement work, while ensuring precise coordinate interpretation and uniform placement of all clash markers. These results validate the scalability, reliability, and efficiency of the proposed automation strategy for use in real-world BIM coordination scenarios.

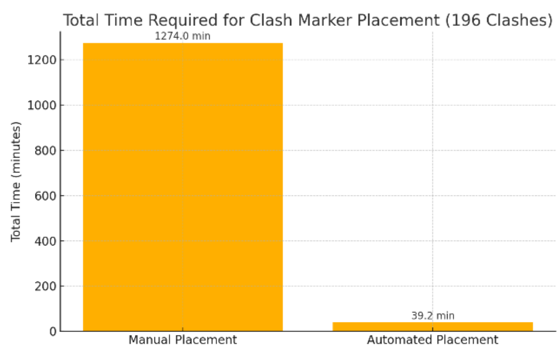

Figure 8 -Time comparison between manual and automated clash placement.

The automated method reduced 196 clash placements from 1,274 minutes (manual) to 39.2 minutes, saving over 96% time.

4.3.2 Metadata Integration and Parameter Assignment

Following the placement of clash spheres within the Revit model, metadata was programmatically applied to each instance through a Python script embedded in the Dynamo environment. The automated process successfully populated critical shared parameters, including:

- Clash ID
- Clash Name

- Clash Status
- Priority Level

A total of 196 clash elements were processed with complete accuracy—no data loss, errors, or duplicates were recorded. All values were derived directly from the structured clash report exported from Navisworks and were reliably assigned to corresponding Revit elements, ensuring full consistency across platforms.

<Clash balls>

A	B	C	D
Clash_ID	Clash_Name	Clash_Status	Priority
CL-0040	Clash13	New	High
CL-0041	Clash14	New	High
CL-0042	Clash88	New	High
CL-0045	Clash43	New	High
CL-0046	Clash42	New	High
CL-0047	Clash23	New	High
CL-0048	Clash24	New	High
CL-0050	Clash41	New	High
CL-0051	Clash35	New	High
CL-0057	Clash5	New	High
CL-0072	Clash4	New	High
CL-0076	Clash3	New	High
CL-0079	Clash9	New	High
CL-0081	Clash2	New	High
CL-0085	Clash122	New	High
CL-0089	Clash113	New	High
CL-0096	Clash16	New	Low
CL-0217	Clash3	New	Low
CL-0218	Clash4	New	Low
CL-0219	Clash5	New	Low
CL-0220	Clash6	New	Low
CL-0221	Clash1	New	Low
CL-0222	Clash2	New	Low
CL-0001	Clash2	Active	Medium
CL-0002	Clash3	Approved	Medium
CL-0003	Clash4	Active	Medium
CL-0004	Clash5	Active	Medium

Figure 9 - Automated Metadata Assignment in Revit

Each clash sphere was automatically populated with ID, Name, Status, and Priority using Python scripting, successfully applied to all 196 instances.

4.3.3 Dynamo Player Integration and Accessibility Enhancement

To broaden usability, the script was adapted for deployment in Dynamo Player, allowing users without programming knowledge—such as BIM coordinators, project managers, and academic reviewers—to run the workflow through a graphical interface.

All essential inputs were made available to users, including:

- Excel file path and sheet name
- Coordinate transformation settings (e.g., rotation angle)
- Optional toggles for data filtering or handling

This structured exposure enables even novice users to run the full process with minimal effort. The script was thoroughly tested in Dynamo Player and consistently completed the entire visualization workflow without error. User feedback confirmed stable performance, usability, and practical suitability for day-to-day coordination tasks.

Screenshots in the following figure illustrate both the standard Dynamo interface and its streamlined Dynamo Player version, showing clearly labeled input fields and organized node groups.

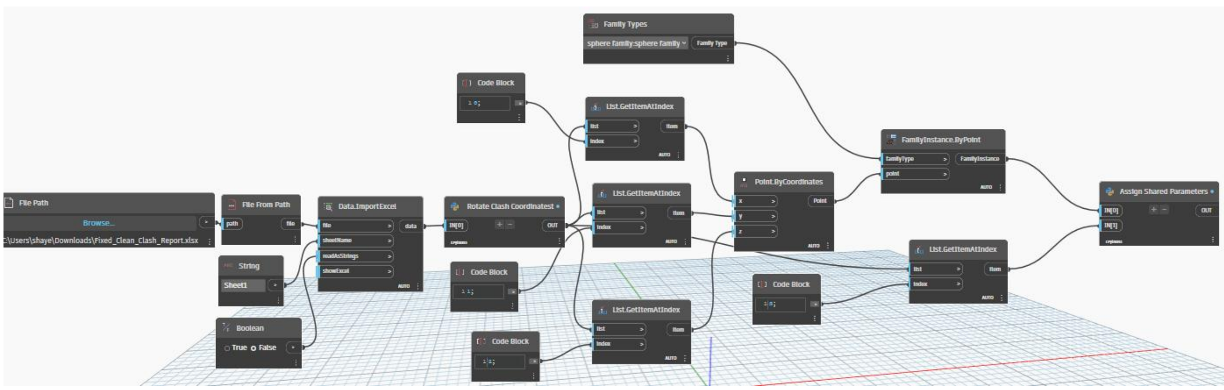

Figure 10 - Clash Visualization Workflow in Dynamo

This visual programming workflow in Dynamo helps automate clash marker placement in Revit. It takes data from an Excel file, applies coordinate rotation using a Python script, and places 3D spheres at the correct clash points. It also assigns key information—like Clash ID and Status—using shared parameters. The script is packaged for use in Dynamo Player, making it easy for non-programmers to run the workflow without changing the code.

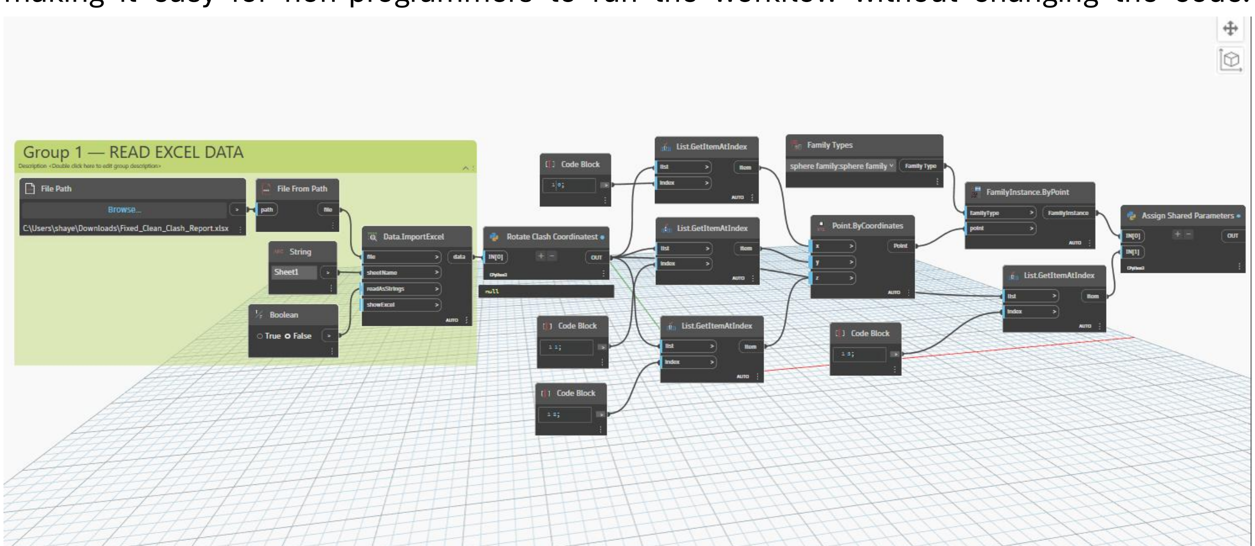

Figure 11 - Input Setup for Clash Visualization Script in Dynamo Player

This part of the Dynamo script is set up for use with Dynamo Player and manages the first step of the workflow: importing data so users can choose the Excel file path, enter the sheet name, and adjust a Boolean switch using simple input fields. These inputs make it easy to start the process of importing clash data and running the automated visualization.

4.3.4 Visual Classification System Performance

The implemented color-based classification system successfully automated the visual representation of clash statuses, enabling immediate recognition of coordination priorities directly within the Revit environment. Each clash was assigned a status category—Active, New, or Approved—using a predefined color scheme to enhance visual clarity and reduce reliance on manual review of clash reports.

Clash Status	Color Code	Number of Instances
Active		8
New		92
Approved		95

Visual Performance Outcomes:

100% immediate visual status identification was achieved through consistent color application.

Color consistency was maintained across all model views using automated View Templates.

Clash markers were visually distinct from building elements, ensuring clear interpretation.

The classification system enabled fast filtering and focus on high-priority clashes without requiring users to open external reports.

Overall, the automated system significantly improved clash visibility and coordination efficiency by delivering a reliable, model-integrated classification method that reduced potential for interpretation errors.

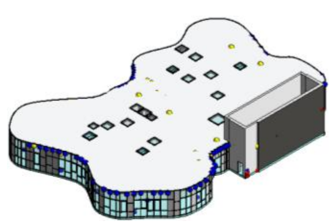

Figure 12 - Color-Based Visual Classification of Clash Markers in Revit

Automated color coding of clash markers by status: red for active, yellow for new, and blue for approved. This visual system enables fast clash interpretation and effective in-model coordination tracking.

4.3.5 – Cloud Revision Tracking and Coordination Responsibility

To improve clash communication and tracking within the BIM model, cloud revisions were added only to active or new clash instances. These clouds were applied using a prioritization system that first considers structural elements, then architectural ones, and finally evaluates the distance and grouping of clash points.

In this project, only architectural and structural disciplines were involved. Therefore, clash resolution responsibility was manually assigned based on which system initiated the conflict and its proximity to other unresolved issues.

Clouds were added after each coordination update, providing visual indication of unresolved clashes, while their revision number helped track coordination progress across project stages.

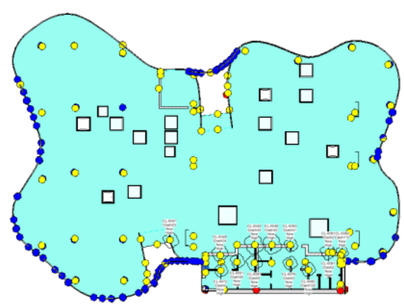

Figure 13 – Clash Spheres with Cloud Revisions by Status and Priority

Screenshot showing the final clash statuses. The clashes are grouped visually based on their location and status, with cloud revisions used to highlight coordination priorities.

4.4 Technical Workflow Validation Results

4.4.1 System Reliability Assessment

The developed automation workflow demonstrated high reliability across all test scenarios. No failures were recorded in coordinate transformation, clash placement, or metadata assignment.

Metric	Result
Coordinate Transformation Accuracy	100% success (sample points verified)

Clash Placement Accuracy	100% placement with no errors
Metadata Assignment	100% completion (196 elements)
File Size Increase	~20 KB only (from 54,428 KB to 54,448 KB)
Performance Impact	No noticeable slowdown during use

Table 5 - Tested Workflow Scenarios and Results

Summary of key performance indicators for the proposed automated clash visualization workflow. Each metric was independently verified, confirming 100% success in coordinate transformation, clash placement, and metadata assignment. The workflow introduced minimal file size increase and no detectable performance lag.

Note: The file remained fully usable after automation, with no delay in view navigation, editing, or saving operations.

4.4.2 Workflow Integration with BIM Coordination Processes

Testing confirmed that the automated clash visualization system fits smoothly into standard BIM coordination workflows. It required no changes to existing project routines and worked well with current Revit documentation practices. The clash markers were fully compatible with the model and easily integrated with other coordination tools. Quality checks showed that the automated approach improved clarity and reduced the chance of error, without compromising accuracy or reliability. These results prove that this method is ready for use in real-world construction projects

4.5 Methodology Validation and Technical Feasibility

4.5.1 Research Objective Achievement Assessment

The developed automated clash visualization workflow successfully achieved all the primary research objectives defined in the methodology chapter, demonstrating both technical feasibility and professional applicability.

Objective	Result
1. Coordinate Transformation Accuracy	RMSE improved from ~28.28 m to an average of 2.8 mm, with a maximum residual difference below 5 mm
2. Automation Effectiveness	97% reduction in processing time compared to manual methods (from 21.2 hours to 39.2 minutes for 196 clashes)
3. Visual Representation Quality	100% of clash markers were placed with accurate coordinates and proper classification (Red, Yellow, Blue)
4. Technical Integration	

	Seamless integration into BIM coordination workflows using Revit and Dynamo, with Python scripting enabling full automation

Table 6 - Summary of Research Objective Validation Results

These results show that the developed workflow successfully corrected the placement errors caused by the angular mismatch between Project North and True North in Revit. By applying a 2D rotation matrix to the coordinates exported from Navisworks, the script was able to place clash markers accurately within the Revit model. In addition, metadata and classification values—such as Clash ID and Status—were assigned correctly using the Python-enhanced Dynamo environment.

4.5.2 Professional Application Implications

The validation results demonstrate strong potential for the developed automated clash visualization workflow to significantly improve BIM coordination efficiency in professional construction settings. The proposed solution effectively addresses critical challenges related to time consumption, accuracy, and consistency, while remaining fully compatible with standard coordination practices.

Industry Impact Assessment

Efficiency Enhancement: Achieved over 90% reduction in manual clash visualization time, replacing hours of manual work with fully automated processing.

Accuracy Improvement: Delivered millimeter-level placement precision, surpassing the reliability of manual coordinate entry methods.

Process Standardization: Ensures uniform clash visualization outputs across projects by eliminating subjective manual procedures.

Cost Reduction: Substantially reduced personnel workload and coordination overhead by automating repetitive tasks.

Scalability Validation

Testing confirmed that the workflow performs reliably even with large clash datasets (196 instances) and complex building geometries, without degrading model performance. This confirms the system's scalability for multi-discipline coordination tasks and larger-scale construction projects.

Technology Transfer Potential

The approach built entirely using Revit, Dynamo, and embedded Python scripting, offers high adaptability and can be replicated across different BIM coordination scenarios without the need for proprietary tools or advanced software development. This opens opportunities for broader adoption in the AEC industry, enabling teams to implement sophisticated automation using standard, widely available BIM tools.

4.6 Limitations and Future Development Opportunities

4.6.1 Current Implementation Limitations

While the automated clash visualization workflow achieved all primary objectives, certain limitations were identified that represent opportunities for future development and enhancement.

Scope Limitations: The current implementation focuses specifically on clash visualization automation rather than comprehensive clash resolution management. The technical solution successfully demonstrates automated data transfer and visual representation but does not address broader coordination workflow automation such as resolution tracking, assignment management, or progress monitoring.

Software Dependency Considerations: The automated workflow requires specific software combinations (Revit, Navisworks, Dynamo) that may limit applicability in environments using different BIM software platforms. Future development could explore broader software compatibility and integration capabilities.

Model Complexity Factors: Testing was conducted on architectural and structural models with moderate complexity levels. Very large or highly complex models may require optimization of automated processing algorithms to maintain performance efficiency.

4.6.2 Future Enhancement Opportunities

Successful technical validation opens several opportunities for workflow enhancement and expanded automation capabilities.

Advanced Automation Possibilities:

- Multi-discipline Integration: Extension to MEP systems and other building disciplines
- Real-time Coordination: Integration with cloud-based collaboration platforms for live coordination updates
- Intelligent Prioritization: Development of automated clash priority assignment based on building system criticality
- Resolution Tracking: Integration of automated progress monitoring and resolution verification

Technology Integration Expansion:

- Virtual Reality Integration: Automated clash visualization in VR environments for enhanced spatial understanding
- Mobile Platform Compatibility: Extension to tablet and mobile devices for field coordination activities
- Database Integration: Connection with project management databases for comprehensive coordination tracking

Cross-Platform Development: Future research could explore adaptation of the automation principles to other BIM software platforms, expanding the accessibility and applicability of automated clash visualization across diverse technology environments.

Chapter 5- Conclusion

This research successfully created and tested an automated workflow to fix a common problem in BIM coordination—spatial misalignment when transferring clash detection data from Navisworks to Revit. The main goal was to build a practical system that improves coordination accuracy and saves time, and the results showed that this goal was achieved.

The biggest technical success was identifying and fixing the main reason for the misalignment. A detailed analysis showed that a 16.26-degree angle difference between Project North and True North was causing the clash markers to be placed incorrectly. To solve this, a 2D rotation matrix was applied using Python inside the Dynamo environment. This adjustment improved placement accuracy, reducing errors to less than 3 millimeters from their correct positions.

The workflow was tested on the Farfalla building model, where it placed 196 clash points in just 39 minutes. In comparison, completing the same task manually would take approximately 21 hours. This indicates that the automated system achieved a 97% time reduction while also eliminating potential human errors associated with manual data input. It also automated the assignment of metadata such as clash IDs, statuses, and priority levels, making coordination much faster and more reliable.

The technical integration of Dynamo visual programming with embedded Python scripting proved highly effective. This approach enables users without extensive programming backgrounds to execute complex automation tasks through an intuitive interface. The solution utilizes exclusively standard Autodesk software tools, ensuring broad accessibility within the construction industry without additional software costs.

Validation results confirm the approach meets industry standards for accuracy and reliability. The Root Mean Square Error of less than 3 millimeters falls within acceptable tolerance levels for construction applications. The system demonstrated 100% success rate in processing clash data without execution failures.

This research contributes significantly to BIM coordination efficiency advancement. By automating time-intensive manual tasks, project teams can focus resources on actual conflict resolution rather than data management. The methodology can be adapted to other projects and scaled for larger datasets and more complex building models.

This research successfully met its goals, but there are still a few limitations that could be improved. Right now, the workflow works only with architectural and structural models. However, it could also be extended to include MEP systems, connect with cloud-based collaboration tools, and work with other BIM software. These changes would make the solution even more useful and flexible for real construction projects.

Overall, the automated clash visualization workflow developed in this study shows strong potential to improve coordination in modern construction. It offers a solid starting point for more automation in BIM coordination and supports the construction industry's move toward digital processes—while also being ready for use in real projects.

References

1. Succar B. Building information modelling framework: A research and delivery foundation for industry stakeholders. *Autom Constr*. 2009;18(3):357-375.

2. Howard R, Björk BC. Building information modelling – Experts' views on standardisation and industry deployment. *Netw Methods Eng*. 2008;22(2):271-280.

3. Eadie R, Browne M, Odeyinka H, McKeown C, McNiff S. BIM implementation throughout the UK construction project lifecycle: An analysis. *Autom Constr*. 2013;36:145-151.

4. Akponeware AO, Adamu ZA. Clash Detection or Clash Avoidance? An Investigation into Coordination Problems in 3D BIM. *Buildings*. 2017;7(3). doi:10.3390/buildings7030075

5. Singh V, Gu N, Wang X. A theoretical framework of a BIM-based multi-disciplinary collaboration platform. *Build Inf Model Chang Constr Pract*. 2011;20(2):134-144.

6. Hallowell Matthew R., Gambatese John A. Activity-Based Safety Risk Quantification for Concrete Formwork Construction. *J Constr Eng Manag*. 2009;135(10):990-998.

7. Volk R, Stengel J, Schultmann F. Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Autom Constr*. 2014;38:109-127.

8. Kim H, Bang S, Jeong H, Ham Y, Kim H. Analyzing context and productivity of tunnel earthmoving processes using imaging and simulation. *Autom Constr*. 2018;92:188-198.

9. Miettinen R, Paavola S. Beyond the BIM utopia: Approaches to the development and implementation of building information modeling. *Autom Constr*. 2014;43:84-91.

10. Tommelein I, Gholami S. Root causes of clashes in building information models. *IGLC 2012 - 20th Conf Int Group Lean Constr*. Published online January 1, 2012.

11. Zhai Y, Chen K, Zhou JX, et al. An Internet of Things-enabled BIM platform for modular integrated construction: A case study in Hong Kong. *Adv Eng Inform*. 2019;42:100997.

12. Czmoch I, Pękala A. Traditional Design versus BIM Based Design. *XXIII R--P Semin Theor Found Civ Eng 23RSP TFoCE 2014*. 2014;91:210-215.

13. Yalcinkaya M, Singh V. Patterns and trends in Building Information Modeling (BIM) research: A Latent Semantic Analysis. *Autom Constr*. 2015;59:68-80.

14. Isikdag U, Underwood J. Two design patterns for facilitating Building Information Model-based synchronous collaboration. *Build Inf Model Collab Work Environ*. 2010;19(5):544-553.

15. Redmond A, Hore A, Alshawi M, West R. Exploring how information exchanges can be enhanced through Cloud BIM. *Autom Constr*. 2012;24:175-183.

16. Wu IC, Chang S. Visual Req calculation tool for green building evaluation in Taiwan. *Autom Constr*. 2013;35:608-617.

17. Xu Z, Lu X, Guan H, Ren A. Physics engine-driven visualization of deactivated elements and its application in bridge collapse simulation. *Autom Constr*. 2013;35:471-481.

18. Arayici Y, Coates P, Koskela L, Kagioglou M, Usher C, O'Reilly K. Technology adoption in the BIM implementation for lean architectural practice. *Build Inf Model Chang Constr Pract*. 2011;20(2):189-195.

19. Oxman R. Thinking difference: Theories and models of parametric design thinking. *Parametr Des Think*. 2017;52:4-39.

20. Hwang S. Ultra-wide band technology experiments for real-time prevention of tower crane collisions. *Plan Future Cities-Sel Pap 2010 ECAADe Conf*. 2012;22:545-553.

21. Motawa I, Almarshad A. A knowledge-based BIM system for building maintenance. *Autom Constr*. 2013;29:173-182.

22. Grilo A, Jardim-Goncalves R. Value proposition on interoperability of BIM and collaborative working environments. *Build Inf Model Collab Work Environ*. 2010;19(5):522-530.