UNIVERZA V LJUBLJANI

FGG

Fakulteta za gradbeništvo in geodezijo

MANASSEH TETTEH

ADVANCING INFRASTRUCTURE BIM THROUGH INTEGRATED SPATIAL DATA AND ENHANCED DATA EXTRACTION

NAPREDEK INFRASTRUKTURNEGA BIM-A Z INTEGRIRANIMI PROSTORSKIMI PODATKI IN IZBOLJŠANIM PRIDOBIVANJEM PODATKOV

BIM A+ European Master in Building Information Modelling

Master thesis No.:

Supervisor:
Assist. Prof. Jernej Tekavec, Ph.D.

President of the Committee
Prof. Goran Turk, Ph.D.

Ljubljana, 2025

ERRATA

Page	Line	Error	Correction

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK:	**004.42:528(043.2)**
Avtor:	**Manasseh Tetteh**
Mentor:	**Assist. Prof. Jernej Tekavec, Ph.D.**
Somentor:	
Naslov:	**Napredek infrastrukturnega BIM-a z integriranimi prostorskimi podatki in izboljšanim pridobivanjem podatkov**
Tip dokumenta:	**Magistrsko delo**
Obseg in oprema:	**57 str., 31 sl.**
Ključne besede:	**BIM, GIS, prostorski podatki, GIS-BIM, ekstrakcija podatkov**

Izvleček:

Stavbe so prostorske komponente, ki obstajajo v kontekstu prostora. Ustvarjanje geografskega konteksta za oblikovalce olajša analizo načrtov in preverjanje skladnosti s predpisi urejanja prostora. Integracija BIM in GIS je koristna za širok spekter prostorskih analiz in validacije načrtov. Obstajajo trije glavni pristopi integracije BIM in GIS, in sicer BIM v GIS, GIS v BIM in tako GIS kot BIM, integrirani v ločeno orodje ali platformo. Ta raziskava se je osredotočila na drugi pristop, ki je GIS v BIM. Ker sta tako BIM kot GIS ustvarjena za različni domeni in se razlikujeta po predstavitvi podatkov, je bila shema IFC obravnavana kot format izmenjave datotek za takšno integracijo. To ni novost; vendar se je večina poskusov integracije s tem pristopom osredotočala predvsem na geometrijo z manjšim upoštevanjem semantike. Za to raziskavo je bila uporabljena študija primera lokacije, razdeljene na tri cone z različnimi geotehničnimi lastnostmi. Po pretvorbi v IFC je bil nastali model povezan s projektom Revit, da bi preizkusili njegovo odzivnost pri povpraševanju po validaciji načrta. Ta raziskava dokazuje prilagodljivost IFC za integracijo BIM in GIS. Poleg tega je bila med transformacijo določena semantika transformiranih podatkov GIS IFC. Topografski objekt, njegovi nabori lastnosti in hierarhični odnosi so bili modelirani v mehanizmu za manipulacijo feature (FME). . Vendar pa nastali model GIS IFC ni primeren za primere uporabe, kot sta analiza prostorskih meja ali poizvedovanje po podatkih znotraj določenih prostorskih omejitev v izvornih okoljih BIM. Ta raziskava preučuje možnost integracije modelov GIS v BIM za analizo v izvornih okoljih BIM.

BIBLIOGRAPHIC– DOKUMENTALISTIC INFORMATION AND ABSTRACT

UDC:	**004.42:528(043.2)**
Author:	**Manasseh Tetteh**
Supervisor:	**Assist. Prof. Jernej Tekavec, Ph.D.**
Cosupervisor:	
Title:	**Advancing Infrastructure BIM through Integrated Spatial Data and Enhanced Data Extraction**
Document type:	**Master Thesis**
Scope and tools:	**57 p., 31 fig.**
Keywords:	**BIM, GIS, Spatial Data, GIS-BIM, Data Extraction**

Abstract:

Buildings are spatial components that exist in a spatial context. Creating the geographical context for designers facilitates design analysis and municipal code compliance checks. Integrating BIM and GIS is beneficial for a wide range of spatial analysis and design validation. There are three main BIM and GIS integration approaches, which are BIM to GIS, GIS - BIM, and both GIS and BIM integrated in a separate tool or platform. This research focused on the second approach, which is GIS - BIM. Since both BIM and GIS are created for different domains and differ in their representation of data, the IFC schema was considered as the file exchange format for such integration. This is not new; however, most integration attempts using this approach have focused primarily on geometry with less consideration for semantics. A case study of a site divided into three zones with distinct geotechnical attributes was used for this research. After the transformation to IFC, the resultant model was linked into a Revit project to test its responsiveness when queried for design validation. This research demonstrates the flexibility of IFC for BIM and GIS integration. Additionally, semantics of the transformed GIS IFC data were established during the transformation. The topographical object, its property sets, and the hierarchical relationships were all modelled in the Feature Manipulation Engine (FME). However, the resultant GIS IFC model is not fit for use cases such as spatial boundary analysis or querying data within certain spatial constraints in native BIM environments. This research examines the possibility GIS - BIM models for analysis in native BIM environments.

ACKNOWLEDGEMENTS

I'm eternally indebted to God for the convictions he places on my heart and the grit He gives me to follow through on them. My fond appreciation to assist. Prof. Jernej Tekavec, Ph.D., for the elaborate discussions that ultimately shaped the framework for this research.

To my family, I am immensely grateful. To my friends from BIM A+, I deeply cherish the moments we created, the all-nighters to finish assignments, the collaboration, and the lofty dreams about what we can achieve with this program. I will carry these moments for a lifetime.

TABLE OF CONTENTS

INDEX OF FIGURES

LIST OF ACCRONYMS AND ABBREVIATIONS / SEZNAM AKRONIMOV IN OKRAJŠAV

AEC	Architecture, Engineering, and Construction
AECO	Architecture, Engineering, Construction and Operations
ASCII	American Standard Code for Information Interchange
BBT	Brenner Baser Tunnel
BEM	Building Energy Modelling
BIM	Building Information Modelling
Bsi	Building Smart International
CRS	Coordinate Reference System
EPSG	European Petroleum Survey Group
ETL	Extract, Transform, Load
FME	Feature Manipulation Engine
GIS	Geographic Information Systems
IAI	International Alliance for Interoperability
IFC	Industry Foundation Classes
LOD	Level of Detail
LOIN	Level of Information Need
OOP	Object Oriented Programming
PCS	Projected Coordinate System
SPF	Step Physical File
STEP	Standard for Exchange of Product Information
SWT	Semantic Web Technology
UTM	Universal Transverse Mercator
WGS	World Geographic System
XML	Extensive Markup Language

1 INTRODUCTION

The Architectural, Engineering, Construction, and Operation (AECO) industry has benefited tremendously from the integration of Spatial data and Building Information Modelling (BIM). The integration of these systems has enhanced the accuracy of analysis and simulations in various aspects, including flood analysis, noise simulations, shadowing, and many other specialized areas. GIS integrated into BIM also provides spatial context that facilitates design decision-making [1]. The benefits of this have been immense, having direct benefits for sustainability, efficiency, and optimization. However, their integration has not been seamless, given the difference in their data structures that target different domains.

Traditionally, three approaches exist for integrating GIS and BIM. The first approach occurs at the application level, converting BIM to GIS. The second approach, on the other hand, converts GIS - BIM. The third approach focuses on integration at the data level, which involves overlaying both BIM and GIS in a separate domain. The first and second approaches focus on integrating BIM and GIS through APIs and plugins. Such platforms are typically limited to geometry with basic attributes and therefore do not allow for the maximization of semantic information sharing, which is the central idea of BIM. The third approach, which occurs at the data level, ensures that both geometry and semantics are transformed, thereby ensuring full fidelity of the models. Integration of BIM and GIS data has predominantly been done through ontologies such as IfcOWL and GeoSPARQL [2]. This research focuses on advancing the capability of the second approach.
Ontologies offer an excellent opportunity for integration; however, recent studies suggest that several factors hinder their practical deployment. These include large data sizes, complexity, scalability, and tooling immaturity [2]. Additionally, [3] pointed out in their study that most proposed solutions to BIM and GIS integration remain at the prototype level. Consequently, researchers recommend a lightweight and practical solution [4].

Converting geospatial data to IFC using a documented schema is a more readily viable pathway to integrate spatial data with BIM workflows and can effectively preserve semantic meaning while keeping data lean. This option readily aligns with the existing industry toolchain and permits incremental adoption of ontologies once the existing challenges and gaps are solved.

This thesis aims to assess the capability of the IFC schema as a format for integrating BIM and GIS while understanding the semantic issues inherent in it. Additionally, the suitability of the IFC schema to support the integration process at a data level is examined.

1.1 Research Questions

- What are the various ways BIM and GIS can be integrated during the design and construction phase?
- What are the existing challenges in adopting the IFC schema as a BIM and GIS exchange format?
- What are the alternative approaches to integrating BIM and GIS that resolve the semantics mismatch?
- What is the suitability of IFC data generated from GIS for BIM integration for data extraction and analytics?

1.2 Research Goals

- Analyse the suitability of the IFC schema as a data exchange format for BIM and GIS integration.
- Implement a case study to integrate spatial data and BIM at the data level using IFC as a file format.
- Test the suitability of geospatial data IFC for data extraction within native BIM software.

2 LITERATURE REVIEW

This section discusses literature on BIM and GIS integration, aiming to highlight the peculiarities of their integration. The IFC Schema, the primary file format being considered for integration between the two systems in this thesis, is reviewed. It summarily discusses the history of IFC. Additionally, pertinent topics that support the course's interest are discussed. The purpose of this chapter is to place the research in its proper historical context.

2.1 BIM Standardisation History & Current Status

The Industry Foundation Classes (IFC) originated in 1994 by buildingSMART, formerly known as the International Alliance for Interoperability (IAI). IAI created the IFC in response to the critical need for interoperability between disciplines in the Architecture, Engineering, and Construction (AEC) industry [5]. The fundamental objective of this initiative was to develop a vendor-neutral, open data model that comprehensively represents a building's information and supports various integrated workflows across diverse software platforms. Since its initial release, IFC has undergone continuous evolution, with the standardization effort expanding from a building-centred focus to encompass infrastructure workflows, coordinate referencing systems, and a broad range of civil infrastructure elements.

The IAI rebranded itself as buidingSMART (bSI) in 2005 to mark its direction to expand its mission beyond IFC development, and subsequent releases of the IFC versions followed until IFC 4 was released in 2013 [5]. IFC introduced significant enhancements to expand the schema's capacity to represent complex building geometries. Additionally, it closely aligned models with emerging BIM standards. However, it was not until the release of the IFC 4.3 in 2023 that the schemas expanded to include horizontal infrastructure [6]. IFC 4.3 marked a significant milestone in the AEC industry, particularly in the civil engineering discipline, as many infrastructure components assumed a standard representation in the IFC schema [5]

IFC's development trajectory over the years depicts its transformation from a small, building-centred data exchange format to a comprehensive semantic modelling framework able to support complex building and geographical information. This benefit has been deeply established in the building industry; however, its capacity to accommodate and integrate different data structures is yet to be fully explored.

2.2 IFC Architecture

2.2.1 Foundational Object-Oriented Principles

The IFC standard represents arguably the most sophisticated applications of object-oriented design principles within the built environment domain. Consequently, its implementation continues to evolve. Understanding the fundamental concepts of object-oriented programming (OOP) within IFC architecture provides helpful insights into both current capabilities and probable evolutionary trends for the standard.

The fundamental concepts of object-oriented programming, encapsulation, inheritance, and polymorphism are extensively implemented throughout the IFC schema. This enables the representation of comprehensive building information while maintaining logical organization, semantic consistency, and system extensibility [7]. Figure 1 shows IFC schema's complex architecture.

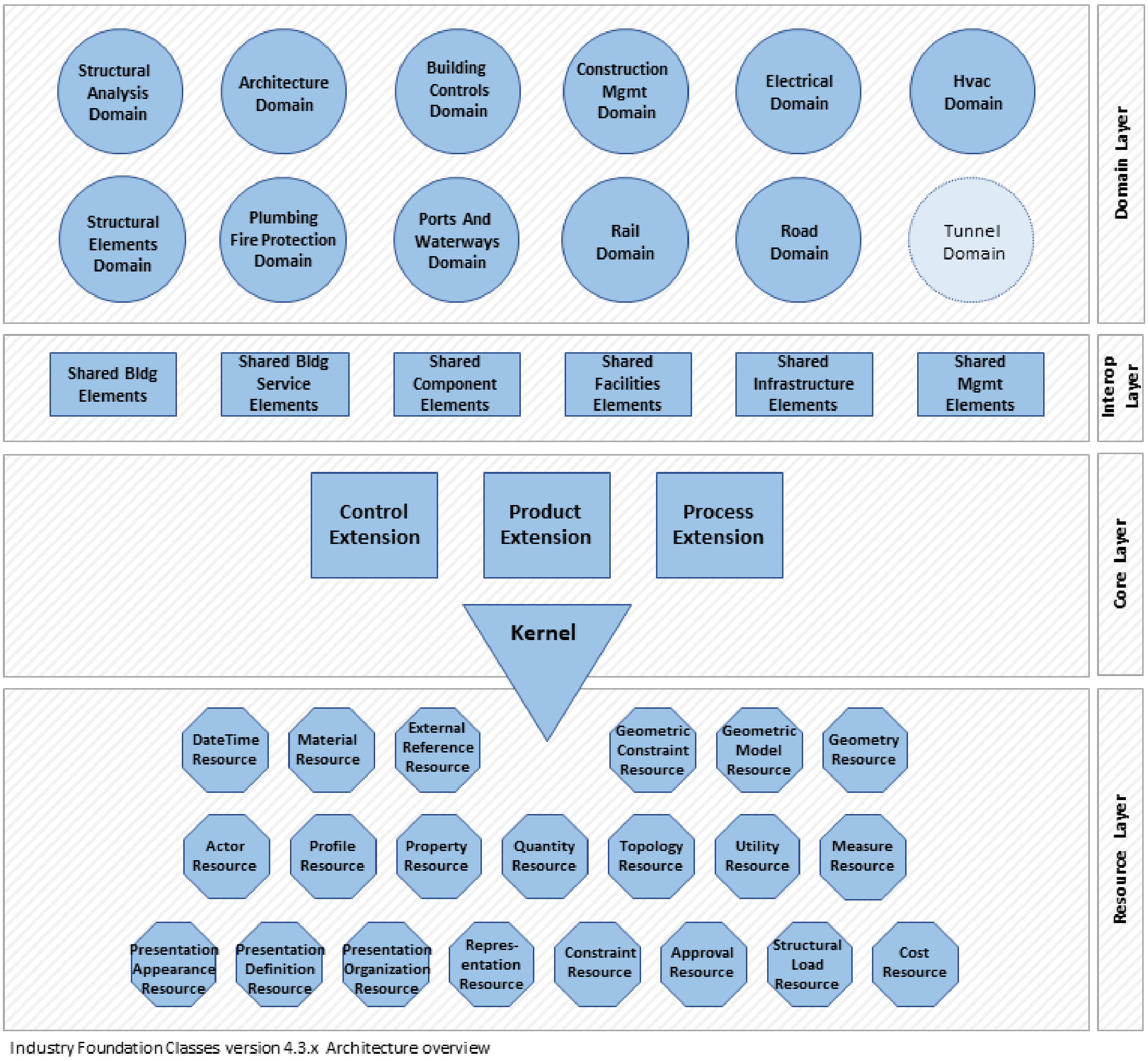


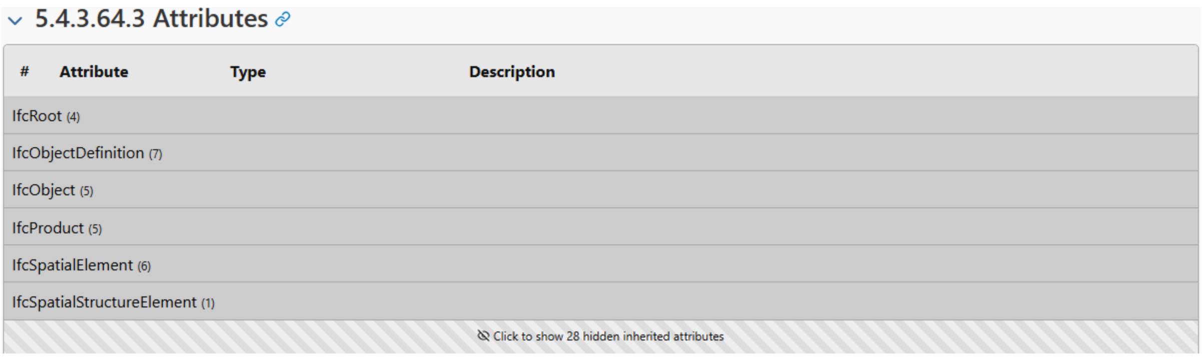
Figure 1 IFC 4.3 Architecture

The constituent parts of IFC can be discussed in many ways. However, for this research, only a selected few that have a direct correlation to the research are discussed. These are discussed briefly in the next bullets.

2.2.2 Core Object-Oriented Components

Classes in IFC serve as templates that define the structure, behaviour, and semantic meaning of elements and their various interrelationships [8]. IFC schema adopts a sophisticated single-inheritance hierarchy that encompasses all entities outside the resource layers. Each hierarchical level is unique, which allows entity classification to be refined progressively. The class-based architecture provides standardized definitions for most disciplines in the AEC industry while maintaining inheritance relationships that promote code reusability, semantic consistency, and logical organization across diverse AEC domains. IFC's hierarchical structure enables systematic categorization of building elements and supports extensibility for domain-specific requirements and emerging construction technologies. It is also important to note that not all classes are object related. Most classes are concepts and abstractions that add relationship or meaning to objects.

Attributes within IFC classes define the complete set of properties and characteristics that belong to an instance of an entity. This representation includes both explicit geometrical data and semantic metadata, which are needed to represent the complete information of a building [9].


5.4.3.64.3 Attributes

#	Attribute	Type	Description
IfcRoot (4)			
IfcObjectDefinition (7)			
IfcObject (5)			
IfcProduct (5)			
IfcSpatialElement (6)			
IfcSpatialStructureElement (1)			

Click to show 28 hidden inherited attributes

Figure 2 IFC Attribute for Space- IFC 4.3

In Figure 2, the attributes of IfcSpace are displayed. The bracketed enumeration indicates inherited attributes embedded in each attribute. This representation denotes sophisticated attribute management strategies, and current researchers have suggested optimized approaches to attribute distribution across class hierarchies that emphasize leveraging property sets for extensible attribute management. This is considered a better approach than hardcoding comprehensive attribute definitions directly into class structures. It enables flexible adaptation to evolving industry requirements while maintaining schema stability and implementation consistency across different software platforms. In addition to the nuanced attribute representation, there are also inherited attributes.

Attribute Inheritance represents a fundamental mechanism through which IFC implements object-oriented inheritance principles. It enables derived classes to automatically acquire properties and behaviours from their parent classes without explicit redefinition [10]. This ensures that entities inherit comprehensive attribute sets from their parent classes. This creates a cascading effect where lower-level entities possess complete attribute collections while maintaining semantic specificity. For example, entities derived from IfcRoot inherit GlobalId, OwnerHistory, Name, and Description attributes by default. This ensures consistent identification and metadata management across the entire schema hierarchy. Attribute inheritance also reduces redundancy and repetition while enabling polymorphic processing at different abstraction levels [11].

Entities are actual instantiations of IFC classes within specific BIM models. This encompasses both geometric representations and semantic information. It also preserves complex relationships with other entities through IFC's sophisticated relationship framework [9]. These entities are fundamental, and their aggregation creates comprehensive representations of built infrastructure. Entities are grouped into three distinct clusters according to whether they represent objects, properties, or relationships. Objects represent semantically described building components. Properties reserve specific parameters that are assigned to objects, and relationships define interactions among objects and properties. This creates a comprehensive semantic network that enables sophisticated querying, analysis, and processing capabilities.

Property Sets, usually represented as Pset, enable detailed and flexible descriptions of building elements within models in IFC. Property sets group related properties and allow for a vast variety of object-specific metadata to be added. For example, IfcSpace has Pset_SpaceCommon that allows for specifying whether the space is located inside or outside a building or site. There's also the Pset_PropertyAgreement, which is beneficial for cadastral purposes. Property sets contain numerous properties that describe characteristics relevant to specific domains or project requirements. Psets are linked to objects through relationships such as IfcRelDefinesByProperties. Property sets enhance interoperability across BIM software platforms by standardizing how extra object information is represented and exchanged. This flexibility supports project-specific customization since users can define tailored property sets to include non-standard data, such as construction phases or sustainability metrics, which are critical for lifecycle management [9]. User-defined property sets are analogous to shared parameters in software like Revit, which can be exported to IFC through schedules or parameter mapping, enabling "clean and tidy" data exchange by including only selected properties [12].

In infrastructure and building projects, property sets enable detailed classification and automation of workflows, including quantity take-offs, compliance checks, and energy performance analysis. Despite certain limitations in representing complex geometry in IFC, property sets remain a robust mechanism to improve the semantic depth of BIM models, providing structured metadata that supports multidisciplinary collaboration and process management in extended project lifecycles.

IfcShapeRepresentation entity is the main geometry representation mechanism within the IFC schema (Figure 3). It facilitates precise geometric data exchange in BIM applications. Various shape representations are determined by the type of component and its geometrical construct. For example, solids that are extruded along a profile, such as walls, have SweptSolid for their shape representation. Boundary Representative (Brep) uses boundary surfaces to define the exterior of a component. There's also CSG or Constructive Solid Geometry, which are 3D shapes created through Boolean operations such as union, intersection, and difference. SurfacModel, FacetedBrep, and Curve2D are other examples of shape representations in IFC.

In addition to shape representations, representation identifiers specify the purpose or type of geometric representation within the IfcShapeRepresentation entity. IFC representation identifiers included Body, Axis, Profile, Footprint, and Surface.

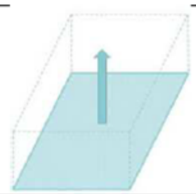
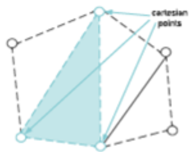
geometry representation	Description (buildingSMART, 2023)	example
SweptAreaSolid (IFC2x3, IFC4)	Swept Area Solid includes several specific types of geometry created by moving a cross-sectional area along a path curve. It allows the representation of various geometry operations such as extrusion, rotation and translation of a circular area.	
Triangulated-FaceSet (IFC4)	Triangulated Face Set is a geometry representation that represents the surface of a three-dimensional object by triangulation. The surface is a collection of triangles described by their vertices.	
ShellBased-SurfaceModel (IFC2x3)	Shell-Based Surface Model is a geometry representation that represents the outer surface of a three-dimensional object using a shell. The shell is a collection of closed faces that together define the surface of the object.	
Geometric-CurveSet (IFC2x3)	Geometric Curve Set is a geometry representation that describes a three-dimensional object using a collection of curves. These curves can be lines, arcs or complex geometric shapes.	

Figure 3 Geometrical shape representation in IFC [13]

The complex hierarchical nature of IFC begins to emerge with the above reading. The nature of IFC, whether it is complex or not, is a debate among researchers; however, it is from this complex hierarchy that the IFC schema derives its ability to represent the complex multi-disciplinary structures in the built industry. To understand the commonalities and differences in geometry representation, the functions of both domains must be understood. Contrary to BIM, GIS uses coordinate-based geometry such as points, lines, and polygons. The combination of these is used to represent various geospatial elements such as roadways, bridges, and streams. Unlike GIS, objects in BIM vary from simple planar objects, such as walls, to parametric roofs with complex geometries. An interesting parallel can be drawn in how both domains represented road surfaces. In GIS, road surfaces are represented by lines or a collection of the same. GIS's focus is to accurately represent the location of the road in the real world, and therefore, lines suffice for this purpose. However, the same road when modelled in BIM will be represented by a triangulated surface, which is a collection of mesh surfaces. This is because visualization and information extraction are key in BIM. The model of the road brings context to BIM models, and extracting volume of the modelled road outputs data that can be useful for analysis and planning.

2.2.3 Performance And Scalability Deficiencies

IFC schema uses the Standard for the Exchange of Product Data (STEP) physical file format. According to Berlo et al., STEP implementations present challenging barriers to modern software development practices. They expressed concerns about the minimal experience contemporary developers possess in Express compared to their extensive expertise in current technologies such as UML, JSON Schema, and XML. This, in their estimation, creates considerable implementation overhead while also limiting the pool of qualified developers capable of efficiently working with IFC systems [11]. STEP is constrained in contemporary software. This is because modern software development paradigms have evolved towards service-oriented architectures, real-time data processing, and cloud-based collaborative platforms [10]. IFC-SPF is in ASCII format, which, while human-readable, is plagued by typical ASCII file issues such as bloated file sizes. The large textual representation characteristic of STEP files results in significantly larger file sizes compared to more efficient serialization formats. This was observed by [10] and further documented cases where equivalent geometric content requires substantially more storage space when represented in IFC-SPF format, as compared with alternative file representation approaches. The sequential reading of files, the unavailability of the mid-file option when extracting, and slow parsing were identified as limitations of the IFC.

Admittedly, [14], acknowledged that modern AEC workflows increasingly demand capabilities that are fundamentally incompatible with STEP-based architectures. Particularly, the development of digital twins, Internet of Things (IoT) integration, artificial intelligence applications, and real-time collaborative environments requires transactional data access, partial model updates, low-latency exchange protocols, and API-driven integration patterns. The current file-based IFC cannot adequately support the exchange [14].

Until IFC 4.3, it is appropriate to acknowledge that the IFC in its STEP format was purpose-fit; however, its capability diminishes considering the rapid modern advancements in technology. The issues pertinent to the IFC 4.3 promises to be resolved with the imminent release of the full IFC 5 version. Promising as it may be, it is important to note the slow change adoption process in the AEC industry, which means that these problems may persist for a prolonged period.

2.2.4 IFC 5: Problems resolved and those that persist.

Unlike IFC 4.3, which maintains compatibility with the traditional STEP foundation, IFC 5.0 represents a fundamental shift in paradigm toward contemporary data modelling and serialization approaches. The IFC 5 standard discontinues EXPRESS and STEP Physical Files in favour of UML-based schema definitions, JSON serialization formats, and modularization principles that align with modern software development and practices.

This shift directly addresses the core limitations implicit in previous IFC versions, which have limited IFC implementation efficiency and adoption rates. In IFC 5, all API-driven data access will be supported. Transactional updates and cloud-native deployment patterns are enabled in the new IFC 5. This progression facilitates seamless integration with contemporary software architecture and emerging technologies such as machine learning, artificial intelligence, and IoT platforms [11].

These updates address the arguments raised by [10], in relation to the IFC STEP format. Beyond mere format changes, the modernization effort includes fundamental restructuring of the data model itself. The new architecture employs late-binding mechanisms that enable dynamic schema extension without requiring software recompilation. Additionally, modular design principles facilitate domain-specific customization and normalized relationship structures that simplify implementation complexity.

Overall, the IFC standard stands out as a practical format that needs critical consideration for integrating BIM and GIS in infrastructure projects due to its open, vendor-neutral schema and object-oriented design principles. IFC enables rich semantic representation and extensibility. Additionally, its hierarchical class structure, property sets, and inheritance model provide a comprehensive semantic network that supports detailed classification, automation, and multidisciplinary collaboration in complex infrastructure projects [10].

2.3 BIM Software

Revit has emerged as one of the leading software in BIM, enabling professionals in the AEC industry to create parametric models that contain rich geometric and semantic information [7]. A dominant feature of Revit is its collaborative workflow capability, which supports multidisciplinary coordination throughout the building lifecycle. This makes it one of the foremost software leaders that enables data transformation [15]. Revit's family-based approach allows the creation of parametric components that are intelligent to support the downstream analysis and design validation process with the right arguments [16]. This feature is useful for rule-based checking in the native modelling environment.

Revit supports the IFC schema and provides both import and export for numerous IFC schemas, including XML. Research, however, indicates that IFC has not been fully utilized, considering its promising potential [7]. Noardo et al., 2021 also indicate challenges that exist when dealing with infrastructure elements and in some cases, complex architectural geometries. This specifically relates to geometries that require exact representation for analysis and design validation [7].

Integrating GIS data into the Revit environment demands cautious consideration of data formats, coordinate reference systems, and the level of detail appropriateness [17]. The typical workflow begins with the acquisition of spatial data from municipal GIS databases. For example, the Netherlands' GIS database hosts 3D City buildings in LOD2 and a topographical model in LOD1. Both files are saved in shapefile format. Once acquired, the data is processed and integrated into the native modelling environment through file conversions and coordinate system transformations. Currently, Revit's support for GIS imports is limited to CAD files and, on some occasions, through third-party plugins. Regardless of the process used, manual conversion and adjustment of GIS data is required to maintain data integrity. The resulting BIM model, beyond providing spatial context, can be used for rule-based design validation to ensure compliance with building codes and municipal zoning regulations. The validated BIM model may further be exported into GIS for broader urban analysis and planning integration [17].

Semantic richness of GIS data is lost during the conversion and importing process, which renders imported data only fit for geometrical functions. This gap highlights a need for improved GIS-BIM integration tools and standardized workflows that preserve both geometry and semantics. The usual workflow for GIS - BIM integration involves converting and importing site topography, existing infrastructure data, and cadastral data into a native modelling tool such as Revit. These imports enable engineers and architects to create contextual designs; however, the conversion process compromises the semantic integrity of the data, and most importantly, the coordinate systems in both formats have to be properly coordinated.

2.3.1 Coordinate Systems in BIM-GIS Integration: Local Placement, Coordinate Operation, and Spatial Reference

Mapping the location of objects in BIM and GIS is an important factor for successful BIM and GIS integration. The precise determination of spatial positioning directly influences both the orientation and absolute location of data once information has been exchanged between platforms [18]. Both BIM and GIS observe universally accepted map reference systems; the underlying differences in their coordinate system implementations create substantial challenges for seamless integration. For example, Revit allows a maximum ground coverage of 20 miles (32 kilometres) per project, leaving much to be desired when compared to the extensive spatial requirements of most civil infrastructure projects.

Given this limitation, native tools typically adopt local coordinate systems that are relative to universally established global reference systems such as WGS 84. This potential is advantageous; however, the fundamental challenge lies in reconciling these different coordinate systems when data is transferred while maintaining spatial accuracy and data integrity. The following discussion examines reference systems within both BIM and GIS platforms, with particular focus on Autodesk Revit as the representative BIM platform. The differences in coordinate systems is depicted in Figure 4 (a) and (b).

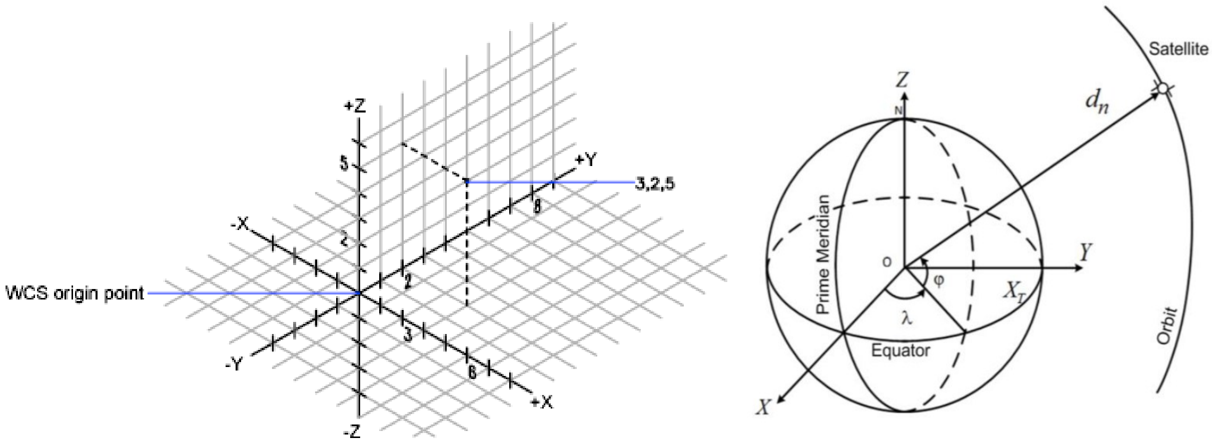


Figure 4 (a) and (b) 3D Cartesian Coordinate System v Geodetic Coordinate Systems

In Revit, georeferencing is supported by linking models to global Coordinate Reference Systems (CRS) such as WGS 84 or, in some cases, localized Projected Coordinate Systems (PCS). It allows for the definition of a project base point and shared coordinates; however, its native handling of geospatial data is limited [15]. To preserve the CRS information in an IFC export, additional metadata is required to protect this information as most BIM software does not fully automate geographic coordinate integration [19]. IFC 4.3 provides an option to reference a model to an identifier of the European Petroleum Survey Group (EPSG) database when exporting. Yet, it is unable to handle custom CRS, which is often defined for megaprojects or transnational projects in GIS platforms.

Another challenge in referencing between the two domains is that IFC offers several options to encode georeferencing (Figure 5). During model conversion to IFC, BIM, various BIM authoring tools choose a combination of these available options. For instance, one tool might store coordinates in latitude and longitude, while another offsets the coordinates in IfcMapConversion. In the IFC schema, this is syntactically valid; however, it is semantically inconsistent. The Level of Georeferencing (LoGeoRef) framework was introduced as a way to address this discrepancy [20]. To establish this framework, the IfcPostalAddress entity was used. An IfcPostalAddress is human-readable and the simplest way to describe a site or building, providing actual postal codes.

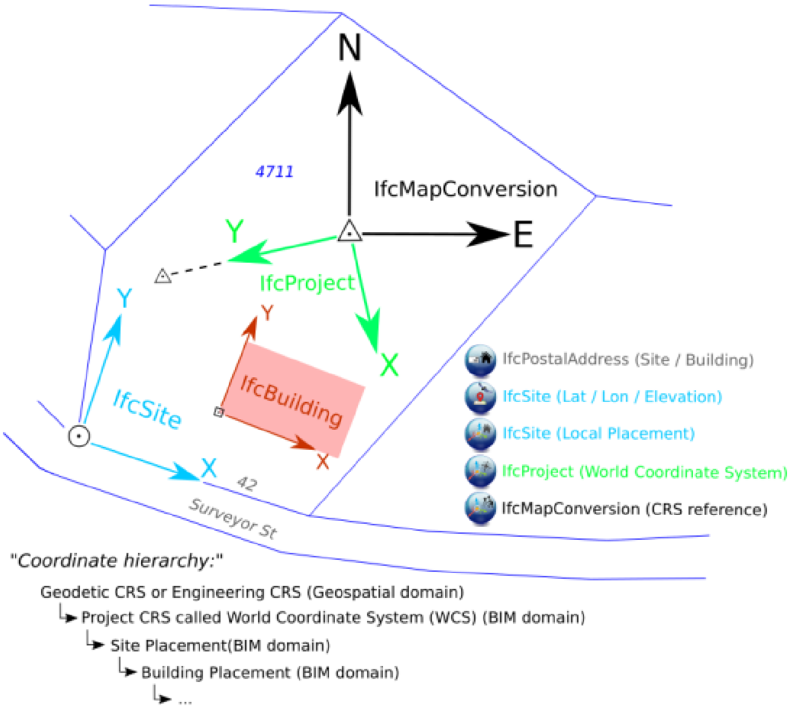


Figure 5 Ways of Georeferencing Encoding in IFC [20]

The proposed LoGeoRef framework is summarised in the following:

LoGeoRef 10 is the simplest way to describe a site's location by adding a postal address in the native BIM project. It offers an approximate location of the site. This level primarily uses the IfcPostalAddress entity to store information about the site's location with basic inputs such as town, region, and country. In Revit the location information can be added at the Location section under the Manage tab.

LoGeoRef 20 uses the geographic coordinates or a point on the map. Here, IFC stores a single point coordinate with latitude and longitude and an option to add elevation. Latitude and longitude are separated by commas and typically represented according to the World Geodetic System. An instance of IfcSite stores the values with RefLatitude, RefLongitude, and optionally**,** RefElevation.

LoGeoRef 30 allows for the storing of any IfcSpatialStructureElement directly in its local placement. This level facilitates the storage of both location and rotation relative to true north. X, Y, Z coordinates along with vector components necessary for angle specification and rotation are stored. This level uses the IFC entities: the RelativePlacement attribute and the IfcLocalPlacement object. It is useful only or spatial structure elements that do not have a relative placement to other spatial element structures.

LoGeoRef 40 provides attributes that stores georeferencing for the entire project context. It also facilitates rotation and translation of the PCS. Typically, the location is stored in WorldCoordinateSystem, and there is an optional true north rotation that can be specified via the

TrueNorth attribute. The TrueNorth attributes provide an option to set a distortion relative to the north direction; however, it can be confusing if direction attributes are also set at WorldCoordinateSystem. At this level, the location of the project is stored in an instance of IfcCartesianPoint, and optional directions for X and Z axis are stored in an instance of IfcDirection.

LoGeoRef 50 represents the highest quality regarding georeferencing of an IFC file. The levels can specify the CRS with metadata. The targeted CRS can be described at this stage with and through the option to specify the EPSG code. This capability was introduced into the IFC 4 schema. The IfcMapConversion entity stores information such as the Eastings, Northings, OrthogonalHeight, XAxisAbscissa, XAxisOrdinate, and Scale. The source CRS must be of the type IfcGeometricRepresentationContext, and the target CRS is typically an instance of IfcProjectCRS.

[21] also emphasized the importance of a fully prepared BIM model that is unambiguous and can be readily used across various AEC software and tools. This should be done without manual intervention. However, levels below LoGeoRef 50 do not provide any information about a project's CRS. The LoGeoRef 50 is the highest quality CRS level, and [21] focused their research on two solutions. Solution A prioritizes keeping the existing IFC schema valid, and it proposes optional attributes that are added to the existing entity definition. A new entity, IfcGeographicCRS, is introduced with its own WellKnownText attribute. Solution B is a sharp contrast to this, adding the WellKnownText directly to the parent entity IfcCoordinateReferenceSystem. The trade-off is that some existing IFC 4 and 4.3 tools may no longer validate such models. Figure 6 presents a summary of the proposed solution.

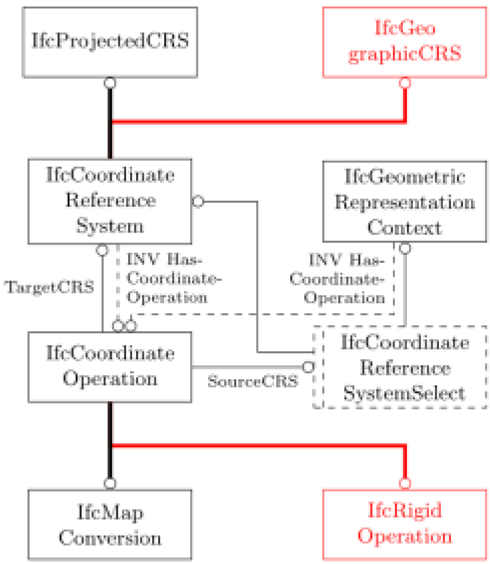


Figure 6 Diagram of existing and proposed entities for Improving Georeferencing in IFC (Existing in Black, Proposed in Red)

Regional projects in BIM are operated in localized PCS for precision; this is common with many BIM authoring software [19]. However, infrastructure projects require alignment with global CRS for transnational consistency. Global CRS systems such as WGS 84 or UTM provide standardized frameworks but introduce scale distortions over large areas (reference). Localized geospatial PCS prioritizes minimal distortion within a project's vicinity by scaling coordinates to a reference plane. Jaud et al., [22] demonstrates the issue of map projections in their study of the Brenner Base Tunnel (BBT), a 55km long railway which links Austria to Italy. BBT employed a customized geospatial CRS that was created to balance ellipsoid-based global data with localized elevation adjustments. A transverse Mercator projection with a central meridian was optimized for the Alpine region to reduce cumulative errors across its 55km length [22]. By doing that, an alignment of the tunnel was accurately achieved to a millimetre. BIM software, such as Revit, lacks support for such customizations, compelling teams to manually adjust scale factors. This challenges data exchange, as IFC schemas struggle to encode non-EPSG standardized CRS [22].

Another limitation of BIM software is the diminishing coordinate accuracy over extensive infrastructure networks. Localized PCS maintains precision within a kilometre of the project base point due to scale factor approximations [22]. Beyond such ranges, earth's curvature and projection distortions introduce measurable errors, and that is the reason most large-scale infrastructure models are segmented. While Revit and other BIM software excel in localized CRS with limited Georeferencing capabilities, their dependence on manual CRS configuration and limited support for custom map projections hinders scalability in translating infrastructure such as the BBT. For seamless BIM and GIS integration, automated coordinate transformations and expanded support for customized CRS, future advancement must be prioritized.

In all, the successful integration of BIM and GIS largely hinges on accurate map projections and Coordinate Reference Systems. Interestingly, Pedó et al., [23], identified georeferencing as having the least importance to architects, and this poses a setback for BIM and GIS efforts. Hence, information requirements must be specified at the start of each design project to stipulate the level of information required to designers [24].

2.4 GIS-BIM integration

The integration of Building Information Modelling (BIM) and Geographic Information Systems (GIS) has garnered growing interest from the Architecture, Engineering, and Construction (AEC) industry in recent years [2]. These efforts have been driven by the need to bridge geometric, procedural, and semantic gaps [2], and [4]. The geometric capabilities of such integration is beneficial to architects, engineers, and planners during the design and construction stage [25] . Communication challenges and the complexity of setting up and publishing apps are also some key challenges that render the procedural approach difficult [23]. Among all the available methods for GIS BIM integration, semantic web ontologies have been the most widely researched [2]. In most studies on the subject, a bidirectional communication framework is strongly encouraged since it facilitates reusability and flexibility. In their research on BIM and GIS integration for roads. [26] reaffirmed this theory and stated further that it is beneficial for seamless integration of Geometric and non-geometric data, enhanced lifecycle decision-making, web-based intelligent platforms, and the aptitude for scalability to Digital Twin. Practically, it is easier to convert BIM models to GIS formats due to the linearity and relative simplicity of the geometric and semantic requirements needed for analysis in GIS [27].

Until now, integration attempts have heavily leaned towards processing BIM data for analysis in GIS, with CityGML being the preferred file format [23]. A concern regarding BIM to GIS integration is the loss of data through conversion in both domains. In the case of BIM to GIS, rich semantic data can be compromised since GIS domains lack the level of detail required to represent BIM's complex geometry. However, converting GIS - BIM is less susceptible to such losses due to the general, simplified geometrical representations in spatial data. This theory is affirmed by [1] in their CityGML to IFC Python script that converted GIS data to IFC. When a BIM model is converted into GIS platforms, it adds context and makes the meaning of various spatial analyses ascertainable. For example, the true impact of flooding analysis can be properly measured when the surrounding buildings are indicated in the GIS platform. The inverse of this is not straightforward for BIM. Though GIS data imported into a native BIM environment adds spatial contexts to designs, BIM's central focus is information modelling, and the full potential of GIS data in BIM can only be maximized when semantic meaning is derived from the linked GIS data.

Contrary to the popular CityGML format commonly used for BIM and GIS integration, an alternative approach was proposed and examined by [27] as a way to resolve the existing integration challenges. the computer graphics technique was used with 13 building models being examined. It was discovered that the shapefile format is relatively easier and more flexible. Evidently, researchers have neglected the many file formats inherent in GIS in their discussion of GIS - BIM integration, but Zhu & Wu, [27] is an exception.

The discussions about GIS - BIM integration should not neglect the other data formats inherent in GIS, since various geospatial data are represented in different formats. Engineers and architects can design faster with spatial data that adds context to their designs; however, it is more beneficial if associated semantic information can be easily queried in the native modelling environment. Additionally, this approach will also reduce the iterative process in most design processes, which is aimed at meeting local and jurisdictional regulations.

2.4.1 GIS - BIM: Existing Tools and Potential

The core imperative of BIM and GIS integration is to decentralize and share information from siloed domains and tools while maintaining the geometric and semantic integrity of the transferred data [2]. This can be used for a multitude of applications, which include setting the context for sites [1], and is also processed for use in various ontologies [2], [4]. To this, several tools have been created. Most of these tools are focused on bridging semantic gaps, and the geometric and procedural gaps are lightly probed [25]. A few of these approaches are discussed.

Semantic Web Technology

A review of existing GIS and BIM integration approaches will be incomplete without a discussion on Semantic Web Technologies (SWT). SWT enables data to be shared and reused across applications and domains through standard ontologies [4]. In BIM and GIS integration, SWT enables the creation of shared ontologies that bridge domain-specific vocabularies, allowing data integration and querying.

SWT is composed of four distinct parts. The first is Ontologies, which are controlled, machine-readable vocabularies that define concepts and relationships within a domain [2]. The IFC files are translated into an ontology such as ifcOWL [28]. Ontologies provide structured vocabularies and relationships to represent entities in a building or infrastructure. Ontologies in GIS represent geospatial entities such as parcels, terrains, and water bodies [2]. Resource Description Framework is another feature of SWT that adopts the triple approach as a standard for data exchange on the web [4]. The triple method uses a subject-predicate-object format to store information. For instance, “Mecartor” – “has_entrance” – “Doors”, where Door is an object in Mecartor, and they are tied together by the “has_entrance” predicate. SPARQL is the third feature of SWT, and it is used to query data from RDF data [4]. Lastly, the results of the linked structured data are interlinked and published for subsequent use. The cube in Figure 7 shows the layered nature of SWT.

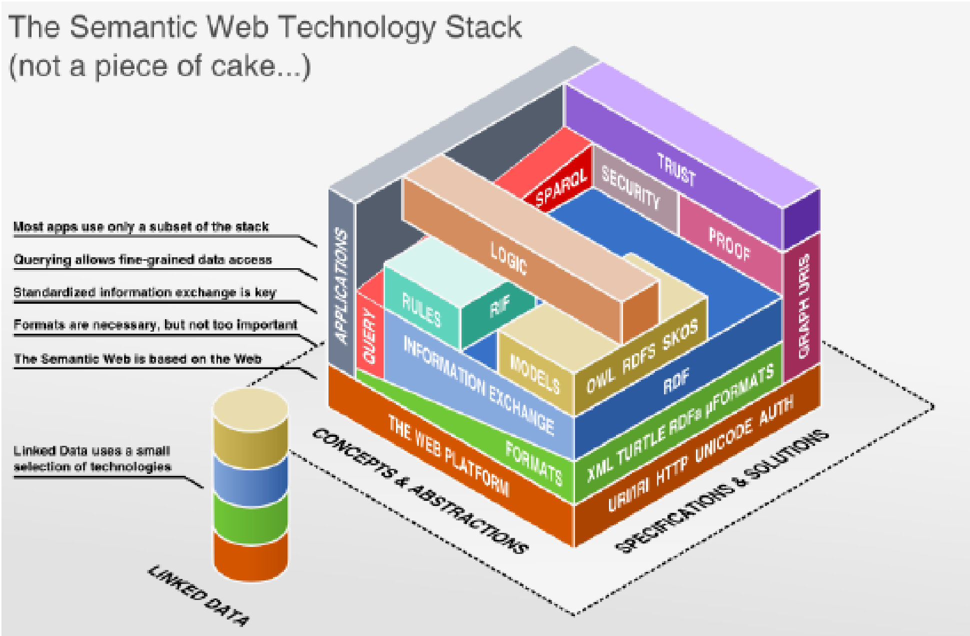


Figure 7 Semantic Web Technologies and Linked Data [29]

Ontologies present an excellent medium for GIS and BIM data integration; however, recent studies indicate that their practical deployment is hindered by many factors. These include large data sizes, complexity, scalability, and tooling immaturity [2]. Additionally, Celeste et al., [3] pointed out in their study that most proposed solutions to BIM and GIS integration remain at the prototype level. Consequently, researchers recommend a lightweight and practical solution [4]. Another challenge is that most regulatory requirements are stored in text formats, and it is difficult to encode these regulations into machine-readable formats. Further, interpreting building codes requires a measure of human judgement, which is difficult to capture in automated systems. This reduces the confidence levels in the reliability of automated design validation processes [30].

Converting geospatial data to IFC using a documented schema is a more immediately viable pathway to integrate spatial data with BIM workflows, effectively preserving semantic meaning while keeping data lean. Although this option readily aligns with the existing industry toolchain and permits incremental adoption, it hasn't garnered much interest from researchers. Yet, appreciable efforts have been in this regard. The next heading delves into these endeavours.

GIS-BIM Conversion Tools

Several tools and third-party plugins have been developed to enhance Revit's capabilities to integrate GIS data. Most of these tools address specific aspects of the integration challenge, such as coordinate reference system transformation, GIS data format conversion, and automated design validation checks. Some of these tools include Dynamo and Esri's ArcGIS for Revit.

Salheb et al., [1] researched the possibility of integrating BIM and GIS using a Python script. This ultimately resulted in a Python-based tool, CityGML2IFC.py. To validate their implementation, various software platforms, including FZK Viewer, ArchiCAD, Revit, and specialized IFC checking tools, were used. Their approach involved four distinct steps, which include encoding transformation, geometry mapping, coordinate system transformation, semantic mapping, and topology preservation. The methodology was examined using the Rotterdam 3D 2.0 dataset, demonstrating successful conversion of LOD2 buildings to semantically accurate IFC2x3 files. The resulting models were:

Their work is unique in that it practically explores the possibility of integrating GIS data into a BIM environment. This is an area that has not received much interest from researchers, although it is needed to realize a bidirectional BIM-GIS integration approach. Recent studies have emphasized the importance of interoperability between these domains [2], [4]. Despite IFC's complexity, it is also flexible [1]. This was strong motivation for me to consider the IFC schema as the file exchange format.

A gap in their research is that IFC 2x3 was used, a version of the IFC that had not matured to include civil infrastructure and was more building-centric. Given that the IFC is a progressive schema, it is prudent to probe the use of other classes to represent geospatial elements that are not classified in the IFC schema. Another gap is that CityGML was used, which is justified because it has become the most used data exchange format between GIS and BIM; however, other formats that are also representational GIS formats must be considered. Lastly, providing spatial context to the building is useful for analysis such as shadowing and clash detection; however, for the purposes of building permitting and compliance to jurisdictional and local regulations, most data exists in attributes that are associated with physical elements, and therefore, the potential of using these attributes for design and design validation in native modelling tools must be explored.

The Change Toolkit for digital building Permit has also contributed significantly to GIS-BIM integration. Their initiative is funded by the European Union and their mandate is to champion implementation of digital building permits. The recent Geo – BIM tool procedure focuses much on integrating CityJSON files into native BIM environment [31]. It consists of GIS library and a convertor that transforms open geospatial standards such as GML, CityGML, CityJSON, LandXML, and InfraGML into valid IFC4 ADD TC1 model. The GIS extends RDF's Geometry Modelling Kernel, providing web representations. Though this advances integration efforts, it still indicates limited support for purely 2D GIS data. Many planning datasets such as cadastral maps, and zoning layers are still in 2D.

2.5 BIM Data Extraction

BIM data can be extracted for different purposes and through different procedures. In most native modelling software, the data of specific elements can be extracted or queried using plugins. The accuracy of extracted data will, however, be determined by the Level of development (LOD) or, in the specific case of BIM, the Level of Information Need (LOIN). LOIN, as defined in ISO 19650, establishes the framework that specifies the granularity of information required for BIM models as relates to geometry, alphanumeric attributes, and documentation. The standard aims to reduce information waste by formulating data delivery milestones to project phases and stakeholder needs [31]. LOD in BIM refers to the geometric granularity of models. In GIS, LOD refers to the degree of geometric, thematic, and attribute detail associated with a model. LOD for basic vector geometry can be specified as well as 3D GIS. At a low LOD, a road can be represented with a single line. The road can be represented with sidewalks, lanes, and attributes such as surface types. CityGML categorizes models into five levels (LOD0 – LOD4) according to the geometric depth. In geospatial modelling, LOD ensures consistency in urban representations, from basic vector data of regional infrastructure footprint (LOD0) to 3D city models (LOD4) [32].

Of all BIM data extractions, the extraction of material quantities has been the one area significantly researched, leading to the development of several plug-ins for software and workflows that maximize quantification. Extractions for energy simulation have also seen advancement in recent years. In energy simulation, BIM models serve as rich sources of geometric and material data that feed into Building Energy Modelling (BEM) tools. However, BIM data extraction to BEM also has export issues between software and often requires manual intervention to correct data mismatches, making the process time-consuming and error-prone [33]. A common interoperability issue of BIM to BEM data exchange is incomplete or inconsistent data mapping, which hinders automated workflows and accurate energy performance predictions.

BIM data is also useful for noise simulation. When combined with spatial data, BIM models can also predict acoustic performance for a given site [34]. BIM data extraction modules retrieve room geometry, material absorption coefficients, and spatial arrangements to support acoustic analysis [35]. This enables designers to assess and optimize acoustic quality early in the design process. This approach integrates BIM with frequency analysis and sound effect simulation, demonstrating the potential use of BIM data extraction to enhance multidisciplinary analyses. Embedding spatial data into simulations increases the accuracy of prediction.

Lastly, integrated spatial data and BIM also facilitate automated design validation processes through rule-based compliance checking [30]. Automated code compliance checking is important for advancing the AEC sector in design validation. The capability to validate designs in BIM software enables continuous compliance monitoring throughout the entire design development process. This reduces the chances of oversights and errors inherent in manual design validation processes. In Revit, design validation is primarily implemented through scheduling and filtering elements and parts in a model [36]. This allows elements to be queried based on parameter values and relationships. Revit's parametric nature provides a solid foundation for rule checking; however, native rule-based checking is limited, and to perform more sophisticated checks, additional tools and plugins are required that can interpret complex building codes.

In Revit, rule-based checking involves using visual programming tools that allow users to create custom validation scripts. Recent research has explored the possibility of integrating knowledge graphs with BIM for automated compliance checking, as this provides a flexible and extensible approach [37].

3 METHODOLOGY

Three approaches exist for GIS and BIM integration. The first and widely used method is integrating BIM models into GIS, which is beneficial for several spatial analyses in GIS platforms. The second involves exporting GIS models to BIM. There's also an option to integrate both GIS and BIM in separate software or platforms, such as semantic web technologies. This research focuses on the second approach, which is exporting GIS data to the BIM environment. There are challenges inherent in all the integration options; however, each approach is advantageous for specific tasks. To achieve integration with the selected option, the IFC schema will be used. This approach is not new; however, previous studies have focused mostly on the geometry of GIS in a native model environment [1]. This research tests the possibility of extending the usefulness of linked GIS data by focusing on semantics.

The release of IFC 4.3 broadened the schema to capture elements in the civil infrastructure industry. However, these new classes introduced were not exhaustive of all geospatial elements. This challenge was emphasized by [1] who admitted the difficulty in representing geospatial objects that do not have IFC class representation. To address this deficiency, a script in FME was created that focuses on creating spatial objects that may not be represented in the IFC schema or that are not supported by the Extract Transfer and Load tool being used. Specifically, IfcGeomodel was created, which was not supported in the FME version used. This attempt was to address the gap in the earlier attempt by [1]. Through this workflow, the flexibility of IFC was exploited.

Another research gap is filtering out GIS data to precise geometric and semantic boundaries before integrating with BIM. This increases storage requirements and leads to information waste. [1] admits this limitation. To address this challenge, the proposed workflow involves filtering out GIS data as the first stage in GIS BIM integration. This reduces the GIS information to only the required data. Additionally, Geopackage, a GIS data exchange format, was used, which extends the thoroughness of BIM GIS integration attempts. The focus of the case study was mainly on the associated spatial attributes and their derived meaning, rather than the geometry and georeferencing.

To achieve this aim, the workflow of the methodology shall include creating geospatial vector elements of a selected project site (Figure 10). Particular attention was given to proper attribute specification and its direct implications for GIS integration. It also addresses the relevance of coordinate reference systems in GIS BIM integration. Using FME, the created GIS data was transformed into IFC. This section focuses on understanding the workflow in FME to transform raw GIS data into IFC. Appropriate property sets were also mapped to enhance the semantic usefulness of the IFC model. The last step involved testing the created IFC in a native BIM model to understand its usefulness for design and design validation. **Error! Reference source not found.** shows the general workflow, and **Error! Reference source not found.** indicates zoning for the case study project.

Autodesk Revit was selected as the BIM software to use based on reports from [38] seminal work in BIM and GIS integration that involved testing integrated models in 33 software packages. In their report, Autodesk Revit emerged highly responsive when it comes to reading various datasets, hence its selection. The Feature Manipulation Engine (FME) was selected as the (ETL) tool due to its vast array of geospatial data format processing. FME's operation can be broadly categorized into three: Readers, Transformers, and Writers. Readers import various data formats into the workspace, Transformers modify and manipulate data into different formats, and Writers output the transformed data in various file formats.

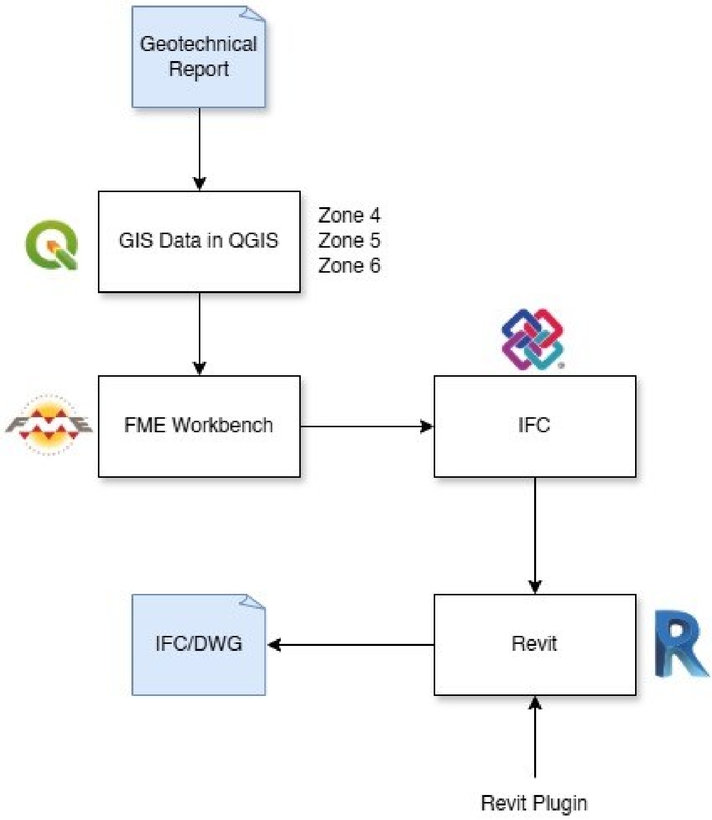


Figure 8 General Workflow

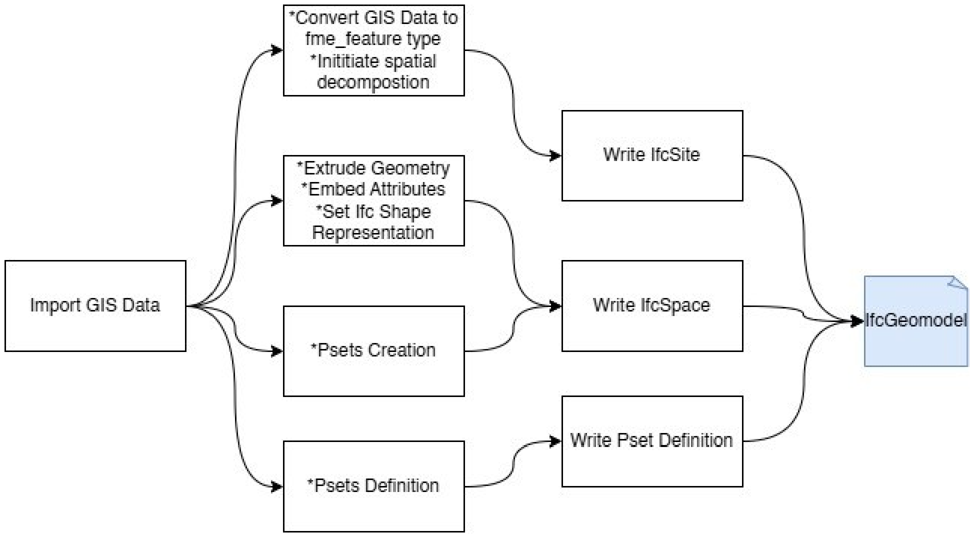


Figure 9 FME Workflow

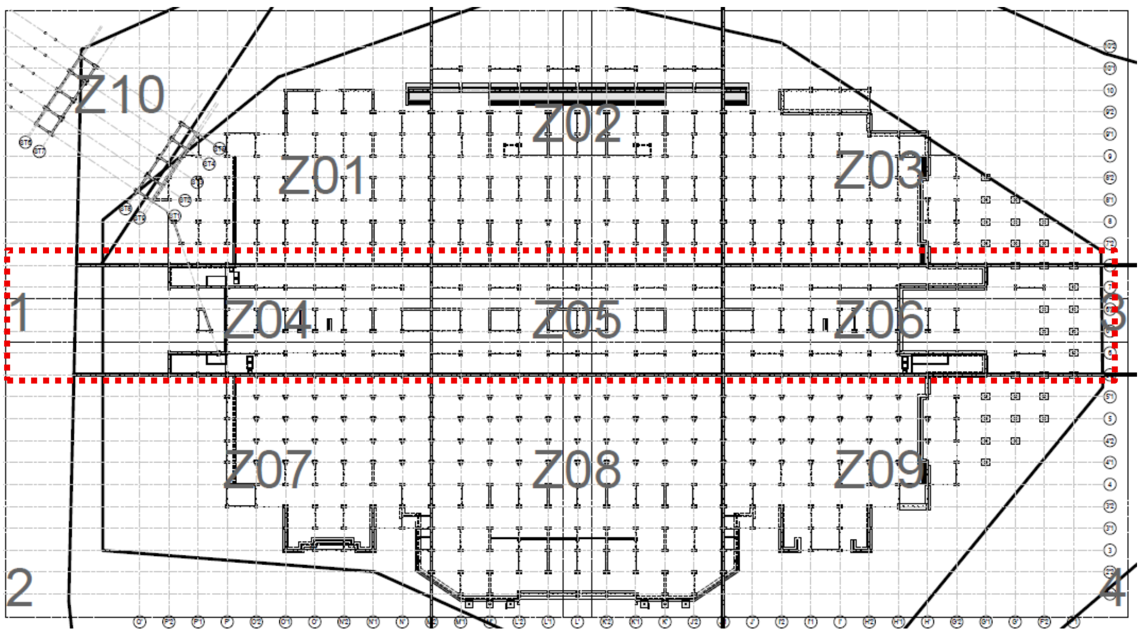


Figure 10 Extracted Zones for Case Studies

3.1 Spatial Data Information Specification

A site in Amsterdam was selected for this study. This is because Amsterdam is one of many cities that have 3D city buildings and geodata for most municipalities. Additionally, the topography of most of the city is available and is typically stored in shapefiles. The variances in the data formats provided on municipal archive sites influenced the decision to use Geopackage as a format for the GIS data. File formats in GIS are numerous, and research advances have mostly focused on CityGML as the file format for GIS. Both [13] and [38] highlighted the gap that exists in trying other geospatial data formats. Hence, it was necessary to use Geopackage to extend the thoroughness of BIM and GIS integration efforts. Three different zones or sites, contained in a parent site, were used. This was to fill the research gap as highlighted by [13], that multiple sites with unique property sets and attributes must be examined. [13] also proposed that creating an IfcSite for individual parcels, instead of an IfcGeographicElement, as used in their research, should be explored. **Error! Reference source not found.** shows the layout of the site.

The spatial dataset used for this study included data from a geotechnical investigation report. The attributes entered included: soil bearing capacity, specific gravity, moisture content, zone, and description. Such information is mandatory for designing the foundation of structures. Soil is heterogeneous, and it may have different compositions on a single site. Additionally, the topographical information of the site terrain is also useful in determining the buried depth of substructure components. In the chosen project used for this study, the site was zoned into 9 zones, each having its unique bearing capacity. The soil composition was also different, revealing clay, peat, and gravel at sporadic locations on the site. Traditionally, such information is stored in text formats, and designers must refer to these texts during design. This creates room for many assumptions, which could lead to overdesigning or under-designing, both of which have a direct impact on cost, negative or positive. When GIS data containing geotechnical data is linked into a BIM environment, the specific technical information that is relevant to each zone of the site can be accessed directly, reducing ambiguity. This will also improve the efficiency and accuracy of design validation as all data related to foundation design can be queried directly from the same model.

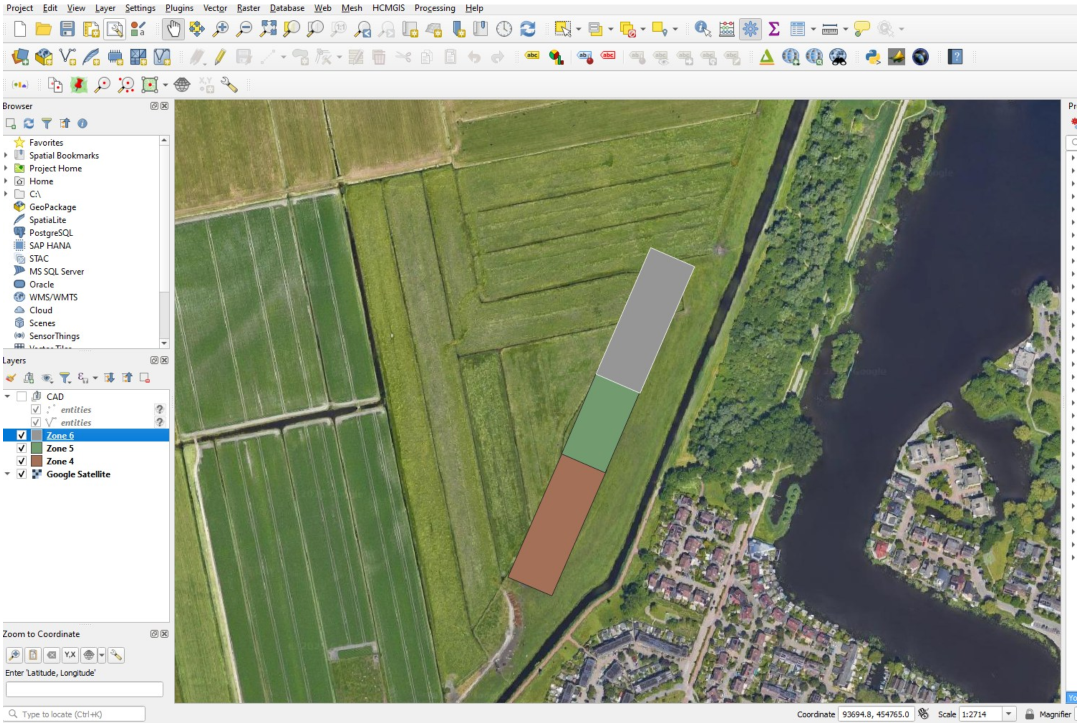
The site is zoned into 3 zones, from Zone 4 to Zone 6. Each zone has its unique attributes. The tree zones are nested into a main site. **Error! Reference source not found.** indicates the created attribute fields for each zone in QGIS.

Fields

Primary key attributes fid
Count 7

Field	Type	Length	Precision	Comment
fid	Integer64	0	0	
Name	String	80	0	
Description	String	80	0	
Depth	Integer64	0	0	
BearingCap	Integer64	0	0	
MoistureCo	Integer64	0	0	
SpecificGr	Integer64	0	0	

Figure 11 Attribute Specification in QGIS



F

Figure 13 sets up the coordinate reference system for the project site. This step is critical to ensure accurate BIM and GIS superimposition.

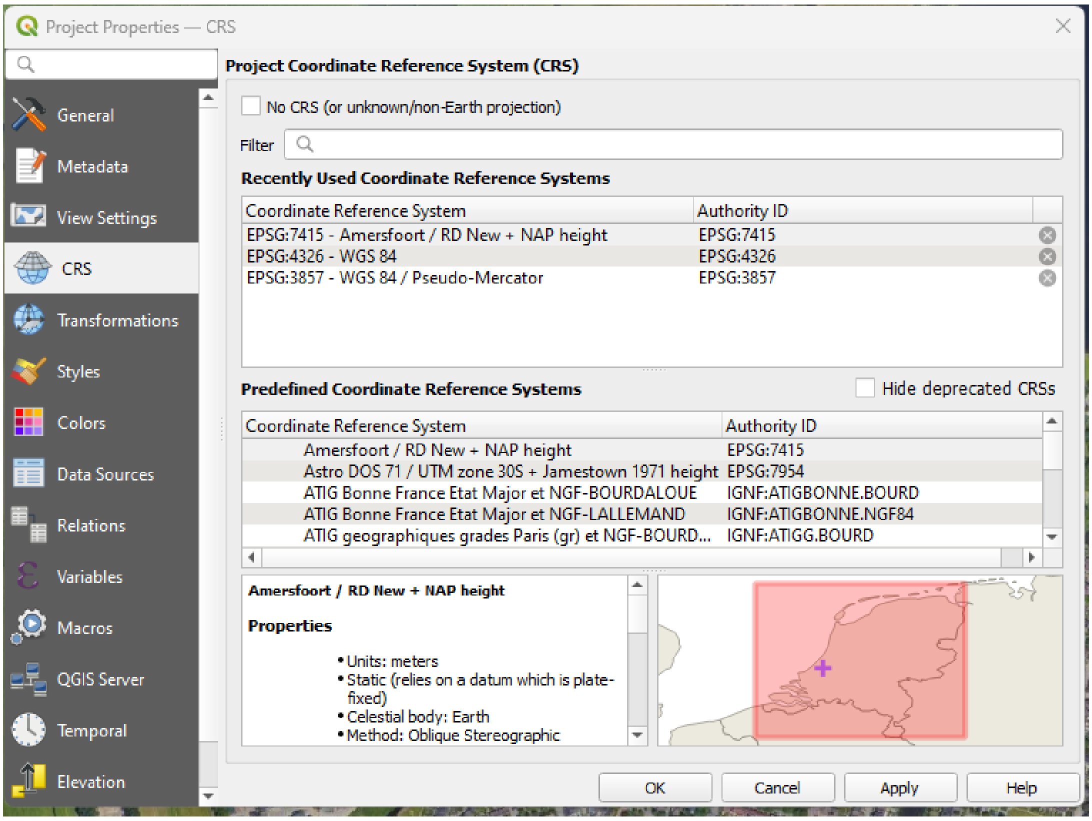


Figure 12 Site Zones 4-6 in QGIS

3.2 Transforming Spatial Data into IFC in FME Workbench

Transforming Spatial data into IFC facilitates the integration of GIS and BIM, and this is beneficial for design and design validation. A challenge in transforming GIS is data filtering, which involves sizing up data to only the required information or attributes [1]. To address this challenge, the Feature Manipulation Engine (FME) was used. FME was selected as the ETL tool due to its wide range of support for geospatial and other file formats. This facilitates the exploration and transformation of the various file formats in GIS. FME operation can be broadly categorized into three types. These are readers, transformers, and writers. Readers import various file formats into the workspace; transformers encode imported file data into other file formats or schemas, and writers output the transformed data.

Error! Reference source not found.4 explains the spatial decomposition observed for the workflow. In all, the proposed workflow involves three transformation stages. Typically, the decomposition for IFC cascades from IfcProject, IfcSite, IfcBuilding, IfcBuildingStorey, and IfcSpace. IfcProject is constant, as is the IfcSite; however, the succeeding decomposition was modified to capture the immediate demand of the project. This is particularly important because spatial data does not always relate to buildings, and their distinction must be maintained. The IFC schema facilitates the adoption of such spatial decompositions. In principle, the nested sites are linked to the parent site by the relationship

IfcRelAggregates. Additionally, IfcSite is both a spatial structure element and can act as a spatial container.

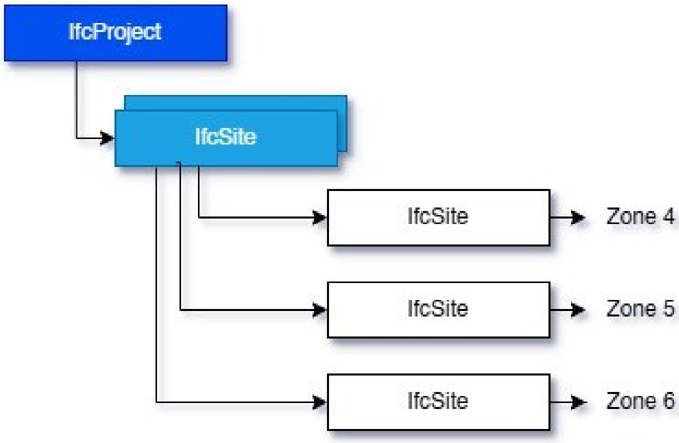


Figure 13 Spatial Decomposition

4 RESULTS

The initial step involves validating the data of the imported spatial data (**Error! Reference source not found.**). The workflow in FME comprises four stages, each stage focusing on specific aspects of the spatial data. The first stage in the process included using the Sample transformer to source features from the spatial data. The various attributes assigned to the vector data in QGIS are considered as features in FME. In the next step, attributes with spatial data are renamed in accordance with the IFC schema. Then, the AttributeCreator is used, which transforms the spatial data into an FME feature type. This stage also cleans imported attributes to only the required data needed, so that information waste is reduced. This addresses the issue of GIS information filtering during transformation in the research by [1]. The created attributes are written into IfcSite by a Writer transformer. Here, the spatial decomposition of the IFC file is initiated. **Error! Reference source not found.** and Figure 16 shows the first stage.

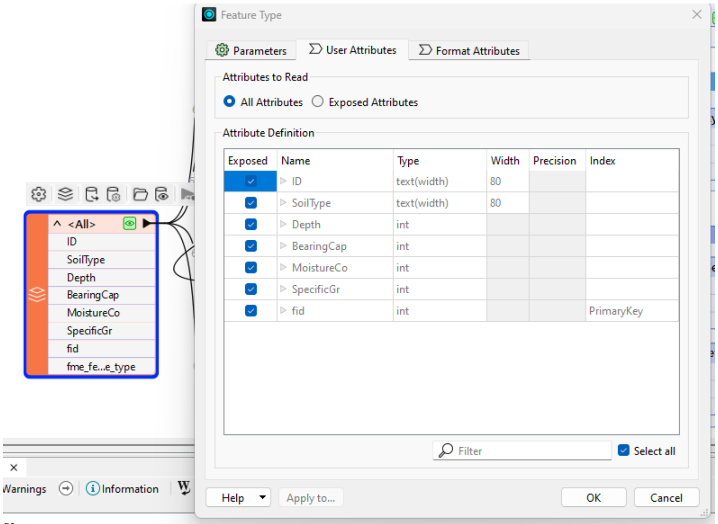


Figure 14 Validating Imported Spatial Data

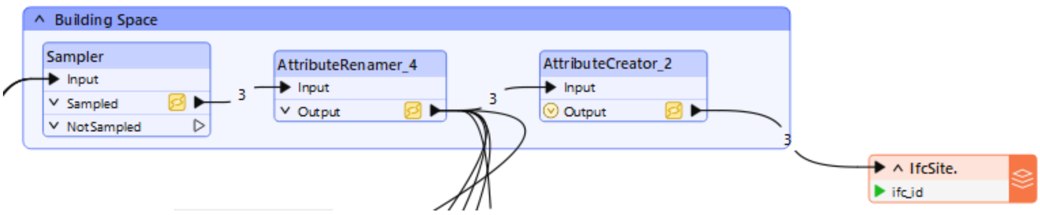


Figure 15 Creating FME feature types, Spatial Decomposition Initiation

The second stage involves transforming the geometry into an IFC-compatible shape (Figure 17). The first transformer used is the Extruder, which adds height to the vector data. This is because the IFC format recognizes 3D and solid geometry. Then, the attribute transformer was used, which creates attributes to be embedded into the extruded geometry. Figure 18a and Figure 18b show the use of repeated GeometryPropertySetter transformers. The first does the actual encoding of the preceding attribute created, thus embedding the attributes to form an integral part of the geometry. The second defines the geometric and topological shape used to represent the extrusion. In IFC, shape representation is used to specify the three-dimensional nature of an object. In this case, "Body" was used, which correlates to a Boundary Representation "Brep" in the IFC schema (8b). The resulting geometry is then passed into an Aggregator transformer, where it will be merged with property sets. Figure 19 shows an inspection of the created geospatial model

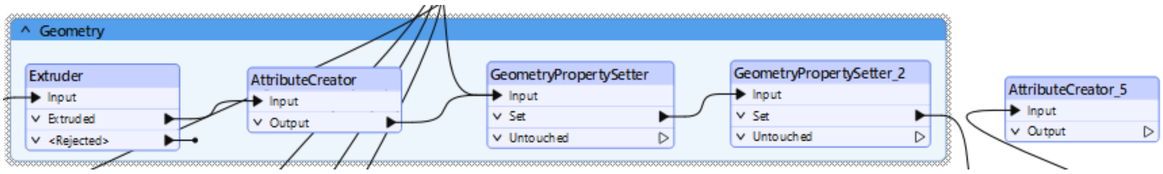


Figure 16 Creating IFC Geometry in FME

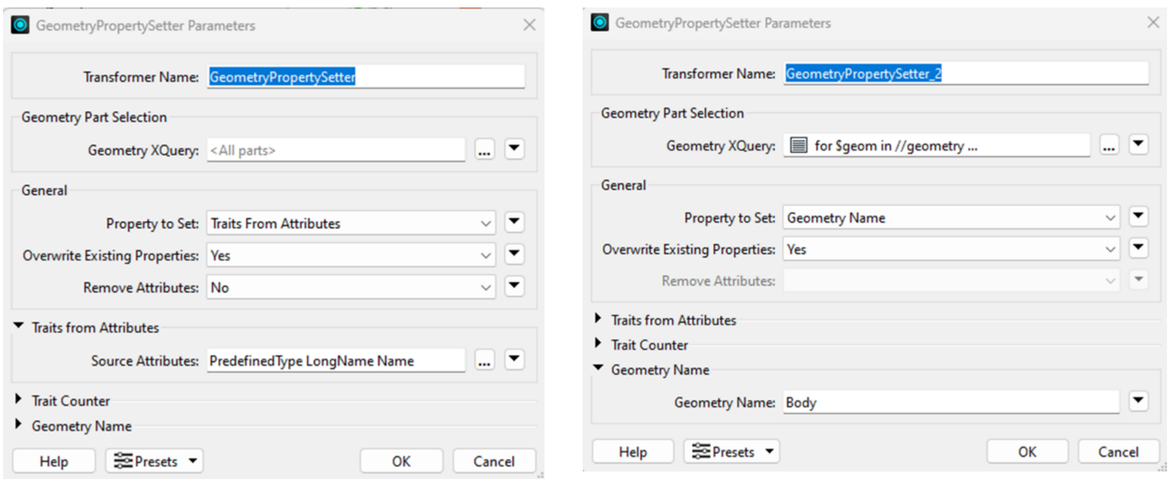


Figure 17 (a) & (b) GeometryPropertySetter in FME

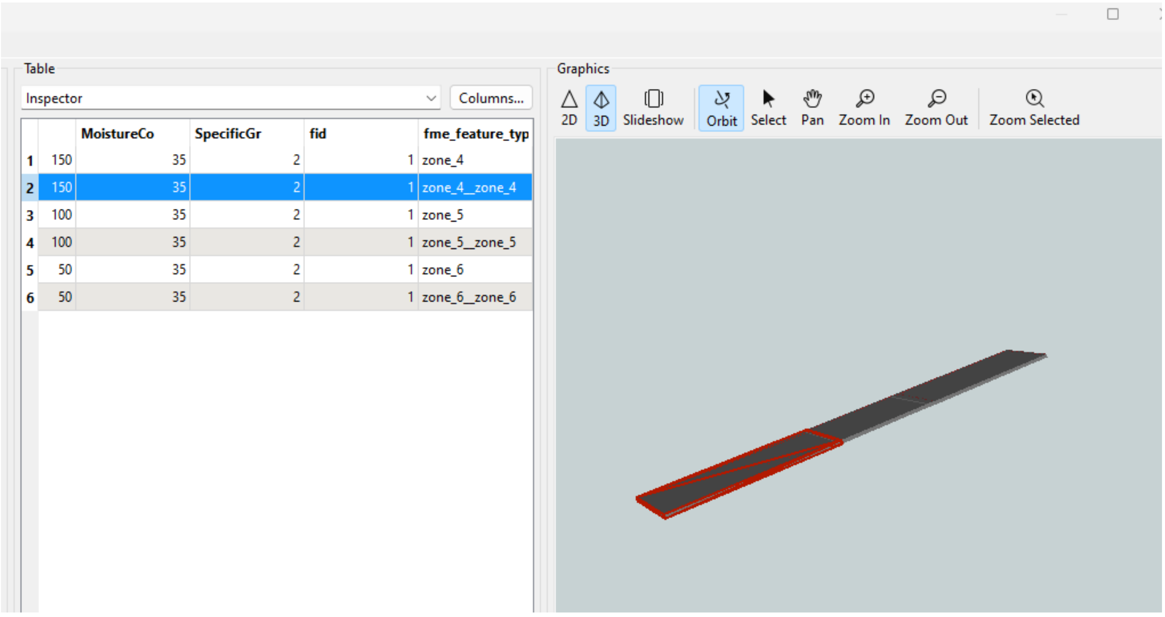


Figure 18 Inspecting Extruded Geometry in FME Data Inspector

The third step involves creating IFC property sets for the spatial data (Figure 20). In this step, the first process is to remove the geometry associated with the data. This is necessary to avoid having repeated geometric data. The next steps included creating the property sets for the spatial data. Two separate property sets were created, but this is expandable. The first property set created is Pset_SpaceCommon, which is a property set associated with the IfcSpace. The addition of Pset should be determined by the required spatial data to be used in the native modelling environment. In this case, the soil bearing capacity of the site was embedded. With this created, the GeometryPropertySetter transformer was used to embed the created attribute into the geometry. The next property set created relates to the identity data. This assigns standard IFC attributes for the IfcSpace class. Using the GeometryPropertySetter, these created Psets were merged and joined to the geometry with an Aggregator transformer. The result of this was written to IfcSpace.

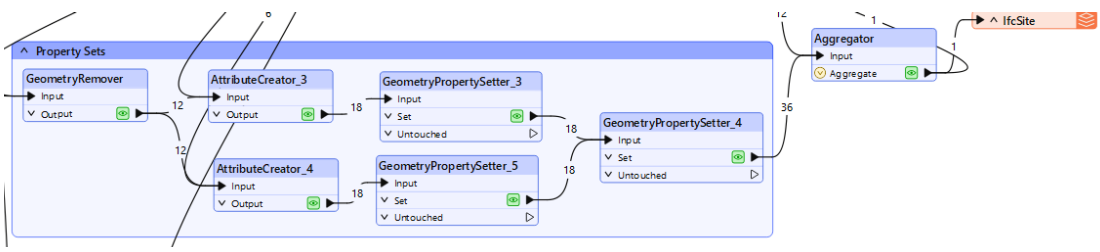


Figure 19 Creating Property Sets (Psets)

The final step involved creating property set definitions for the property sets (Figure 21). The Samper transformer is set to output the first feature (attribute) of each zone that arrives in the workstream. The sampling rate is set to, sampling type “First N feature”, and randomize sampling set to “No”. Again, the GeometryRemover ensures geometry is not repeated by removing the spatial data geometry while keeping the attributes. Each of the IFCPropertySetDefinitionCreator transformers used creates a corresponding IFC property set for each property set created. The IFCPropertySetDefinitionCreator is then written into Psets using the PropertySetDefinition writer.

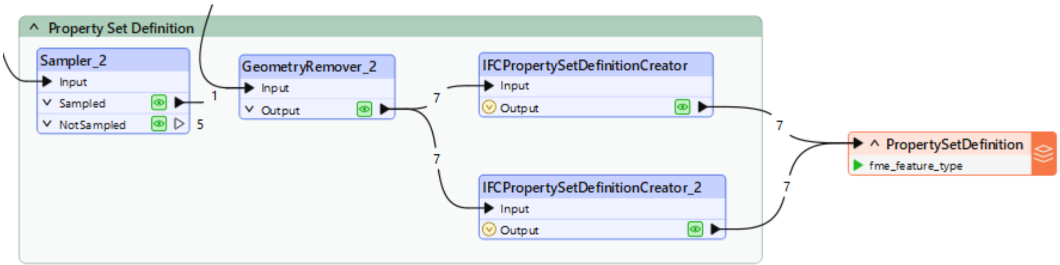


Figure 20 Property Set (Pset) Definition

The workflow, categorized in four stages, outlines the steps used to transform raw spatial data in Geopackage into IFC. Additionally, this workflow focuses on semantic meaning, which is necessary for design and design validation in native modelling software. Additionally, this workflow also addresses the challenge of data filtration during spatial data transformation [1]. By using the Sampling transformer, data was filtered and grouped in a manner that ensures semantic usefulness. The Entire workflow is captured in Figure 22.

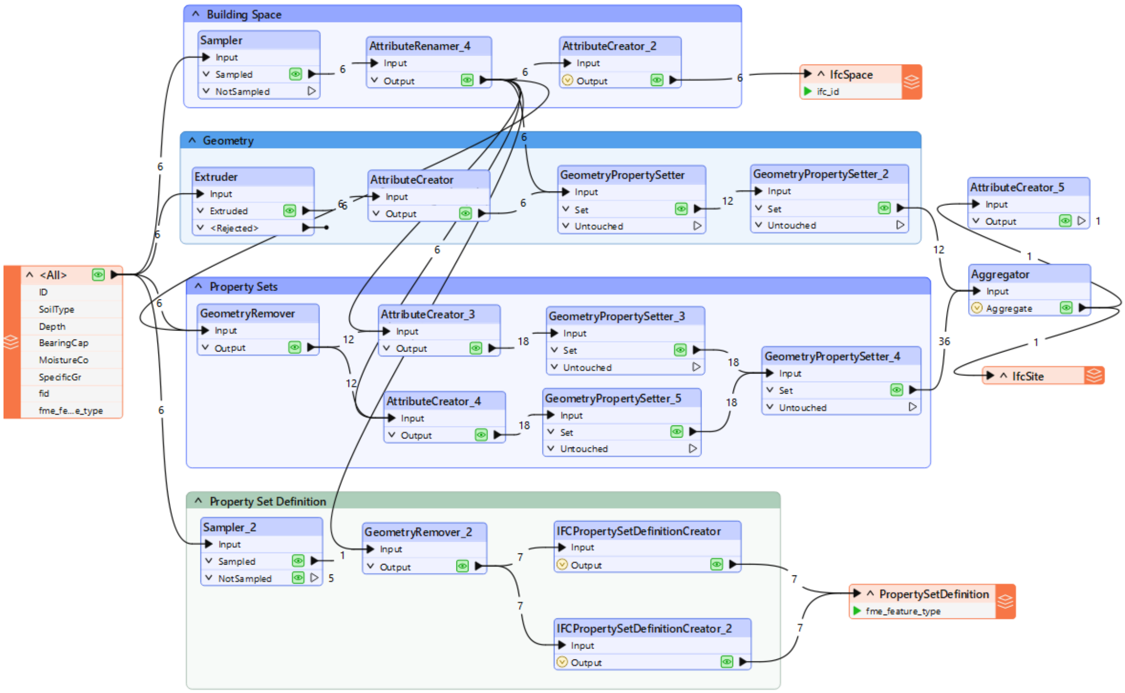


Figure 21 Combined Stages in FME

4.1 Quality of Spatial Data IFC and Extraction Responsiveness

Integrating BIM and Spatial data is advantageous for design and validation. It provides spatial context, which is beneficial for design analysis and validation. Primarily, three approaches exist to integrate BIM and spatial data. These included BIM to GIS, GIS - BIM, and both BIM and GIS can be processed in a separate tool, such as semantic web technologies. This research focused on the second approach, thus, GIS-BIM. Given that both BIM and GIS are created for different domains and therefore differ in data formats, the IFC schema was selected as the file format to enable such integration. The following discussion presents the results of transforming raw geospatial data into IFC format with a focus on the Geometry and Semantics of the data.

4.2 Geometry

In building information modelling, information is the most important component, and IFC facilitates the storage and exchange of information between various software platforms. Classes in IFC can be geometric or non-geometric in nature; in any case, each class is tied to an intricate matrix of activities, resources, or geometry. Geometry in IFC can be represented in several ways, such as SweptAreaSolid, TriangulatedFaceSet, ShellBasedSurfaceModel, and GeometricCurveSet. This study used the

SweptAreaSolid due to the extrusion, which was performed on the flat geospatial data in FME. It was discovered that SweptAreaSolid, which is a valid geometry representation in IFC, was automatically detected once extrusion had been performed in the ETL tool (Figure 23). The representation identifiers, however, had to be set, and in this case, "Body" was set. However, the resultant model is not useful when used as a container for spatial analysis. For instance, using Ifcopenshell, a Python script was created after the GIS IFC was integrated into a BIM model and exported as an IFC file. This script examined an argument that returns all foundation components bounded by one of the three sites. This query was unsuccessful (Figure 24 & Figure 25). The reason for this was not immediately obvious, and seven different iterations of the spatial data conversion script were examined in FME, which did not yield different results. A related challenge was also that the created model was not visible in some IFC viewers. This issue persisted after explicitly using a GeometryCoercer transformer in FME to ensure the geometry representation is set to "Brep". Interestingly, [38] reported the same issue regarding the difficulty in identifying the specific pattern in the software's behaviour when interpreting IFC geometry. Consequently, [38] suggested a reduction in the set of geometry representations in the IFC schema, together with clear constraints on when to use specific geometries to avoid loss of information. This study, to the best of my knowledge, is the first to test the usefulness of GIS BIM models for analysis within native BIM environments, and it corroborates the indefiniteness in identifying specific challenges with geometry representation in IFC data conversions.

```
#84=IFCEXTRUDEDAREASOLID(#83,$,#9,3.);
#85=IFCSHAPEREPRESENTATION(#20,'Body','SweptSolid',(#49,#56,#63,#70,#77,#84));
```

Figure 22 SweptSolid as an IFC shape representation

```
⚠ Could not get bounding box for IfcSpace 0WswDDMhmC9Veg9BxeXpBS: Geometry object has no 'bounds' attribute.
Skipping bounding box check for pile caps as target space 0WswDDMhmC9Veg9BxeXpBS bbox is not available.
```

Figure 23 SweptAreaSolid not recognized as a Bounding Box

```python
# Create geometry for the target IfcSpace
try:
    target_space_geom = ifcopenshell.geom.create_shape(settings, target_space).geometry
    # Check if the geometry object has the 'bounds' attribute before accessing it
    if hasattr(target_space_geom, 'bounds'):
        target_space_bbox = target_space_geom.bounds # (xmin, ymin, zmin, xmax, ymax, zmax)
        print(f"✅ Found IfcSpace with GlobalId: {target_space_guid}, and its geometry created with bounds.")
    else:
        print(f"⚠️ Could not get bounding box for IfcSpace {target_space_guid}: Geometry object has no 'bounds' attribute.")


    # Find and print all QD-EST-PileCap family types bounded within the target space (if bbox is available)
    pile_caps_in_space = []
    if target_space_bbox: # Only proceed if the target space bbox was successfully obtained
        for element in ifc.by_type("IfcElement"):
            if element.Name and "QD-EST-PileCap" in element.Name:
                try:
                    element_geom = ifcopenshell.geom.create_shape(settings, element).geometry
                    if hasattr(element_geom, 'bounds'): # Also check if the element geometry has bounds
                        element_bbox = element_geom.bounds

                        # Bounding box containment test
                        if (element_bbox[0] >= target_space_bbox[0] and element_bbox[3] <= target_space_bbox[3] and
                            element_bbox[1] >= target_space_bbox[1] and element_bbox[4] <= target_space_bbox[4] and
                            element_bbox[2] >= target_space_bbox[2] and element_bbox[5] <= target_space_bbox[5]):
                            pile_caps_in_space.append(element)
                    else:
                         print(f"⚠️ Could not get bounding box for element {element.GlobalId}: Geometry object has no 'bounds' attribute.")

                except Exception as e:
                    # Handle cases where geometry creation might fail for some elements
                    print(f"⚠️ Could not create geometry for element {element.GlobalId}: {e}")


        print(f"✅ Found {len(pile_caps_in_space)} QD-EST-PileCap elements bounded within {target_space_guid}")

        for pc in pile_caps_in_space:
            print(f"Element ID: {pc.GlobalId}, Name: {pc.Name}")
    else:
        print(f"Skipping bounding box check for pile caps as target space {target_space_guid} bbox is not available.")


except Exception as e:
    print(f"⚠️ Could not create geometry for IfcSpace {target_space_guid}: {e}")
```

```
⚠️ Could not get bounding box for IfcSpace 0WswDDMhmC9Veg9BxeXpBS: Geometry object has no 'bounds' attribute.
Skipping bounding box check for pile caps as target space 0WswDDMhmC9Veg9BxeXpBS bbox is not available.
```

Figure 24 Bounding Box analysis, Full Script

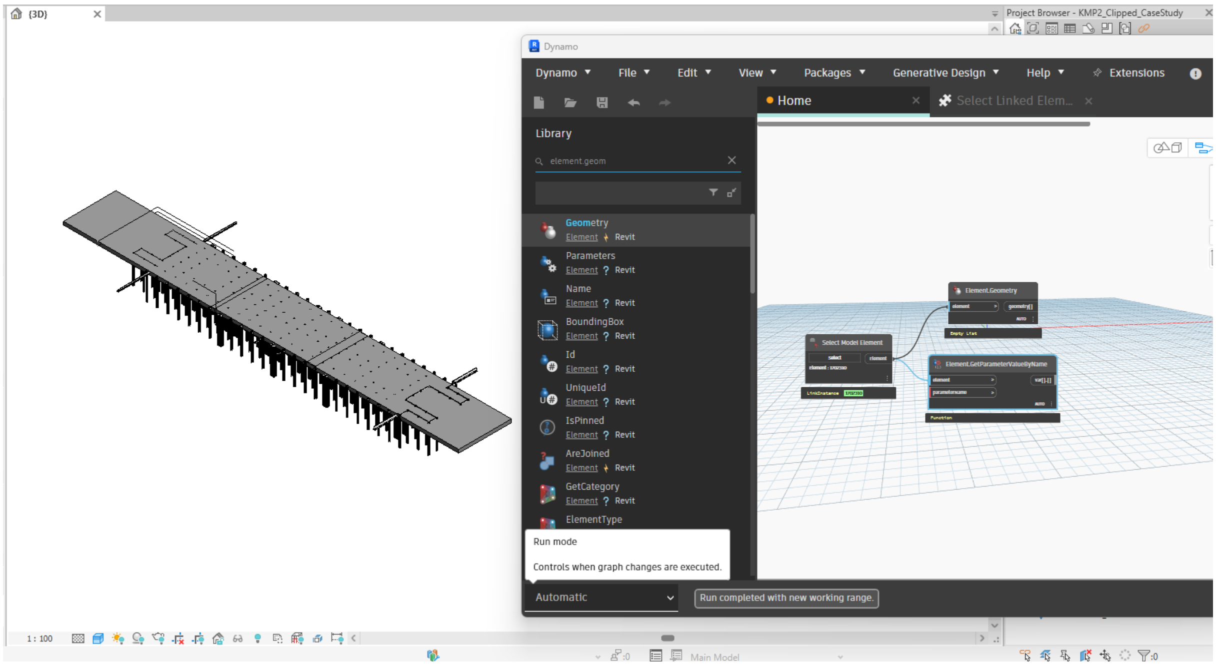


Figure 25 Geomodel IFC imported into Revit

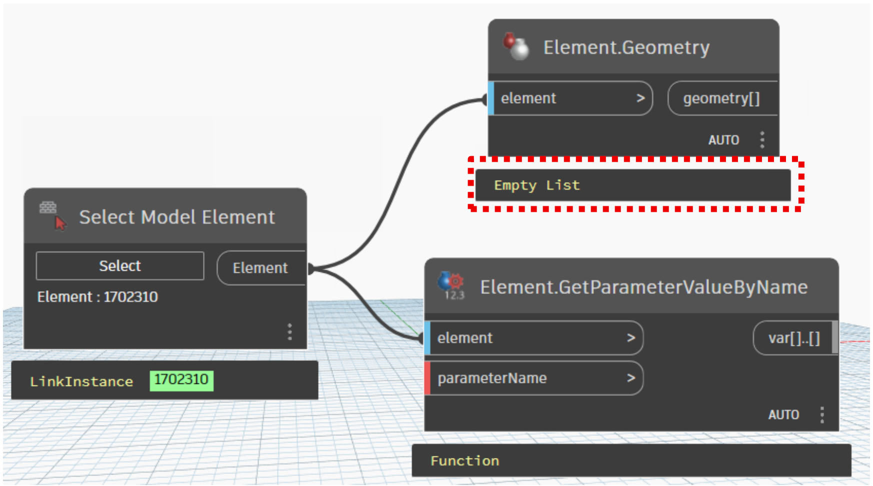


Figure 26 (b) Geomodel IFC Called up in Dynamo

Figure 26 (a) and (b), and Figure 26 show the geomodel loaded into Revit and linked into the project, respectively. In Revit, the linked model loaded successfully, providing much-needed context to the site. This study adopted a smaller scope with a relatively flat surface; however, the complexity of the transformed spatial data can vary from simple to complex. In Figure 24, the linked model is further called up in dynamo for analysis. Though the linked model was successfully called up in dynamo, it could not be processed any further

An important aspect of BIM and GIS integration is map projections and coordinate reference systems. To ensure proper geospatial positioning within both BIM and GIS, their coordinate reference systems must be configured during integration. BIM files use local coordinate systems or Cartesian coordinates in X, Y, and Z format. On the contrary, GIS models use latitude and longitude (Y, X, Z), which are tied to a specific coordinate reference system. Figure 28 (a) and (b) show the geodetic coordinates for the spatial data used for this study in QGIS. After the conversion, this coordinate was saved as an IfcCartesianPoint. IfcCartesianPoint is how local coordinate systems are stored in native modelling tools such as Revit. So basically, the IFC file reads the geodetic coordinate from the GIS model as local coordinates with large X and Y values. This is remedied by creating an IfcCoordinateReferenceSystem that stores up Coordinate Reference Systems. This points out the necessity of clearly defining the intended use of GIS IFC before the start of the workflow in FME in accordance with the specific requirements.

An alternative option is that in Revit, the Project Base Point can be approximated to the real-world coordinate of the specific site. This should be specified in the Exchange Information Requirement (EIR). This option is ideal for preliminary analysis (Figure 29); however, the exact LoGeoRef infrastructure must be specified for real world projects.

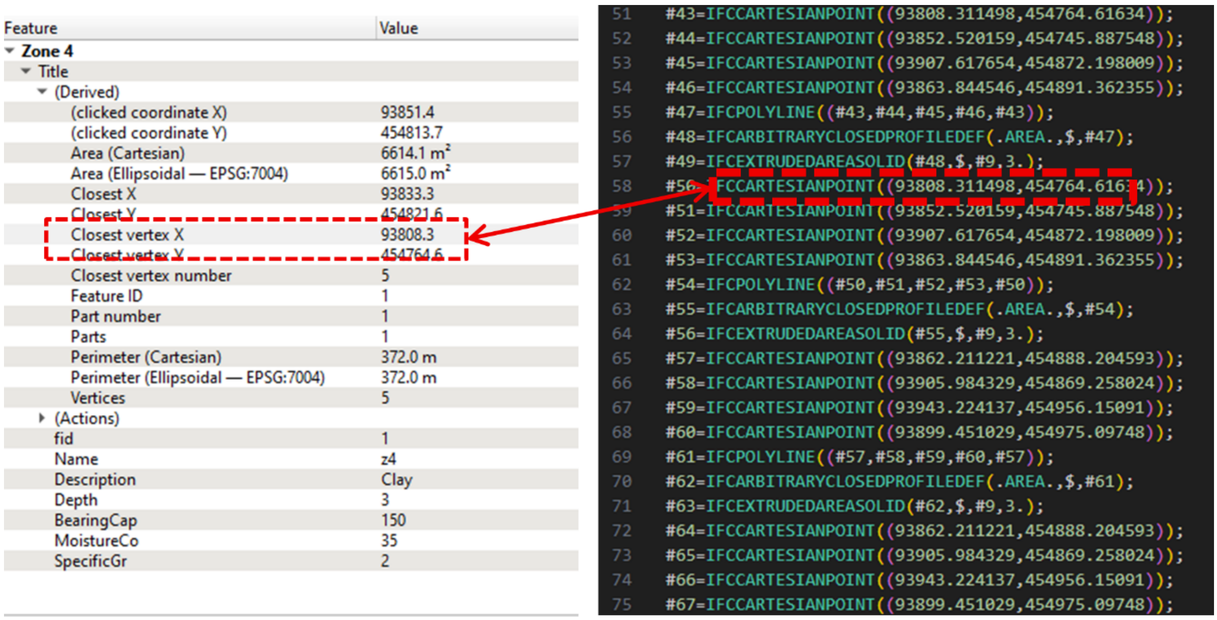


Figure 27 (a) and (b) CRS in QGIS v Local Coordinate System in IFC

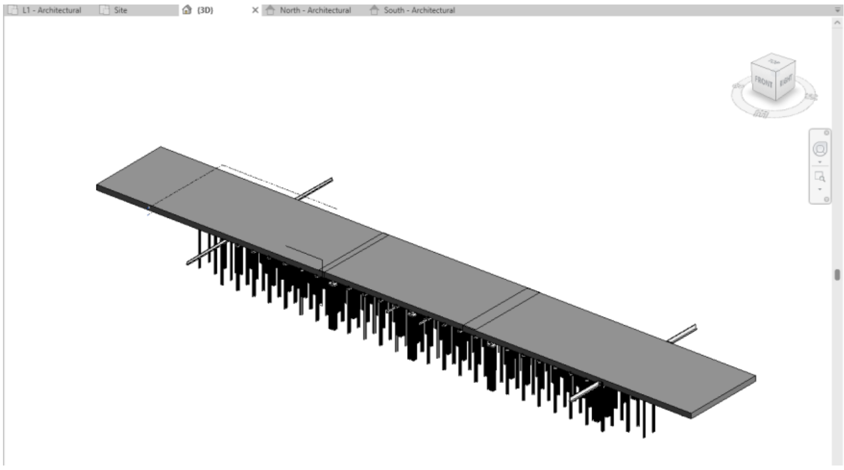


Figure 28 Converted Geospatial Data

4.3 Semantics

Information is the most important aspect of the BIM concept. Therefore, in the integration of spatial data and BIM, the focus of researchers is shifting from geometry to semantics. In the three methods to integrate BIM and GIS, semantic web technologies have proven efficient at retaining and retrieving semantic data, though most of the studies carried out have been at the academic level. The full potential of GIS - BIM approach has not been fully exploited, and this study sought to test that.

The model considered for this study is a site decomposed into three parts, with each zone containing specific attributes. **Error! Reference source not found.** shows how each zone in the geotechnical mode from QGIS has representation in IfcSite. Each zone had its own unique Guid, which facilitates data querying for analytics.

```
#32=IFCSITE('24UL9PS$RWC6pvTbDtmOPw',#24,'Zone6','Clay',$,#31,$,'Zone6',$,$,$,$,$,$,$);
#33=IFCLOCALPLACEMENT($,#16);
#34=IFCSITE('7UzFMxJ7SxidE206Ojz7fg',#24,'Zone5','Clay',$,#33,$,'Zone5',$,$,$,$,$,$,$);
#35=IFCLOCALPLACEMENT($,#16);
#36=IFCSITE('e715yJPRQ5qnss1UpkS35A',#24,'Zone4','Clay',$,#35,$,'Zone4',$,$,$,$,$,$,$);
```

Figure 29 Spatial zones for Geotech converted to IfcSite Class

The spatial decomposition of the case study categorized the three zones as sites nested into a parent site. The IFC property set Pset_SpaceCommon was used to add the desired attributes. Though specific geotechnical properties do not exist in this class, values can be added and interpreted since the property types are all IfcPropertySingleValue. Again, this demonstrates the flexibility of the IFC schema. The various values entered for the geotechnical data, such as bearing capacity, were stored in IfcSinglePropertyValue. IfcSinglePropertyValue defines a single distinct value for an object, which could be numeric or descriptive (**Error! Reference source not found.**). It is then linked to each IfcSpace by IfcRelDefinesProperties. IfcRelDefinesProperties automatically instantiates when property sets are created for objects.

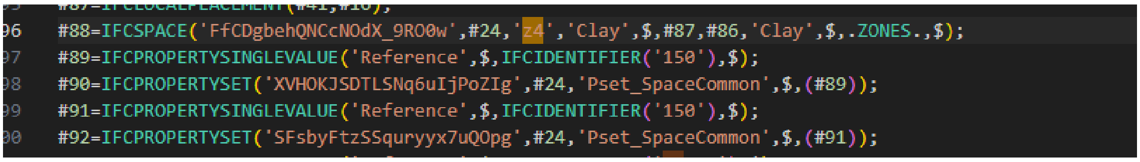


Figure 30 Geotechnical parameters contained in IfcPropertySingleValue

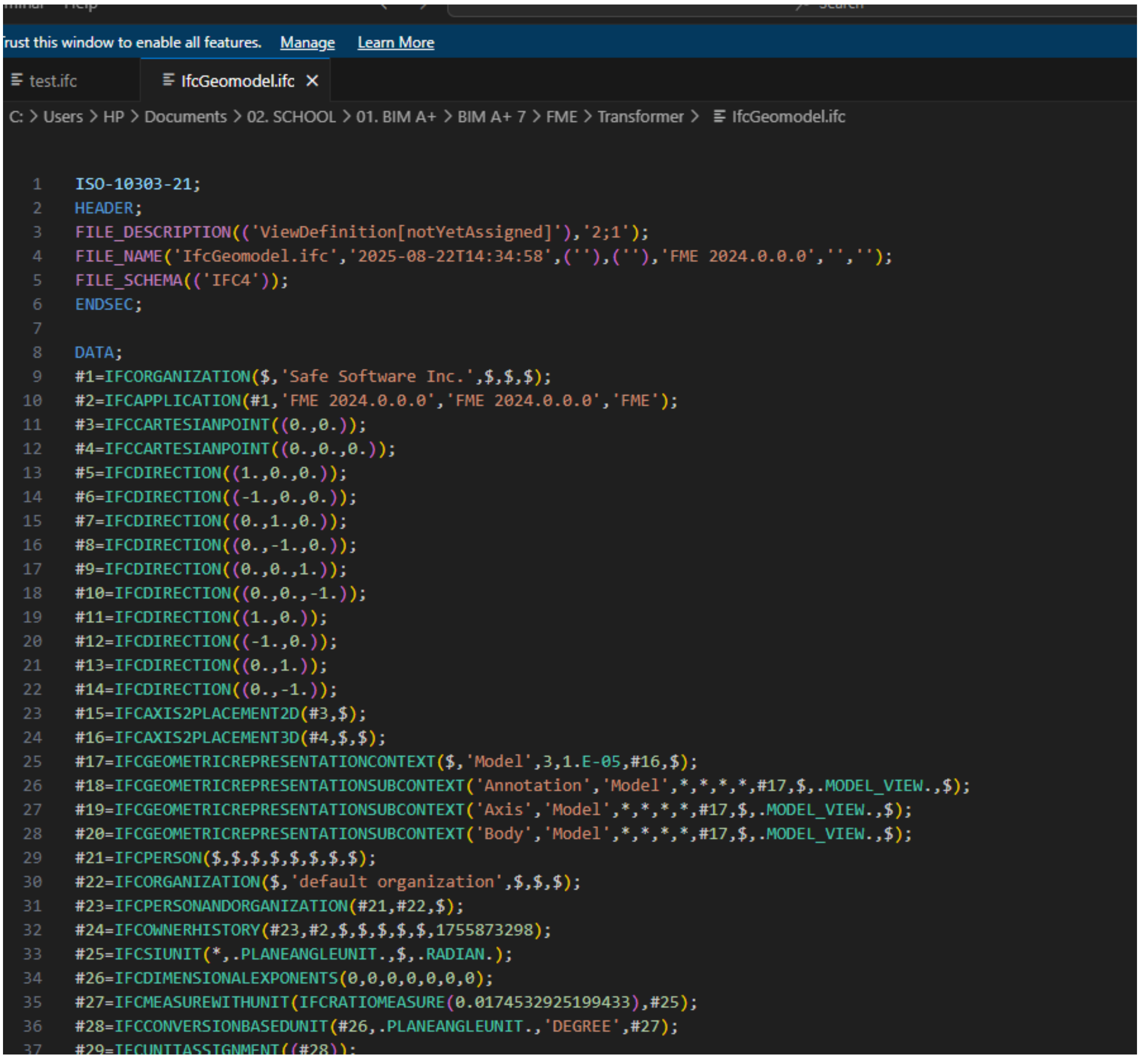


Figure 31 Created IFC STEP File from GIS Model

In summary, the created model observes the IFC hierarchy (Figure 32) and offers deep meaning to the created IFC; however, it has limited ability beyond providing spatial contexts. Integrating spatial data in native modelling environments provides much-needed context for design and design validation. The integrated spatial model conformed to the IFC hierarchy.

5 DISCUSSIONS

Implementing the integration of GIS-BIM is useful for performing a wide range of special analyses, design, and design validation. However, both BIM and GIS are created for different domains and therefore are structured differently. Of the three methods that exist for their integration, GIS-BIM has been the least explored. This option has the potential to provide the necessary spatial context needed for design and design validation in native modelling software.

Geometry in GIS can be exported to IFC if the structure in both domains is understood. Aside from creating geometry from 2D geospatial information, this study examined the capacity of spatial data models to provide semantic meaning, which can be used for design and design validation in BIM. Establishing property sets and their relation to geometry was also achieved; however, performing tasks pertinent to boundary analysis was unsuccessful. The main cause could not be explained even after seven iterations of the FME script. This challenge was observed by [38] who tested converted IFC models in 33 software packages. The case study used an extrusion to add a “Z” parameter to an otherwise flat spatial model. In the IFC schema, objects extruded or extruded along a profile are considered “SweptAreaSolid”.

These results demonstrate the complexity of the IFC schema but also provide a good understanding of its flexibility and aptitude to represent spatial data. The workflow categorizes GIS BIM conversion into four streams: spatial decomposition, geometry, property sets, and property set definition. This template ensures the conversion of GIS information to IFC-compliant data. The study also demonstrated the capacity of Extract, Transform, and Load (ETL) software, specifically, the Feature Manipulation Engine, for interoperability with BIM.

Data filtration during GIS BIM integration was also emphasized. By using the Sampler transformer in FME, the data was sorted in the right order. Hence, the recommended workflow priorities data filtering as an important step in BIM and GIS integration. This reduces information waste, reduces data latency, and storage space requirements.

5.1 Challenges and Limitations

This research attempted to harness the full benefit of IFC for GIS BIM integration. The iterative process to understand how IFC geometry can be created in FME was laborious. It is impossible to detect how different software behaves when reading converted BIM data.

Another challenge was understanding geometrical representation in IFC and how it is output after conversion. The recently created IFC elements in the IFC schema, most of which relate to Civil works, such as IfcGeomodel, are not supported in FME; hence, configuring property sets to IfcGeomodel will not work. However, this also demonstrates the flexibility of IFC since other property sets can be adapted.

5.2 Future Research

This research examined the responsiveness of IFC as a data exchange format for GIS BIM integration, with a focus on geometry and semantics. FME was used as the ETL tool. The integration succeeded; however, the model could not be maximized beyond its geometry. The research establishes the aptitude to adapt IFC for integration; hence, future research should focus on:

1. Testing various IFC shape representations in GIS - BIM transformation to establish the best use cases.
2. A GIS - BIM integration that uses the IFC 5 format for design validation within a native BIM environment.
3. Examine data from various GIS BIM conversion tools within the native BIM environment to test their responsiveness to queries.
4. Creating a workflow that directly links transformed data in ETL tools to native modelling environments.

5.3 Conclusion

This research examined the integration of GIS-BIM using the geopackage file format. Data representation in GIS is vast, and since existing research has not examined this file format, the research extends the thoroughness of GIS-BIM integration efforts. FME was used as the ETL tool, the file comprised three distinct zones transformed into IFC. The spatial decomposition, geometry and semantics were all successfully modelled. The created file was also linked into a BIM native environment.

The created IFC, however, did not respond to semantic queries when examined in Revit. Though the geometry is selectable, it does not offer much aside from providing the spatial context in the native BIM environment.

This research furthers the GIS BIM integration effort, focusing on ordinary 2D planar elements and transforming it into IFC-compliant data. Additionally, the workflow used in this study will provide an important guide to future researchers.

REFERENCES

[1] N. Salheb, K. Arroyo Ohori, and J. Stoter, "AUTOMATIC CONVERSION OF CITYGML TO IFC," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLIV-4/W1-2020, pp. 127–134, Sep. 2020, doi: 10.5194/isprs-archives-XLIV-4-W1-2020-127-2020.

[2] J. Jia, H. Ma, and Z. Zhang, "Integration of Industry Foundation Classes and Ontology: Data, Applications, Modes, Challenges, and Opportunities," *Buildings*, vol. 14, no. 4, p. 911, Mar. 2024, doi: 10.3390/buildings14040911.

[3] G. Celeste, M. Lazoi, M. Mangia, and G. Mangialardi, "Innovating the Construction Life Cycle through BIM/GIS Integration: A Review," *Sustainability*, vol. 14, no. 2, p. 766, Jan. 2022, doi: 10.3390/su14020766.

[4] A. D. Palihakkara, C. A. Osorio-Sandoval, and W. Tizani, "A Review of the Application of Ontologies and Semantic Web for Building Information Modelling and Digital Twins Based Construction Management," *J. Inf. Technol. Constr.*, vol. 30, no. 1, p. 1208, Aug. 2025, doi: 10.36680/j.itcon.2025.049.

[5] Š. Jaud, S. Esser, S. Muhic, and A. Borrmann, "DEVELOPMENT OF IFC SCHEMA FOR INFRASTRUCTURE," Nov. 2020.

[6] The buildingSMART International, "The buildingSMART International Annual Report 2023," The buildingSMART, 9 Quy Court, Colliers Lane, Stow-cum-Quy, Cambridge CB25 9AU, UK, Annual Report, Jun. 2023. Accessed: May 19, 2025. [Online]. Available: https://www.buildingsmart.org/annual-reports/

[7] F. Noardo *et al.*, "Reference study of IFC software support: The GeoBIM benchmark 2019—Part I," *Trans. GIS*, vol. 25, no. 2, pp. 805–841, Apr. 2021, doi: 10.1111/tgis.12709.

[8] W. Mazairac and J. Beetz, "BIMQL – An open query language for building information models," *Adv. Eng. Inform.*, vol. 27, no. 4, pp. 444–456, Oct. 2013, doi: 10.1016/j.aei.2013.06.001.

[9] R. Wang and Y. Tang, "Research on Parsing and Storage of BIM Information Based on IFC Standard," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 643, no. 1, p. 012172, Jan. 2021, doi: 10.1088/1755-1315/643/1/012172.

[10] R. S. Nizam, "A Review of Industry Foundation Class (IFC) Based Data Exchange Model," in *Integrated Building Information Modelling*, P. Wu, H. Li, and X. Wang, Eds., BENTHAM SCIENCE PUBLISHERS, 2017, pp. 158–180. doi: 10.2174/9781681084572117010009.

[11] L. V. Berlo, T. Krijnen, H. Tauscher, T. Liebich, A. V. Kranenburg, and P. Paasiala, "Future of Industry Foundation Classes: towards IFC 5. Proceeding of the Conference CIB W78 2021, Luxembourg," Luxembourg, Oct. 2021. Accessed: Aug. 26, 2025. [Online]. Available: https://www.buildingsmart.org/wp-content/uploads/2021/06/IFC_5.pdf

[12] J. Karlapudi, "A methodology to classify and determine data sharing requirements between OpenBIM models and energy simulation models," 2020. doi: 10.13140/RG.2.2.26167.27041/1.

[13] C. Frey, C. Dahmen, and C. Clemen, "Conception, Validation and Publication of ETL Processes for the Conversion of Geospatial Data for openBIM Projects with FME," May 2024.

[14] "The Evolution of IFC: The path to IFC5," buildingSMART Spain. Accessed: Sep. 09, 2025. [Online]. Available: http://www.buildingsmart.es/2024/12/03/the-evolution-of-ifc-the-path-to-ifc5/

[15] J. J. Cepa, M. G. Alberti, R. M. Pavón, and J. A. Calvo, "Integrating BIM and GIS for an Existing Infrastructure," *Appl. Sci.*, vol. 14, no. 23, p. 10962, Nov. 2024, doi: 10.3390/app142310962.

[16] J. Wu, J. Zhang, and L. Debs, "Model Validation for Automated Building Code Compliance Checking," in *Construction Research Congress 2022*, Arlington, Virginia: American Society of Civil Engineers, Mar. 2022, pp. 640–650. doi: 10.1061/9780784483961.067.

[17] M. J. Sani, I. A. Musliman, and A. Abdul Rahman, "IFC TO CITYGML CONVERSION ALGORITHM BASED ON GEOMETRY AND SEMANTIC MAPPING," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XLVI-4/W3-2021, pp. 287–293, Jan. 2022, doi: 10.5194/isprs-archives-XLVI-4-W3-2021-287-2022.

[18] Z. Gao, K. Ezekwem, and M. Aslam, "An Integration of BIM and GIS for Integrated Project Delivery," Nov. 2020, pp. 505–514. doi: 10.1061/9780784482889.053.
[19] Š. Jaud and C. Clemen, "GeoMVD: the Journey to High-Quality Georeferencing Profiles in IFC Datasets," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. X-4/W5-2024, pp. 203–210, Jun. 2024, doi: 10.5194/isprs-annals-X-4-W5-2024-203-2024.
[20] C. Christian and G. Hendrik, "Level of Georeferencing (LoGeoRef) using IFC for BIM," *J. Geod.*.
[21] Š. Jaud, C. Clemen, S. Muhič, and A. Borrmann, "GEOREFERENCING IN IFC: MEETING THE REQUIREMENTS OF INFRASTRUCTURE AND BUILDING INDUSTRIES," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. X-4/W2-2022, pp. 145–152, Oct. 2022, doi: 10.5194/isprs-annals-X-4-W2-2022-145-2022.
[22] Š. Jaud, A. Donaubauer, and A. Borrmann, "Georeferencing within IFC: A Novel Approach for Infrastructure Objects," in *Computing in Civil Engineering 2019*, Atlanta, Georgia: American Society of Civil Engineers, Jun. 2019, pp. 377–384. doi: 10.1061/9780784482421.048.
[23] B. Pedó *et al.*, "BIM and GIS integration: lessons learned from multiple case studies," presented at the 2023 European Conference on Computing in Construction and the 40th International CIB W78 Conference, Jul. 2023. doi: 10.35490/EC3.2023.248.
[24] A. Tomczak, L. V. Berlo, T. Krijnen, A. Borrmann, and M. Bolpagni, "A review of methods to specify information requirements in digital construction projects," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 1101, no. 9, p. 092024, Nov. 2022, doi: 10.1088/1755-1315/1101/9/092024.
[25] K. Arroyo Ohori, A. Diakité, T. Krijnen, H. Ledoux, and J. Stoter, "Processing BIM and GIS Models in Practice: Experiences and Recommendations from a GeoBIM Project in The Netherlands," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 8, p. 311, Aug. 2018, doi: 10.3390/ijgi7080311.
[26] J. J. Cepa, R. M. Pavon, M. G. Alberti, and P. Carames, "Towards BIM-GIS Integration For Road Intelligent Management System," *J. Civ. Eng. Manag.*, vol. 29, no. 7: 621-638, May 2023, [Online]. Available: https://oa.upm.es/78886/1/19514-Article%20Text-78900-2-10-20230911.pdf
[27] J. Zhu and P. Wu, "Towards Effective BIM/GIS Data Integration for Smart City by Integrating Computer Graphics Technique," *Remote Sens.*, vol. 13, no. 10, p. 1889, May 2021, doi: 10.3390/rs13101889.
[28] J. Oraskari, S. Törmä, M. Yalcinkaya, V. Singh, and J. Beetz, "Sharing Linked Building Data in a Peer-to-Peer Network: ifcOWL Meets InterPlanetary File System," *J. Comput. Civ. Eng.*, vol. 38, no. 1, p. 04023037, Jan. 2024, doi: 10.1061/JCCEE5.CPENG-5381.
[29] B. Bazzanella and Y. Tzitzikas, "Interoperability Objectives and Approaches (APARSEN Deliverable D2501)," *APARSEN Deliv.*, Feb. 2013.
[30] M. Aydın, "A Review of BIM-Based Automated Code Compliance Checking: A Meta-Analysis Research," in *Automation and Control - Theories and Applications*, E. P. Dadios, Ed., IntechOpen, 2022. doi: 10.5772/intechopen.101690.
[31] P. Bonsma, T. Zayakova, S. Varbanov, I. Bonsma, K. A. Ohori, and J. Stoter, "Change Toolkit for Digital Building Permit," Oct. 2024.
[32] K. Zima and E. Mitera-Kiełbasa, "Level of Information Need for BIM Models: Australia, New Zealand and ISO 19650," *Civ. Environ. Eng. Rep.*, vol. 32, no. 4, pp. 1–3, Dec. 2022, doi: 10.2478/ceer-2022-0041.
[33] H. Qi *et al.*, "Research on the 3D Fine Modeling Method of In-Service Road," *Adv. Civ. Eng.*, vol. 2023, pp. 1–14, Jun. 2023, doi: 10.1155/2023/7336379.
[34] H. Li and J. Zhang, "Improving IFC-Based Interoperability between BIM and BEM Using Invariant Signatures of HVAC Objects," *J. Comput. Civ. Eng.*, vol. 37, no. 2, p. 04022059, Mar. 2023, doi: 10.1061/(ASCE)CP.1943-5487.0001063.
[35] N. Babazadeh, J. Teizer, H.-J. Bargstädt, and J. Melzner, "Predictive simulation of construction site noise emissions from heavy equipment," *Smart Sustain. Built Environ.*, vol. 14, no. 3, pp. 715–739, Apr. 2025, doi: 10.1108/SASBE-08-2023-0226.
[36] M. Yaman, "Application of Acoustic Analysis with BIM and Prediction of Reverberation Time," *Archit. Civ. Eng. Environ.*, vol. 18, no. 1, pp. 87–95, Mar. 2025, doi: 10.2478/acee-2025-0007.
[37] C. Sydora and E. Stroulia, "Rule-based compliance checking and generative design for building interiors using BIM," *Autom. Constr.*, vol. 120, p. 103368, Dec. 2020, doi: 10.1016/j.autcon.2020.103368.

[38] J. Peng and X. Liu, "Automated code compliance checking research based on BIM and knowledge graph," *Sci. Rep.*, vol. 13, no. 1, p. 7065, May 2023, doi: 10.1038/s41598-023-34342-1.
[39] F. Noardo *et al.*, "Reference study of IFC software support: The GeoBIM benchmark 2019—Part I," *Trans. GIS*, vol. 25, no. 2, pp. 805–841, Apr. 2021, doi: 10.1111/tgis.12709.