

Univerza v Ljubljani
Fakulteta *za gradbeništvo*
in geodezijo



CHAMINDU PIYATHILAKE

DIGITIZATION OF WORKFLOW ON MANUFACTURE AND
ASSEMBLY ELEMENTS INTEGRATED FAÇADE
RENOVATION

DIGITALIZACIJA POTEKA DELA PRI PRENOVI FASAD
Z UPORABO
PROIZVODNIH IN MONTAŽNIH ELEMENTOV



European Master in
Building Information Modelling

Master thesis No.:

Supervisor: Assist. Prof. Matevž Dolenc, Ph.D.

Ljubljana, 2023



ERRATA

Page	Line	Error	Correction
-------------	-------------	--------------	-------------------

»This page is intentionally blank«

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK:	005.922.52:692.23(043.3)
Avtor:	Chamindu Piyathilake
Mentor:	doc. dr. Matevž Dolenc
Somentor:	Dr. Dietmar Siegele
Naslov:	Digitalizacija poteka dela pri prenovi fasad z uporabo proizvodnih in montažnih elementov
Tip dokumenta:	Magistrsko delo
Obseg in oprema:	60p., 71 fig.
Ključne besede:	BIM, Façade renovation, Manufacture and Assembly, Exchange information requirement (EIR), openBIM, open-source, Point-cloud, IFC, Collaboration, Blender, BlenderBIM, Open3D, Ifcopenshell

Izvleček:

Informacijsko modeliranje gradenj (BIM) preoblikuje gradbeno industrijo z digitalno usmerjenimi pristopi, sodelovanjem in standardizacijo, ki so usmerjenimi v celoten življenjski cikel grajenega sredstva. Eden ključnih vidikov procesa digitalizacije je raziskovanje delovnih tokov z namenom izboljšanja učinkovitosti izmenjave podatkov, standardizacije ter uvajanja odprto-kodnih tehnologij.

Raziskavalna naloga, z uporabo študije primera, obravnava integriran delotok prenove fasade z izdelavo in montažo fasadnih elementov. Naloga v prvem delu opisuje obstoječe procese oz. delotoke ter začrta smernice novih digitaliziranih procesov, ki so v osrednjem delu naloge implementirani z uporabo pristopa openBIM ter različnih odprto-kodnih programskih rešitev.

OpenBIM in uporaba odprto-kodnih programskih rešitev predstavljata eno izmed smeri izboljšanja sodelovanja in izmenjave podatkov v obravnavanih BIM procesih. Odprto-kodna programska oprema uporabljena v raziskavi, npr. Blender z dodatkom BlenderBIM, Ifcopenshell, idr. predstavljajo zanimivo alternativo lastniški programski opremi in odpira nove možnosti digitalizacije gradbeništva.

»This page is intentionally blank«

BIBLIOGRAPHIC– DOKUMENTALISTIC INFORMATION AND ABSTRACT

UDC:	005.922.52:692.23(043.3)
Author:	Chamindu Piyathilake
Supervisor:	Assist. Prof. Matevž Dolenc, PhD.
Cosupervisor:	Dr. Dietmar Siegele
Title:	Digitization of workflow on manufacture and assembly elements integrated façade renovation
Document type:	Master thesis
Scope and tools:	60p., 71 fig.
Keywords:	BIM, Façade renovation, Manufacture and Assembly, Exchange information requirement (EIR), openBIM, open-source, Point-cloud, IFC, Collaboration, Blender, BlenderBIM, Open3D, Ifcopenshell

Abstract:

Building Information Modelling (BIM) is profoundly transforming the Architecture, Engineering, and Construction (AEC) industry sector characterized by digital-driven approaches, enhanced collaboration, and standardization targeted towards the full life cycle of a built asset. One key aspect of this digitization paradigm is to examine prevailing workflows in the industry to improve data flow efficiency, foster standardization, and make technology accessible and open to stakeholders.

This research focuses on the manufacturing and assembly elements integrated within the facade renovation workflow with a case study application. The research commences by employing process models which are used to capture existing and proposed workflows that provide the foundation for the digitization strategy. The practical implementation of the strategy is explored using openBIM and open-source approaches. This includes generating Exchange Information Requirements, point cloud to BIM process automation, and collaboration guidelines.

openBIM and the use of open-source software solutions represent one of the directions for improving cooperation and data exchange in the discussed BIM processes. Open-source software used in the research, e.g. Blender with BlenderBIM plugin, Ifcopenshell, etc. presents an interesting alternative to proprietary software and opens up new possibilities for the digitization of the construction industry.

»This page is intentionally blank«

ACKNOWLEDGEMENTS

I want to thank BIM A+ Consortium for the exposure and knowledge shared on Building Information Modelling. The experience has taught me a lot and driven my passion for digital-driven design and methodologies.

Thank you Assist. Prof. Matevž Dolenc and dr. Tomo Cerovšek for the support through the thesis.

Thank you, Dr. Dietmar Siegele and Orjola Braholi from Fraunhofer Italia for providing me with a case study and support for the thesis.

Thank you to the OSArch community, Blender and BlendeBIM community, and the open-source community for sharing knowledge and support.

Thank you, Manu Nirasha, and Pasindu Epa for your time and support.

Thank you for the great time and memories from my friends in BIM A+.

»This page is intentionally blank«

TABLE OF CONTENTS

ERRATA.....	I
BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK	IV
BIBLIOGRAPHIC– DOKUMENTALISTIC INFORMATION AND ABSTRACT	VI
ACKNOWLEDGEMENTS.....	VIII
TABLE OF CONTENTS.....	X
INDEX OF FIGURES.....	XII
1 INTRODUCTION.....	1
1.1 Renovation in BIM.....	2
1.1.1 Renovation Using Manufacture and Assembly Elements	2
1.1.2 BIM and Facade Renovation.....	3
1.2 Problem Statement	3
1.3 Research Objectives	4
1.4 Methodology	5
2 Façade Renovation in BIM.....	6
2.1 Process Model for Current Workflow	7
2.1.1 Data Acquisition and Pre-processing	7
2.1.2 Scan to BIM	11
2.1.3 Design for Manufacture and Assembly.....	12
2.2 Information Requirement and Exchange.....	14
2.3 Summary	16
3 OPEN SOURCE AND OpenBIM	18
3.1 OpenBIM.....	18
3.2 Open-Source BIM Platforms.....	21
3.2.1 FreeCAD	21
3.2.2 Speckle	22
3.2.3 Blender and BlenderBIM	23

3.3	Useful tools and software	25
3.3.1	CloudCompare.....	25
3.3.2	Ifcopenshell	25
3.3.3	Open3D.....	26
3.4	Summary.....	27
4	CASE STUDY	28
4.1	Introduction	28
4.1.1	Case Study Overview	29
4.1.2	Software and Tools.....	29
4.2	Developing Framework.....	30
4.2.1	TO-BE Model.....	30
4.2.2	Digitization Strategy.....	33
4.3	Proposed Workflow.....	34
4.3.1	Exchange Information Requirement.....	34
4.3.2	Point Cloud to BIM	38
4.3.3	Collaboration.....	48
4.4	Results	50
4.5	Discussion.....	50
5	CONCLUSION AND FUTURE DEVELOPMENT	52
6	ACRONYMS AND APPLICATION	54
7	REFERENCES	57

INDEX OF FIGURES

<i>Figure 1: Processes, roles, data, and tools are the key elements of an automatable workflow (Borrmann et al., 2018)</i>	1
<i>Figure 2: Events hall in Romainville by EMBT and Ilimeigo architects using prefabricated concrete renovation with an addition of perforated aluminum façade (Floornature, 2021)</i>	2
<i>Figure 3: Manual creation of as-built BIM with cross sections and surface extrusions (Tang et al., 2010a)</i>	4
<i>Figure 4: Research Methodology (Author)</i>	5
<i>Figure 5: Overview of the IDM/MVD method used for the IFC-based exchange of information (Borrmann et al., 2018)</i>	6
<i>Figure 6: AS-IS process model for facade renovation using manufacture and assembly elements (Author)</i>	7
<i>Figure 7. The proposed application-oriented scan-to-BIM framework (Q. Wang et al., 2019).</i>	8
<i>Figure 8: Examples of three types of laser scanners according to working platforms (Q. Wang et al., 2020)</i>	8
<i>Figure 9: Terrestrial laser scanning (Bormann et al., 2018)</i>	9
<i>Figure 10: Working principles of the two methods, time of flight and phase shift (Q. Wang et al., 2020)</i>	9
<i>Figure 11: Processing of point cloud data (Q. Wang et al., 2020)</i>	10
<i>Figure 12: Segmenting of scan and recognition of individual desks (Q. Wang et al., 2019)</i>	11
<i>Figure 13: Labor productivity index for on-site construction, off-site construction, and manufacturing in United States market (Sacks et al., 2018)</i>	12
<i>Figure 14: Stages of CAM process (Borrmann et al., 2018)</i>	13
<i>Figure 15: St. Joseph Hospital in Denver, US (Sacks et al., 2018)</i>	13
<i>Figure 16: Optioneering flow (Alfieri et al., 2020)</i>	14
<i>Figure 17: BIM Maturity stages and information management (International Organization for Standardization, 2018)</i>	14
<i>Figure 18: Hierarchy of information requirement (International Organization for Standardization, 2018)</i>	15
<i>Figure 19: Summary of a scan to BIM framework (Q. Wang et al., 2019)</i>	16
<i>Figure 20: IFC data model (buildingSMART, n.d.)</i>	18
<i>Figure 21: Part of the inheritance hierarchy and inheritance hierarchy of a wall (Borrmann et al., 2018)</i>	19
<i>Figure 22: Solid modeling in IFC data model (Borrmann et al., 2018)</i>	20

<i>Figure 23: Boundary representation with IfcFacetedBrep for advanced flat surface model representation (buildingsSMART, n.d.)</i>	20
<i>Figure 24: FreeCAD interface (FreeCAD, n.d.)</i>	21
<i>Figure 25: Speckle Web interface (Speckle, n.d.)</i>	22
<i>Figure 26: Speckle connectors for Revit, Rhino, and web viewing (Speckle, n.d.)</i>	22
<i>Figure 27: Assigning class and placing in spatial tree in BlenderBIM (BlenderBIM, n.d.)</i>	23
<i>Figure 28: Geometry node examples in Blender (Blender, n.d.)</i>	23
<i>Figure 29: Native IFC authoring of BlenderBIM vs traditional openBIM (BlenderBIM, n.d.)</i>	24
<i>Figure 30: Geometry in IFC and scheduling in BlenderBIM (BlenderBIM, n.d.)</i>	24
<i>Figure 31: Meshing and distancing operations in CloudCompare (CloudCompare, n.d.)</i>	25
<i>Figure 32: Sample code to get all wall types with IfcOpenShell (IfcOpenShell, n.d.)</i>	25
<i>Figure 33: Python console and text editor functions in Blender to run IfcOpenShell scripts (IfcOpenShell, n.d.)</i>	26
<i>Figure 34: Example of code to visualize a point cloud (Open3D, n.d.)</i>	26
<i>Figure 35: Example of voxel down sampling operation on a point cloud (Open3D, n.d.)</i>	27
<i>Figure 36: Methodology for case study application (Author)</i>	28
<i>Figure 37: Case study point cloud viewed in CloudCompare (Author)</i>	29
<i>Figure 38: Open-source and openBIM software and tools for case study (Author)</i>	29
<i>Figure 39: Proposed TO-BE process model (Author)</i>	30
<i>Figure 40: A1 (3D building survey) sub-process model (Author)</i>	30
<i>Figure 41: A2 (Automated as-built BIM) sub process model (Author)</i>	31
<i>Figure 42: A3 (Facade design) sub process model (Author)</i>	31
<i>Figure 43: A4 (Manufacture and Assembly) sub processes part 1 (Author)</i>	32
<i>Figure 44: A4 (Manufacture and Assembly) sub processes part 2 (Author)</i>	32
<i>Figure 45: Digitization strategy for the case study (Author)</i>	33
<i>Figure 46: EIR for the proposed workflow divided to two parts (Author)</i>	34
<i>Figure 47: Part 1 of the EIR for the proposed workflow (Author)</i>	35
<i>Figure 48: Part 2 of EIR for the proposed workflow (Author)</i>	37
<i>Figure 49: Natively supported file formats for Blender such as PLY, SVG, OBJ (Author)</i>	38
<i>Figure 50: Atomic Blender addon details (Author)</i>	39
<i>Figure 51: Details of Photogrammetry Import Export addon for Blender (Author)</i>	39
<i>Figure 52: Cloud sub sampling in CloudCompare (Author)</i>	40
<i>Figure 53: Converted point cloud imported to Blender (Author)</i>	41
<i>Figure 54: Geometry node tree to create a mesh from points in Blender (Author)</i>	41
<i>Figure 55: From top left to right and bottom left to right - 1. Points to volume, 2. Volume to mesh, 3. Boolean intersection with plane, 4. points containing the intersection (Author)</i>	42
<i>Figure 56: Creating an IFC in Blender with BlenderBIM (Author)</i>	43

<i>Figure 57: IFC spatial tree in BlenderBIM (Author)</i>	43
<i>Figure 58: IFC project with grids and levels (Author)</i>	44
<i>Figure 59: Manually drawn wall and windows in BlenderBIM and viewed in Open IFC Viewer (Author)</i>	44
<i>Figure 60: IFC project containing parametric beam system created using scripting (Author)</i>	45
<i>Figure 61: IFC project containing slabs, grids and levels created using scripting (Author)</i>	45
<i>Figure 62: Open3D code used to visualize, planar segment, and write both line and mesh files with planes (Author)</i>	46
<i>Figure 63: Planar patch visualization of case study (Author)</i>	47
<i>Figure 64 : Mesh (top) and point files (bottom) viewed Blender, contains information on 24 patches (Author)</i>	47
<i>Figure 65: IFC Git in BlenderBIM and linking to remote repository (Author)</i>	48
<i>Figure 66: Showing changes to IFC (Author)</i>	48
<i>Figure 67: Adding remote repository information (Author)</i>	49
<i>Figure 68: Remote repository in GitHub (Author)</i>	49
<i>Figure 69: From detected planes of point cloud to wall creation (Author)</i>	51
<i>Figure 70: Point cloud operation process followed in the research (Author)</i>	52
<i>Figure 71: Workflow for point cloud to BIM using open-source and openBIM for future research (Author)</i>	53

1 INTRODUCTION

The current Architecture, Engineering, Construction, Operation, and Facility Management (AECO / FM) industry is benefitting from the use of Building Information Modeling (BIM) to mitigate issues such as unanticipated costs, fragmentation in project teams, organizational boundaries, time spent on critical analysis of energy, structural, time schedules, and cost (Sacks et al., 2018). It is useful to identify BIM as the uninterrupted use of digital building models throughout the complete life cycle of a constructed facility (Borrmann et al., 2018). During the life cycle of a built facility information exchange happens frequently between different individuals, organizations, and teams resulting in workflows. A workflow comprises processes, roles, data, and tools (Borrmann et al., 2018).

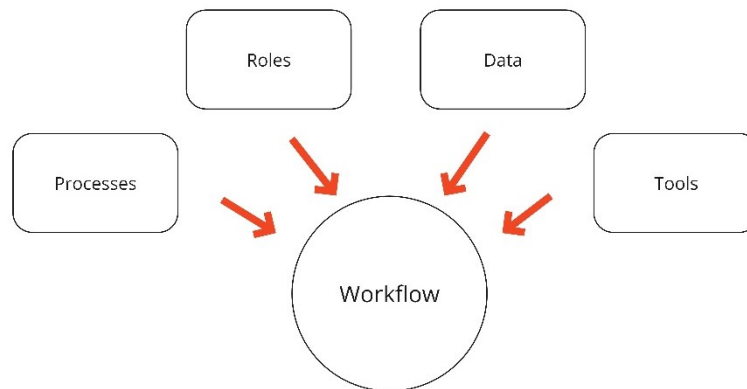


Figure 1: Processes, roles, data, and tools are the key elements of an automatable workflow (Borrmann et al., 2018)

A workflow (*Figure 1*) is a scenario in which planners and participator companies exchange information in a transparent and structured manner (Borrmann et al., 2018). It is crucial to look at these workflows in the practical applications of BIM to digitalize the Architecture, Engineering, and Construction (AEC) industry ensuring information flow throughout the life cycle of a built asset.

With a case study application, this research aims to analyze facade renovation workflow using manufacture and assembly elements from a digitization point of view. Initially, key aspects of an existing workflow in the practice are identified using process models. Moreover, this research aims to introduce a digitization strategy to implement in the case study which will focus on the openBIM approach.

openbim is primarily a vendor-neutral approach to collaboration, and management for BIM practice using formats like Industry Foundation Class (IFC), BIM Collaboration Format (BCF), MVD (Model View Definition), and Information Delivery Manual (IDM). This approach ensures interoperability, open and neutral standards, reliable data exchanges, collaborative workflows, flexibility for stakeholders, and sustainability (openBiM, 2020).

1.1 Renovation in BIM

Renovation constitutes a pivotal component within the AECO / FM industry and compels us to consider a building throughout its entire life cycle phases. It is useful to look at renovation as a composition of terms such as retrofit, rehabilitation, restoration, refurbishment, and renovation (Joblot et al., 2017). Existing research towards renovation comprises topics such as proposing frameworks, systems, and tools for BIM adaptations in renovation, LCAs (Life Cycle Assessment), and sustainable renovations (Daniotti et al., 2021; Joblot et al., 2017; Kamari et al., 2018). The applications within the industry continually arise due to the increasing awareness of imperatives such as consciousness about energy efficiency, occupant safety and health, building maintenance, and change of uses and functions. Within the European Union (EU), the segment of aged building stock (when a building is over 50 years old) accounts for 35% of the existing building stock and is responsible for 40% of the energy consumption attributed to the built environment (Kamari et al., 2018). Hence it is vital to focus on the use of BIM in the Renovation sector. Research on this topic has proliferated across diverse applications.

1.1.1 Renovation Using Manufacture and Assembly Elements

Design for Manufacture and Assembly (DfMA) emerges from combining two ideas, Design for Manufacture (DfM) and Design for Assembly (DfA) which explains an approach to designing elements considering manufacturability and assemblability during the construction stage (Said et al., 2017). Considering DfMA through the perspective of offsite prefabrication unveils its utility in enabling customization, manufacturability, assemblability, and seamless integration of designers and technicians in the design stage (Said et al., 2017; Yuan et al., 2018a). Within the industry common applications of DfMA elements comprise facade renovation, roof and ceiling fit-outs, and interior designs including partition walls, furniture, and decorations. Out of these uses, facade renovations have been widely practiced in the industry for reasons such as energy efficiency and retrofitting (*Figure 2*).

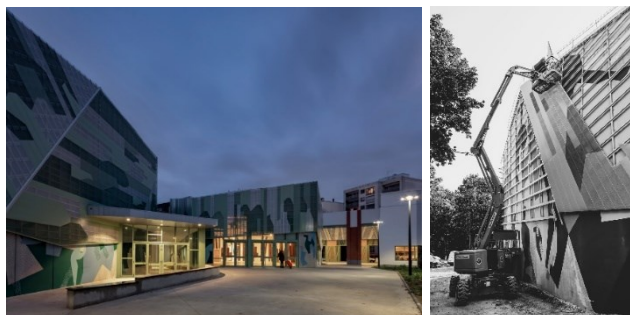


Figure 2: Events hall in Romainville by EMBT and Ilimelgo architects using prefabricated concrete renovation with an addition of perforated aluminum facade (Floornature, 2021)

1.1.2 BIM and Facade Renovation

Given the majority of existing buildings lack a BIM model representation, creating a façade BIM model becomes a necessary task (Chen et al., 2022a). The common workflow would start with capturing the three-dimensional (3D) data of the façade. Presently, Light detection and ranging (LiDAR) is commonly employed, including techniques like mobile laser scanning (MLS), and terrestrial laser scanning (TLS). Among these, TLS has emerged as the predominant method for capturing point clouds of building facades due to its notable benefits, such as exceptional integrity, dense data coverage, remarkable accuracy, and versatile station setup (Achakir et al., 2021; Chen et al., 2022b; Dong et al., 2018; Tang et al., 2010a; C. Wang et al., 2015).

After the processing of point cloud data, the next step would be generating the façade BIM model. There are several research in this area as the generation of the BIM model from point cloud is often a manual and labor-intensive task (Tang et al., 2010a; Q. Wang et al., 2019, 2020).

In the industry, there are programs such as CloudCompare, Trimble Realworks, Autodesk Recap, and Faro Scene Software to process point clouds. Manufacturers and suppliers also have their software and addons specifically catered to their products such as “Open BIM RenoZEB - Focchi Facades” which is a plug-and-play tool developed to layout Focchi façade panels for any renovation project (*Funding & Tenders*, n.d.).

Furthermore, in this workflow, it is useful to study the 3D scanning requirements, surveying strategies and technologies, and essentials to consider facilitating Manufacture and Assembly requirements. These aspects are subject to the type of renovation such as total renovation with new panels, replacing a panel, maintenance involving Heating Ventilation, and Air Conditioning (HVAC), and context-specific aspects such as the health of the façade and identifying critical parts.

1.2 Problem Statement

The presented case study provides a point cloud data set of a residential building where the intention is to design a façade through the incorporation of manufacture and assembly elements. The current workflow uses the processed point cloud to generate a mesh model by manually modeling and designing the façade accordingly using Autodesk Revit. The mesh model does not include information that could be useful for BIM processes and does not exchange information for BIM applications. Generating a BIM model out of point cloud has been researched several times since the process is typically a manual, labor-intensive (*Figure 3*), and closed BIM process (Tang et al., 2010a; Q. Wang et al., 2019, 2020). A closed BIM process is when proprietary software is used with their standards and file formats for the exchange of information that facilitates the use of that respective software only.

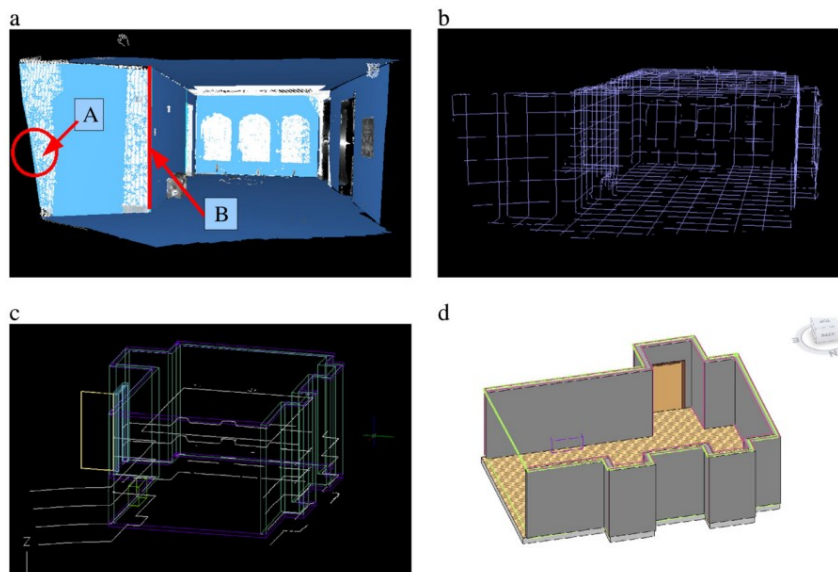


Figure 3: Manual creation of as-built BIM with cross sections and surface extrusions (Tang et al., 2010a)

Furthermore, in the façade renovation workflow, the information needed from the manufacture and assembly team for the design team is not transferred. This results in project delays and cost increases when the façade design may have a major design change requested by the manufacture and assembly team. This scenario is a result of not having an information requirement and information exchange for the project connecting the design and construction stages.

Additionally, the workflow involving surveyors, technicians, architecture and engineering teams, and designers within the context of manufacture and assembly is not in seamless collaboration and remains fragmented. This situation occurs because of the lack of definition in the BIM standardization, guidelines for collaboration, Common Data Environment (CDE), tools, and methodologies in BIM implementation.

1.3 Research Objectives

The objectives of this research can be articulated through the following key aspects.

- To identify the processes, actors, data, and tools in the façade renovation workflow using manufacture and assembly elements.
- Generate a digitization strategy for the proposed workflow and identify information requirements and information exchange.
- Explore the use of openBIM standards and open-source software as a BIM implementation strategy.

1.4 Methodology

The methodology used in this research is demonstrated in the *Figure 4* diagram. This process map is useful to identify the stages of this workflow, its current actors, and its subprocesses as well.

While Chapter 1 provides a background to the research area, Chapter 2 introduces the existing literature findings according to an AS-IS process model of the current workflow. Furthermore, Chapter 2 explains the information requirement and information exchange with aspects to consider in the practical application of façade renovation.

The third chapter investigates the openBIM standards mainly focusing on the Industry Foundation Class. It also comprises open-source software for BIM authoring, collaboration, and other useful tools that could be useful in the façade renovation workflow.

The fourth chapter explains the case study information and TO-BE process model that is useful for generating the digitization strategy for the proposed workflow and case study application.

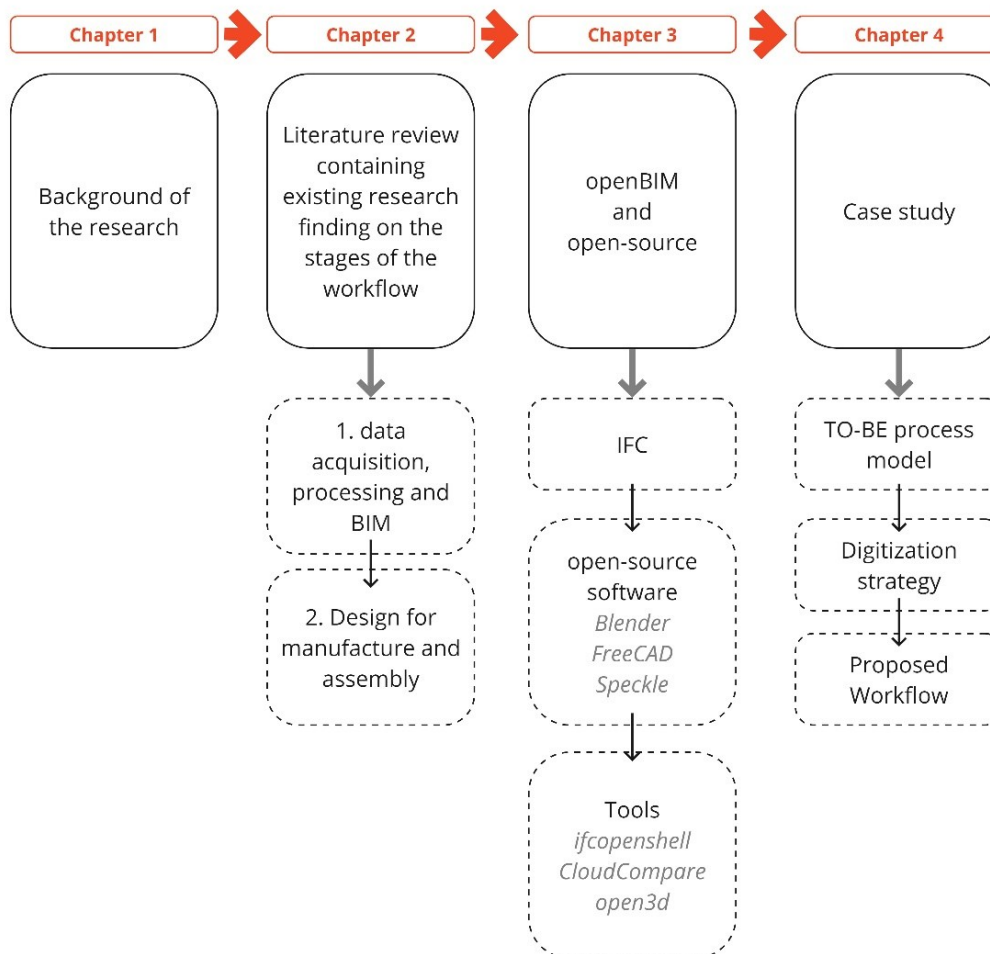


Figure 4: Research Methodology (Author)

2 FAÇADE RENOVATION IN BIM

Martinez et al. (2015) state that the emergence of initiatives within both the US and Europe envisions extensive building renovations in the future with a focus on reducing the consumption of energy and carbon emissions. Furthermore, the necessity for façade renovation arises in various scenarios such as fixing failures, code and energy compliance, or aesthetic purposes (Martinez et al., 2015).

To approach this complex scenario with an openBIM approach and generate IFC-based collaboration this chapter investigates a workflow that could be followed in BIM.

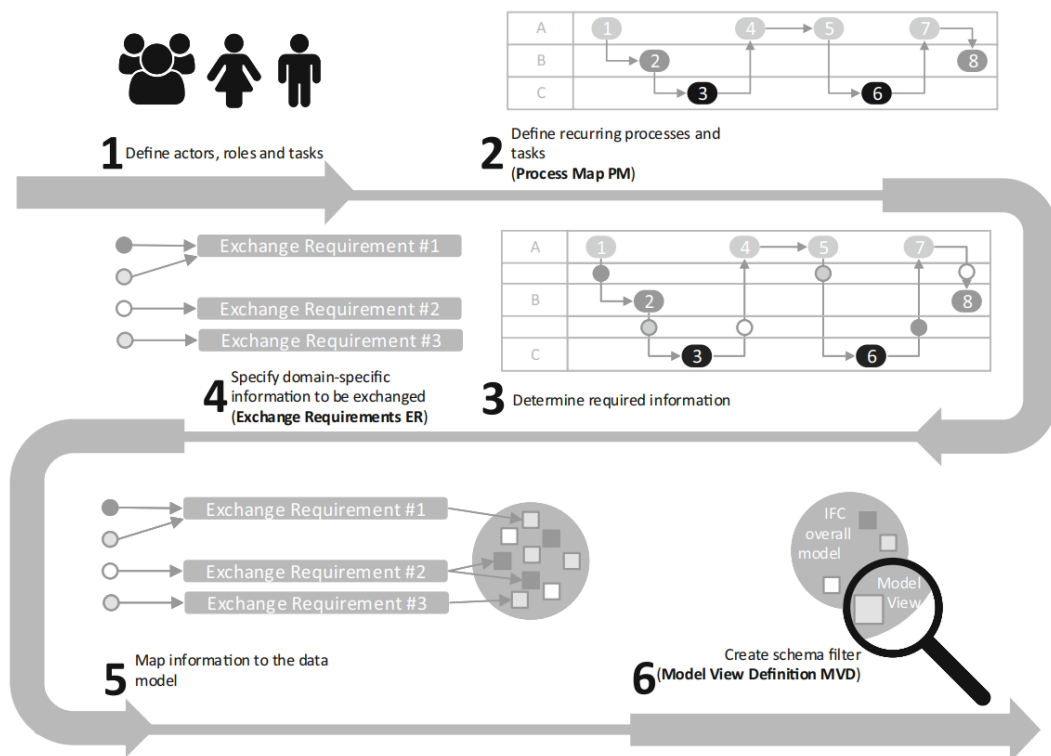


Figure 5: Overview of the IDM/MVD method used for the IFC-based exchange of information (Borrmann et al., 2018)

Figure 5 shows the relationship between defining actors, roles, and tasks at the beginning of a project, process maps, and exchange information requirements as the basis workflow for IFC-based collaboration. The goal is to define a Model View Definition based on the required information (Borrmann et al., 2018). This research will follow these steps starting from standardized formalized process maps with Business Process Model Notation (BPMN) to create processes and sub-processes for façade renovation workflow integrating manufacture and assembly elements.

Section 2.1 elaborates on the process model for this research, AS-IS, and investigates existing research on the findings.

2.1 Process Model for Current Workflow

In the façade renovation workflow using manufacture and assembly elements, we could identify four main steps (*Figure 6*). The workflow would start with the acquisition of geometric data (Surveying) to create the existing building model (As-Built BIM or Mesh) where this model would be used next to host the façade design (Design stage). The fourth and final stage is fabrication of the elements and installation (Manufacture and Assembly stage) onsite which would lead to the start of the maintenance or use stage.

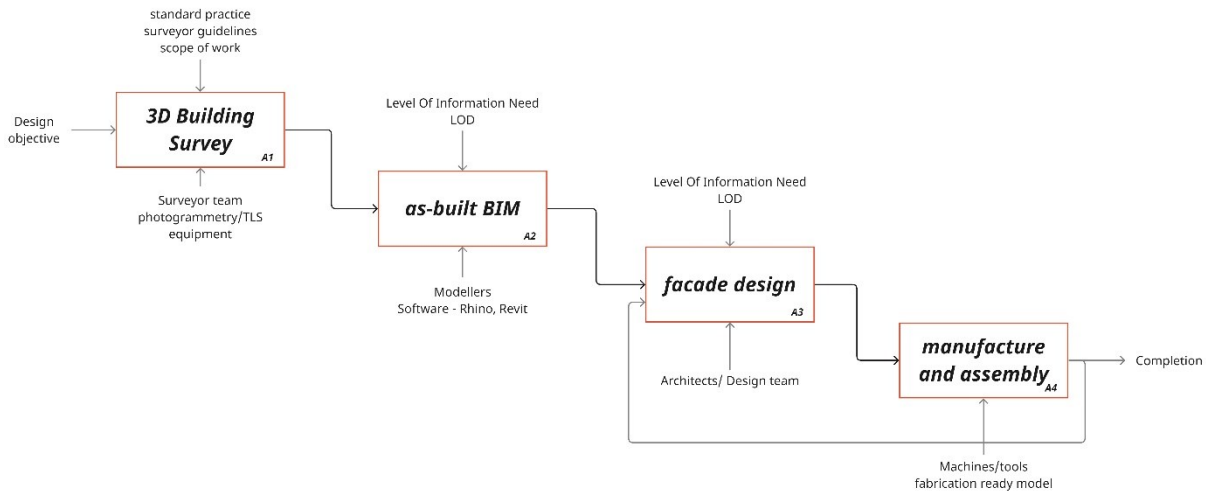


Figure 6: AS-IS process model for facade renovation using manufacture and assembly elements (Author)

Within these four main steps, many information exchanges, collaborations, and sub-processes will occur and in the following sub-sections, existing literature will be used to further investigate the current state of façade renovation.

2.1.1 Data Acquisition and Pre-processing

Before diving into tools and technologies of data acquisition and pre-processing of point cloud it is useful to refer to information requirement identification and data quality of scans (*Figure 7*). Identification of requirements could comprise required building elements, Level of Detail (LOD), and non-geometric attributes (Q. Wang et al., 2019).

The building elements that should be modeled in the as-is BIM must be a consideration when a 3D building survey is carried out (Q. Wang et al., 2019). For a façade, the elements could include exterior walls, doors, windows, ceiling and roof, and other extrusions such as slabs, staircases, and mechanical installations. LOD specifies the degree of specificity and detailed level of elements. Some examples of non-geometric attributes required are structural elements, acoustic properties, and surface properties such as colors and reflectivity (Q. Wang et al., 2019).

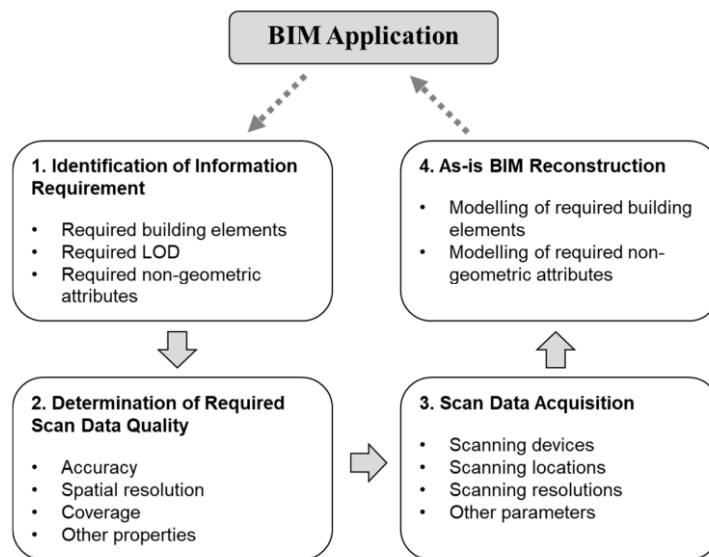


Figure 7. The proposed application-oriented scan-to-BIM framework (Q. Wang et al., 2019).

3D point cloud data could be acquired using image-derived methods (photogrammetry), 3D laser scanning (LiDAR), RGB-D (Red, Green, Blue, and Depth) cameras Synthetic Aperture Radar (SAR), or stereo cameras (Q. Wang et al., 2020; Xie et al., 2020).

Laser scanning and photogrammetry are used to compile point cloud data that could be useful for understanding the context of a building as in capturing as-built conditions (Sacks et al., 2018). The laser scanning technique uses sensors to measure distance nearby buildings and their context surfaces with high accuracy levels as much as several millimeters to a centimeter accuracy and at a speed of hundred thousand points per second (Tang et al., 2010b). These point clouds could contain several million to several hundred million 3D points.



Figure 8: Examples of three types of laser scanners according to working platforms (Q. Wang et al., 2020)

TLS in BIM is increasing in importance since it helps to capture more complex construction scenarios with different working platforms (Figure 8). In operation, TLS scans vertically, horizontally, and systematically within the predetermined angles and range (Figure 9). The distance to the point is calculated from the returning signal and thus creates a 3D polar coordinate system. These points are then converted to a Cartesian 3D coordinate system which results in a point cloud per scan (Bormann et al., 2018).

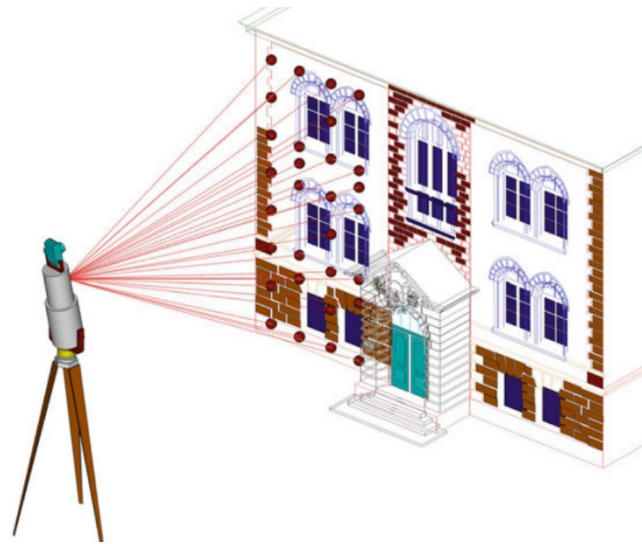


Figure 9: Terrestrial laser scanning (Bormann et al., 2018)

The distance measuring methods (Figure 10) are twofold mainly, time of flight and phase shift (phase comparison). In the time-of-flight method, the scanner emits a laser pulse and measures the time for the returning pulse to measure the distance, since the speed of the scanner is known. An Amplitude-Modulated Continuous-Wave (AMCW) is emitted in the other method, and the phase shift of the returning wave is used to calculate the distances. Figure 10 shows the two methods in a graphical way (Q. Wang et al., 2020).

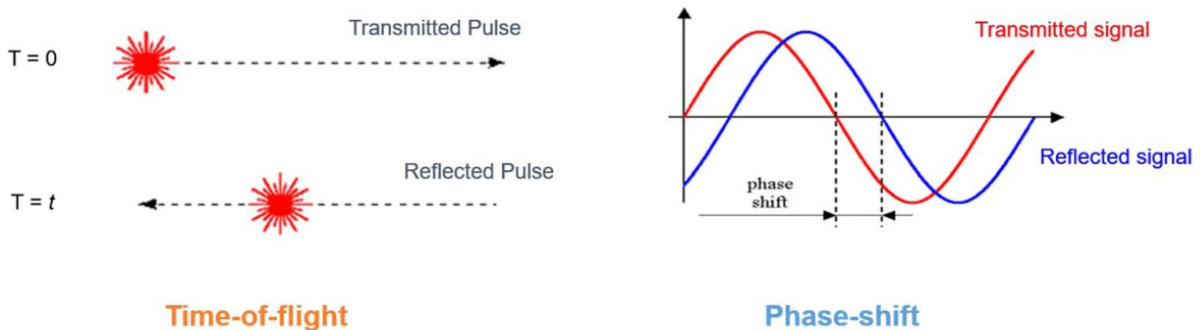


Figure 10: Working principles of the two methods, time of flight and phase shift (Q. Wang et al., 2020)

Photography and photogrammetry are two basic 3D surveying techniques for the existing building. The main difference between the two is the metric data we can acquire in photogrammetry as opposed to photographic surveys. Photogrammetry is taking images of an object from various angles and distances thus creating a series of photographs that could be used with algorithms such as structure from motion (SfM) or multi-view stereo (MVS) (Q. Wang et al., 2020; Xie et al., 2020). Recent technological advancements in mobile phones have also led to mobile and smartphone photogrammetry, making the activity more accessible. Also, aerial photogrammetry combined with Unmanned Aerial Vehicles (UAVs) provides greater flexibility and accuracy while it complements TLS as well (Borrmann et al., 2018).

RGB-D cameras can acquire RGB (Red, Green, and Blue) and depth information while interferometric SAR (InSAR) is a remote sensing radar technique to capture surface deformations or digital elevations (Xie et al., 2020). Stereo cameras have two or more lenses that can produce point clouds based on the two-dimensional (2D) images by the lenses (Q. Wang et al., 2020).

As shown in *Figure 11* data processing in captured point clouds could be categorized into data cleansing, registration, data segmentation, and object recognition (Q. Wang et al., 2020). In this research data pre-processing is explained as data cleansing and registration only.

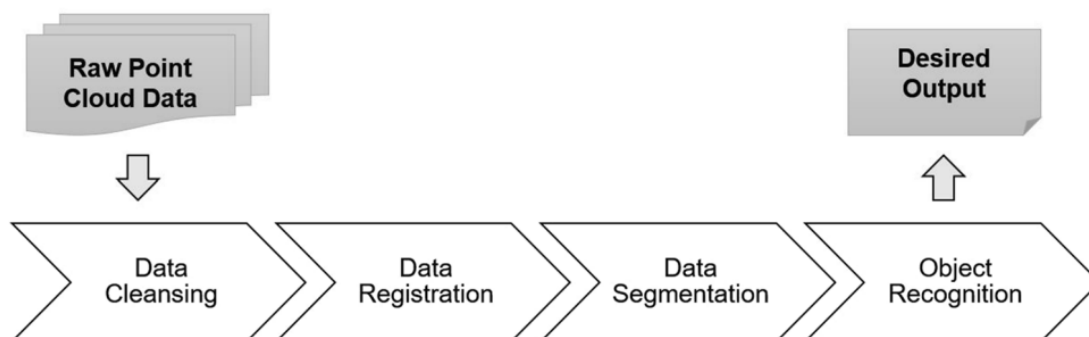


Figure 11: Processing of point cloud data (Q. Wang et al., 2020)

Point cloud files may contain noise mainly because of technical and fundamental limitations of the scanner's sensors. The noise may occur due to points not relating to any object and points from moving objects such as people. There are several algorithms developed by researchers over the years for data cleansing of point clouds (Q. Wang et al., 2020).

Every scan containing point clouds is in a local coordinate system and needs to be aligned and fit together in a global coordinate system. This preprocess is called registration and is a semi-automated process. (Tang et al., 2010a). The most common method is to start with course registration where point clouds are registered with less accuracy but to an aligned state. Then this initial solution of registration could then be finely registered with Iterative Closest Point (ICP) algorithm (Q. Wang et al., 2020).

2.1.2 Scan to BIM

Scan to BIM denotes the idea of generating BIM from a point cloud scan. It is useful to investigate point cloud segmentation and object recognition as a part of automating a BIM process since BIM specifies which methods to choose and which objects need to be identified from a point cloud.

Point cloud segmentation is when points are clustered relating to a specific surface or region that have similar characteristics resulting in homogenous regions. These clustering methods could be categorized into six depending on their methodology of segmentation (Q. Wang et al., 2020). In the clustering-based method, geometric, and radiometric aspects of each point are used as features. In edge-based segmentation algorithms are used to identify edges of regions. Segmentation based on regions uses algorithms that start with one or a few seed points and grow the region iteratively. Graph-based segmentations treat a point cloud as a graph and segmentation based on model fitting uses mathematical representations of primitive shapes for segmentation. A hybrid method is when two or more segmentation methods are used in combination (Q. Wang et al., 2020).

Figure 12 shows a segmented point cloud of a classroom containing desks, segmented with colors and with knowledge of the desk size. With two matrices called degree of completeness (DOC) and degree of matching (DOM), the segments are then evaluated to see how likely a segment is to represent a desk (Q. Wang et al., 2019).

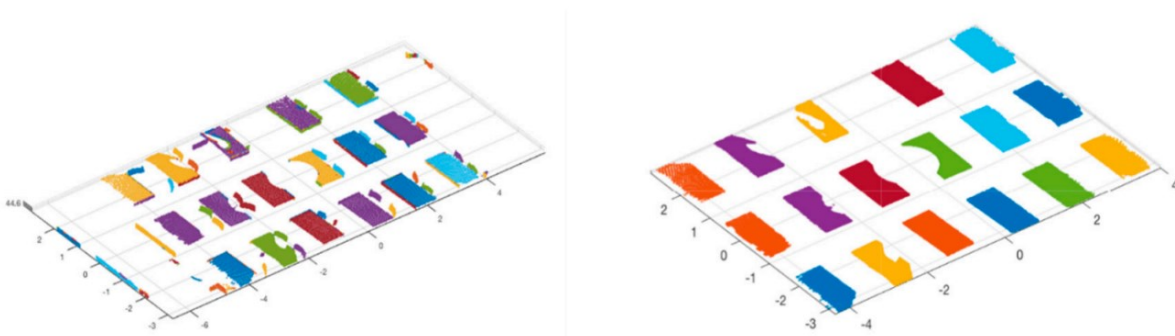


Figure 12: Segmenting of scan and recognition of individual desks (Q. Wang et al., 2019)

Research on object recognition on point clouds could be categorized into four main methods. A geometric shape descriptor explains a method to identify an object with a certain geometry such as pipe identification with curvature. The hard-coded knowledge-based method often recognizes planar patches in the point cloud which could be floors, walls, or roofs. This could be divided into four categories size, position, orientation, and topology where it is useful in recognizing building elements with distinct geometric characteristics. But when it comes to complex forms and geometry this method is less useful (Q. Wang et al., 2020).

Supervised learning mainly comprises two strategies, segment-based classification, and point-based classification. In point classification, each point is classified using a local feature of that point and on

the other strategy, point cloud data is segmented into meaningful parts and then classified into a class of objects using features.

The BIM vs. scan method assumes there is an as-designed BIM model already in place. This is mostly used in progress monitoring activities of a construction project, as-built reconstruction, geometry quality inspection(Q. Wang et al., 2020), or in renovation too, where there is already an as-built BIM.

2.1.3 Design for Manufacture and Assembly

Off-site construction and manufacturing have gained more productivity throughout the years as shown in *Figure 13* (Sacks et al., 2018). Since it is useful to investigate manufacture and assembly in the design and construction stages to gain more efficiency, productivity, and less major design changes.

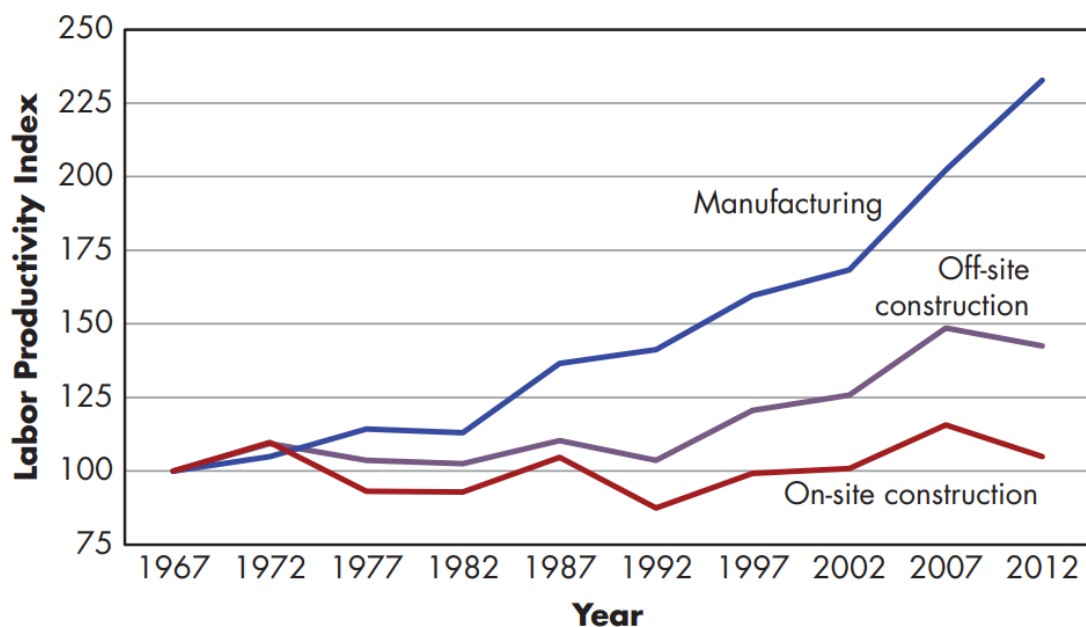


Figure 13: Labor productivity index for on-site construction, off-site construction, and manufacturing in United States market (Sacks et al., 2018).

If the as-designed BIM could be transferred to the fabrication model, the process of fabrication also could be automated using numerically controlled (NC) machinery. This process allows quick response to aspects such as design changes and synchronizes design and construction planning. As-designed BIM model could be used to evaluate the effective use of the manufacture and assembly elements and is better to incorporate in the early design phase. The project team will be allowed to consider design, cost, and time factors aligned with the elements to facilitate the as-designed BIM (Sacks et al., 2018).

Computer Aided Design (CAD) could be digitally transferred to Computer Numerically Controlled (CNC) machinery through a Computer Aided Manufacturing (CAM) tool (*Figure 14*). CAM tools initially generate a common NC control code which is system-independent and then with a post-processor tool the required data is transferred to a specific machinery (Borrmann et al., 2018).

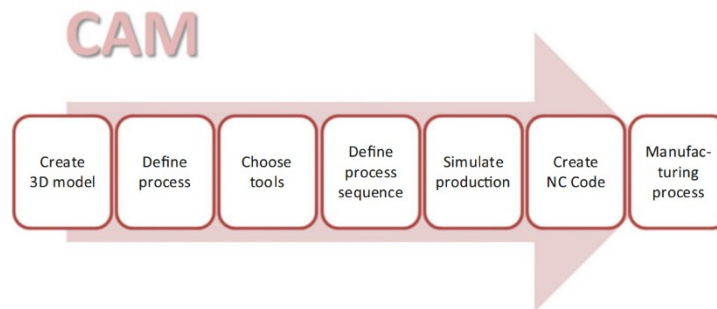


Figure 14: Stages of CAM process (Borrmann et al., 2018)

In the example shown in *Figure 15*, BIM implemented in project initiation, could be useful in the fabrication of façade panels. The feedback about the fabrication and installation was crucial to the project team in their design stage and it ensured successful manufacture and assembly. The feedback received included details about lifting panels, transporting more panels at a time, connection details of panels, and precise as-built information for panel fitting (Sacks et al., 2018).

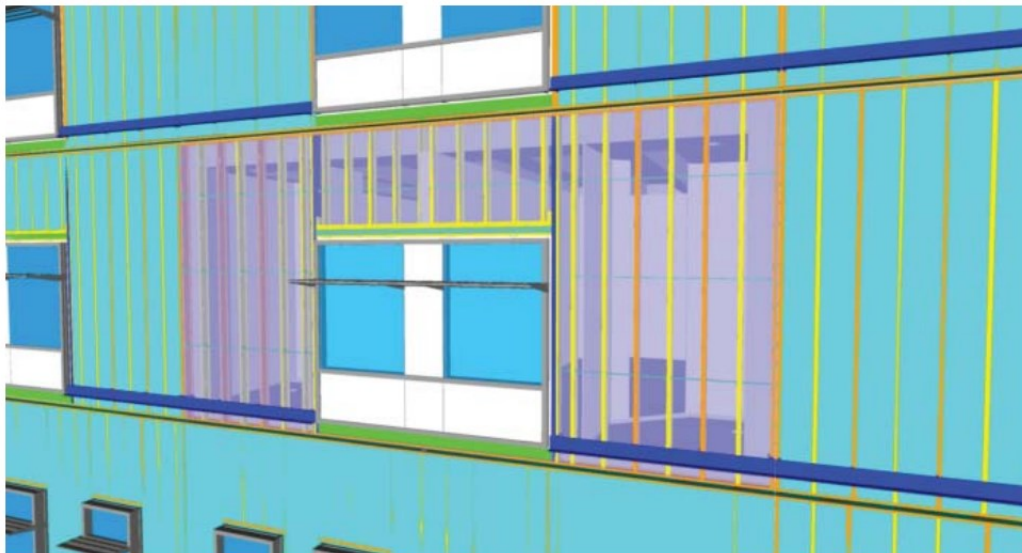


Figure 15: St. Joseph Hospital in Denver, US (Sacks et al., 2018)

There is numerous research focused on fabrication-aware design that represents the same notions such as design for manufacture and assembly (Lu et al., 2021). In some research, it has been suggested to use architectural design geometry as the fabrication-aware design to fill the gap between construction and design (Pottman, 2009; Pottmann, 2010, 2013). These approaches improve manufacturability and assemblability in the early stages of design (Lu et al., 2021).

Also, parametric design is key to this approach. One approach is DfMA-oriented parametric design to transfer professional knowledge of manufacture and assembly to early design stages through the DfMA-oriented design team (Yuan et al., 2018b). Below is a diagram showing optioneering in the concept design stage with parametric data. This results in instant validation via brief requirements (Alfieri et al., 2020).

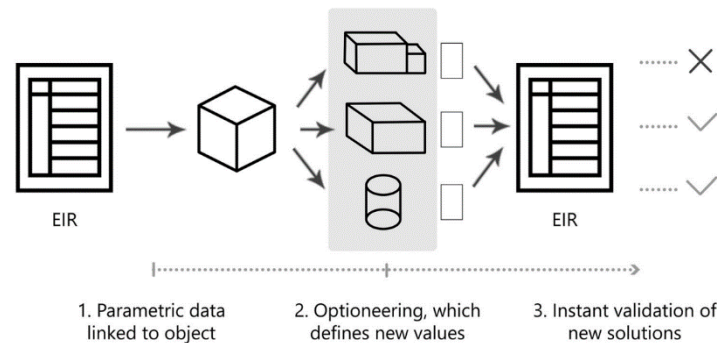


Figure 16: Optioneering flow (Alfieri et al., 2020)

2.2 Information Requirement and Exchange

Information requirement in a project starts from an appointing party's requirements concerning a project or an asset in the life cycle to align with their objectives (International Organization for Standardization, 2018). International Standard Organization's 19650-1:2018, suggests a comprehensive understanding of information in an asset or project life cycle. It is useful to refer to each stage in the information flow and apply it to the workflows.

Firstly, ISO 19650-1 states its place in the BIM maturity stages as shown below. BIM maturity stages could be useful to evaluate the degree of information exchange and collaboration taking place.

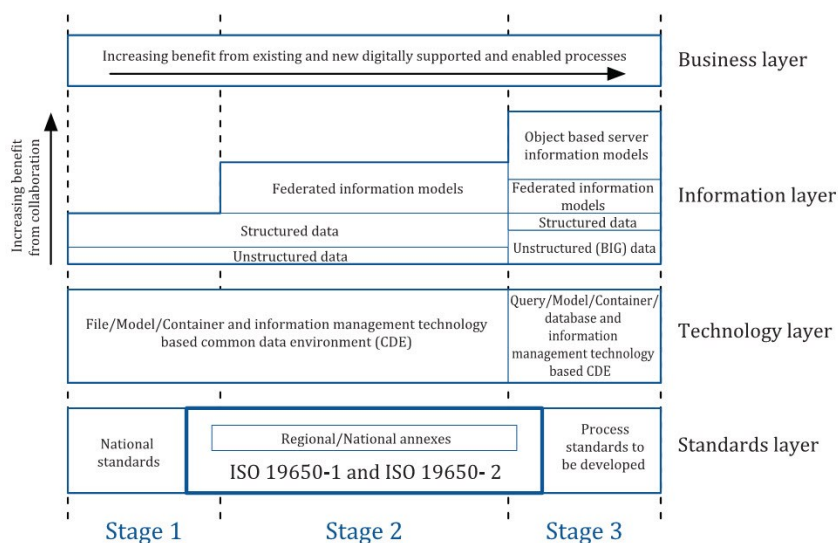


Figure 17: BIM Maturity stages and information management (International Organization for Standardization, 2018)

Both ISO 19650-1 and 19650-2 majorly fall in BIM maturity stage 2, but it could also be partially applicable to stages 1 and 3. This partial applicability suggests that collaborative information management takes place with a mixture of automated and manual management of information that provides a federated model. The federated model contains all the information about the project (International Organization for Standardization, 2018).

From the information management perspective, ISO 19650-1 states four categories that are useful in the specification, planning, and delivery of the required information. They are the asset owner's perspective, the asset user's perspective, the project delivery or asset management perspective, and society's perspective. Even though all these perspectives interest and overlap with each other, out of these four, the project delivery perspective concerns the appointed party the most. Here the appointed party refers to the project delivery team comprising but not limited to BIM managers, BIM coordinators, architects, and engineers.

This section explores information management from an appointed party's perspective while noting their connections to the different information exchanges and requirements.

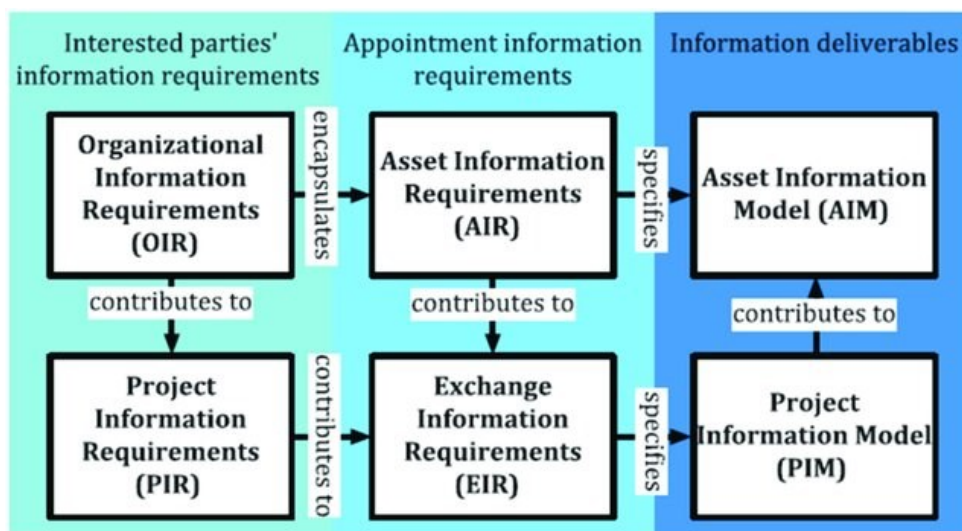


Figure 18: Hierarchy of information requirement (International Organization for Standardization, 2018)

The above diagram explains the major information exchanges and their relationship to each other. The starting point, Organizational Information Requirement (OIR), denotes the required information by an appointing party aligning with their organizational objectives (International Organization for Standardization, 2018). This will result in Asset Information Requirements (AIR) and Project Information Requirements (PIR), where the laid-out reasons in the OIR could be explained in managerial, commercial, and technical terms. The appointing party's purposes for specific information requirements should also be stated and typically comprise asset registry, regulatory responsibilities, risk

management, support for business decisions, security, and surveillance, and support for renovation, maintenance, repair, replacement, disposal, or decommissioning.

The appointed party considers all the information requirements in AIR, and PIR to prepare EIR. This is a crucial point in the project since EIR sets out the deliverables there in, PIM and AIM. EIR comprises commercial, managerial, and technical features of project information to be produced. The requirement should be explained as project-related appointments and wherever an appointment is made. Here an appointment denotes instructions for information regardless of a formal appointment between parties.

The lead appointed party could make their own EIR based on the received EIR from the appointing party to subdivide work and when information exchange is crucial but exclusive within a delivery team. Within the project, there may be several appointments throughout. However, the EIR should contain all the appointments in the form of a single coherent and coordinated series of information requirements that covers all the PIR (International Organization for Standardization, 2018).

Lastly, AIM and PIM denote the asset management processes that occur on a day-to-day basis and an archive of information on the project such as geometry, methods of construction, costing, scheduling, installed systems, equipment information, and maintenance requirements. These two models' requirements are defined by the appointing party and the delivery team produces them.

2.3 Summary

From this chapter, the following findings are identified as most important to the scope of this research.

It is important to consider the scan requirements and data quality of the point cloud before carrying out surveying. *Figure 19* summarizes the relevance of Scan to BIM process.

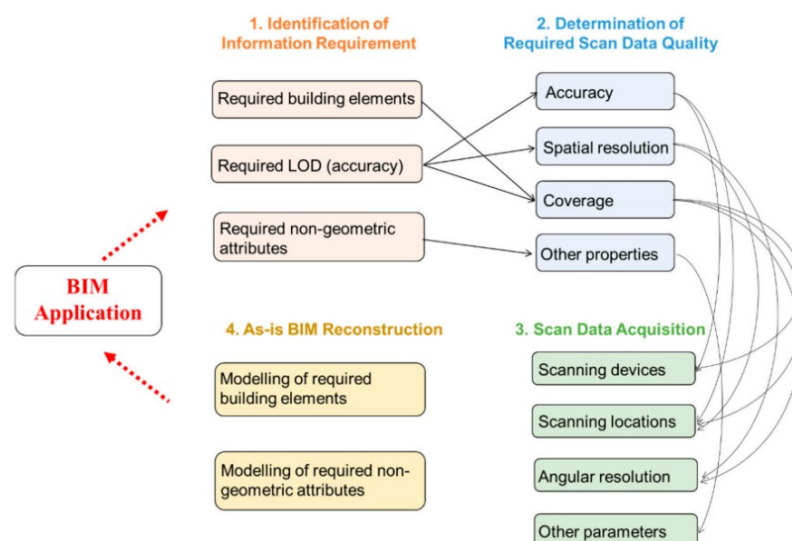


Figure 19: Summary of a scan to BIM framework (Q. Wang et al., 2019)

It is important to pre-process point clouds with data cleansing and registration to use the point cloud in the next steps.

Out of the object recognition methods, segment-based classification is promising and could be used for complex point clouds.

Parametric design is a crucial aspect when designing for manufacture and assembly. Fabrication-aware design and feedback in the early design stage are useful for efficient BIM workflow.

The above-said information requirements and exchanges are vital for this workflow digitization. An EIR may capture all information needs within the appointed party, in managerial, commercial, and technical terms.

3 OPEN SOURCE AND OPENBIM

In the article titled, “Native OpenBIM, and the rise of open source in AEC”, Dion Moul, the primary developer behind BlenderBIM writes that openBIM and open source are linked by the common idea of giving control back to the user. openBIM allows the data to be structured in a standardized manner while being vendor agnostic. Open source on the other hand allows users to share, customize and learn freely in the vendor offerings (Moul, 2022). This chapter investigates the use of open-source software and openBIM standards in digitizing the façade renovation workflow using manufacture and assembly elements.

3.1 OpenBIM

buildingSMART International is the institute that develops and maintains the openBIM industry standards such as IFC, bsDD, and BCF. These standards allow AECO industry stakeholders to collaborate and exchange information better. The principles of openBIM contain interoperability, open and neutral standards, reliable data exchanges, enhanced collaboration, flexible technology choices, and sustainability (buildingSMART International, 2020). This section further elaborates on IFC and its possible applications.

ISO-certified IFC is at the center of openBIM which contains a digital description of a built asset in a standardized way (buildingSMART International, 2019).

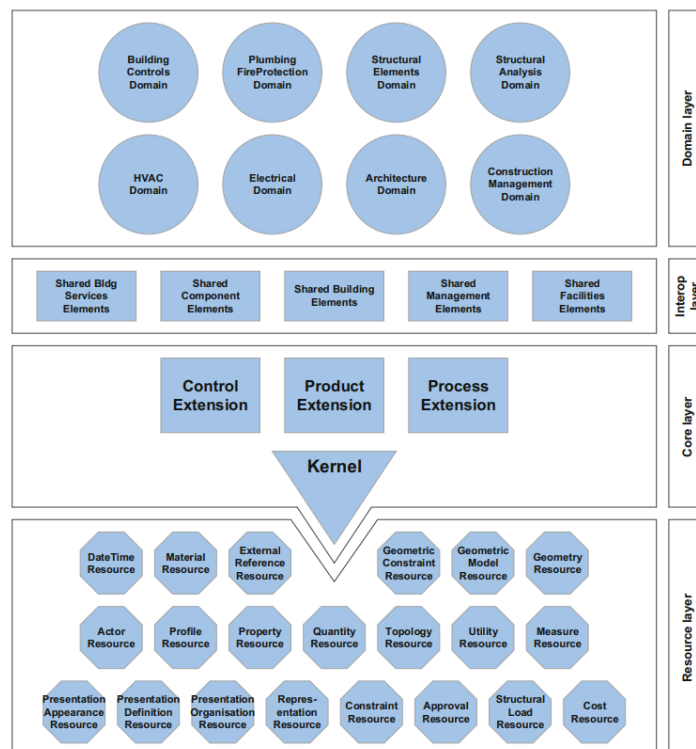


Figure 20: IFC data model (buildingSMART, n.d.)

IFC relies on ISO-STEP EXPRESS language and reflects the semantically rich data relating to the building industry. The basis of the IFC structure could be explained as objects and their respective relationships. In an IFC all object information is in a hierarchy of *Project* → *Site* → *Building* → *BuildingStorey* → *Space* (Sacks et al., 2018).

In the data model (Figure 20), the core layer carries the most elementary classes and relationships that are used by all the layers above. It comprises of Kernal schema representing core abstract classes such as *IfcRoot*, *IfcObjectDefinition*, *IfcPropertyDefinition*, *IfcRelationship*, *IfcActor*, and *IfcProduct*. The core layer also includes the three schema extensions Control, Product, and Process. Interoperability is crucial in deriving different application schemes such as *IfcWall*, *IfcWindow*, and *IfcDoor*. The domain layer consists of highly specialized layers and IFC4 defines domains such as architecture, construction management, building control, and structural elements. Resource schema includes basic data structures that are used throughout the model (Borrmann et al., 2018).

Figure 21 shows parts of the inheritance hierarchy of an IFC that follows a semantic approach. It shows which classes and attributes could be inherited by other classes.

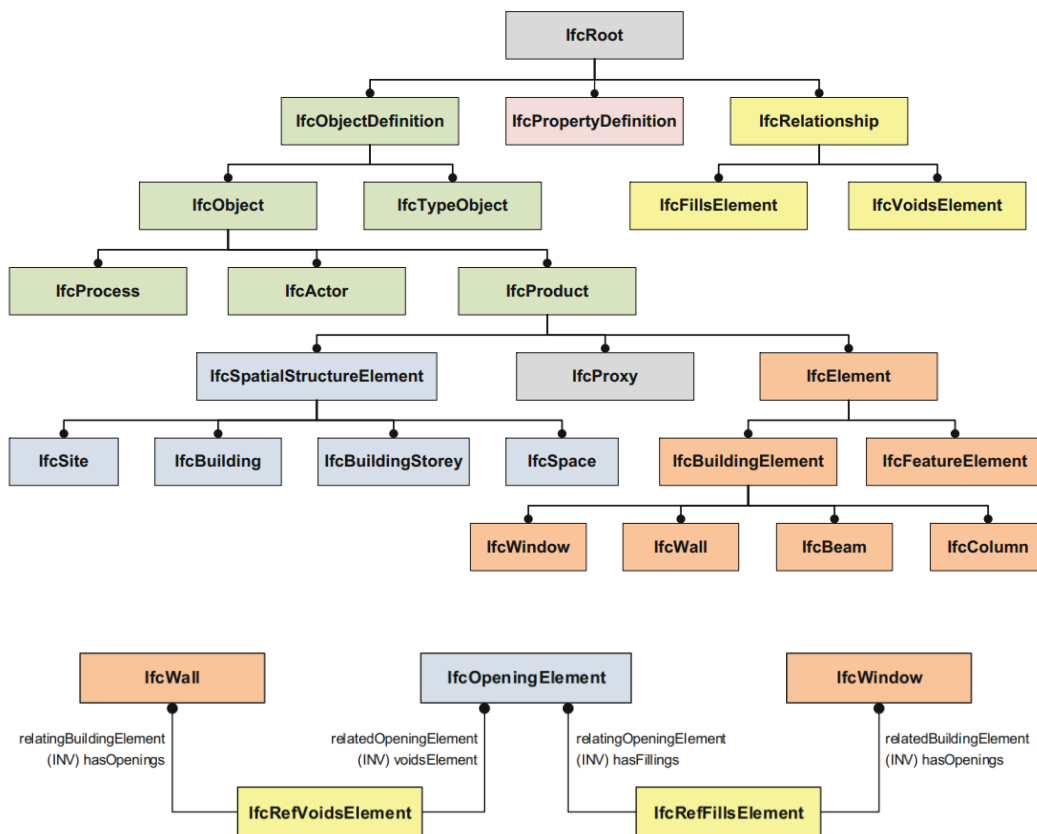


Figure 21: Part of the inheritance hierarchy and inheritance hierarchy of a wall (Borrmann et al., 2018)

The IFC data model has a clear division between geometric representation and semantic description, allowing for more flexibility in assigning one or many geometries to a semantic description. `IfcGeometricRepresentationItem` is the abstract class that all geometry classes inherit from. One of its sub-classes is `IfcSolidModel` which represents solids (Figure 22).

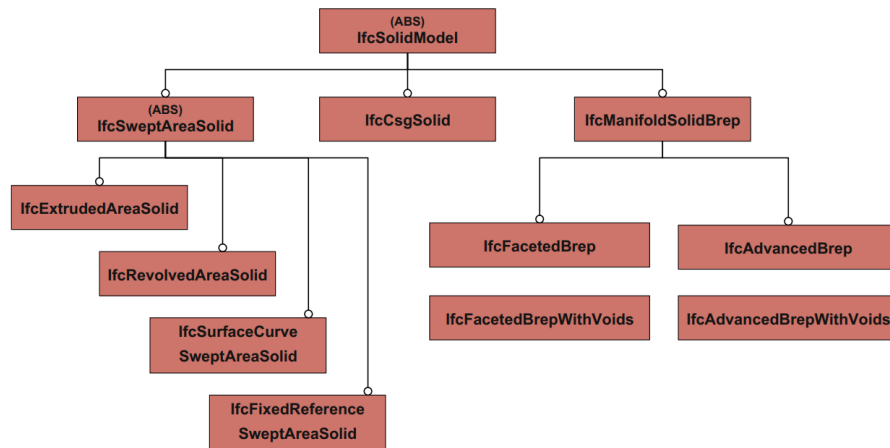


Figure 22: Solid modeling in IFC data model (Borrmann et al., 2018)

It is useful to investigate `IfcExtrudedAreaSolid` means of generating solid models. Here, `IfcRectangularProfileDef` or `IfcParameterizedProfileDef` could be used with direction (buildingSMART, n.d.) It is also useful to consider `IfcCartesianPointList`, `IfcVector`, `IfcCartesianPoint`, and `IfcDirection` to define vectors, points, and directions (Borrmann et al., 2018). `IfcFacetedBrep` could also be useful to consider when creating complex representations with flat surfaces (Figure 23).

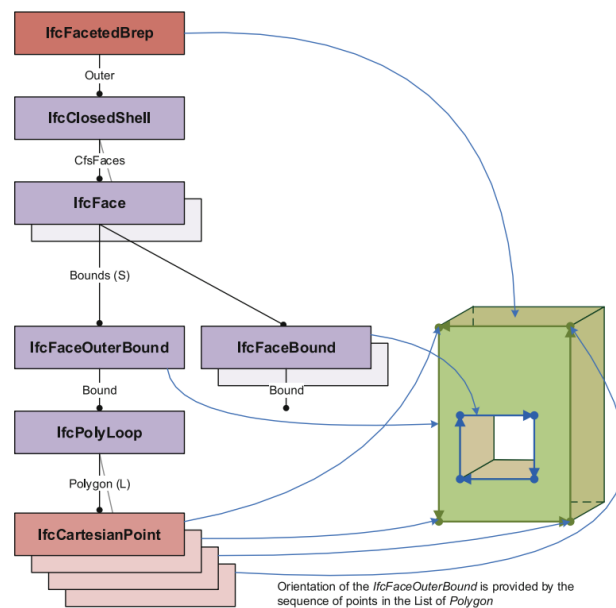


Figure 23: Boundary representation with `IfcFacetedBrep` for advanced flat surface model representation (buildingSMART, n.d.)

3.2 Open-Source BIM Platforms

Contrary to proprietary closed-source software platforms, open-source software is developed and maintained via collaboration, typically at no cost, and made available, for any user to examine, change, and redistribute as they like (IBM, n.d.). The use could be both commercial or non-commercial but mainly guided by key principles of transparency, decentralization, and collaboration. These have numerous advantages such as flexibility and customization, being equipped with the latest technology, no vendor barriers, and community support. The software is distributed by a license such as MIT, GNU General Public License (GPL) v2, or GNU GPLv3. Some specify if the user is permitted to use and distribute for any purpose or requirement to explicitly log any changes when sharing (GitHub, n.d.).

Even though there is a growing number of open-source software that supports BIM workflows, the following sections will explore the most promising software and platforms.

3.2.1 FreeCAD

FreeCAD is an open-source parametric CAD modeler under LGPL License. Even though it is directed at product design and mechanical engineering it is also capable of BIM. Currently, there is an external BIM workbench built into the 'Arch workbench'. The BIM workbench could be added to FreeCAD using the addon manager (*Figure 24*). BIM tools are categorized into 2D drafting, annotation, 3D/BIM modeling, modification tools, and management tools (FreeCAD, n.d.).

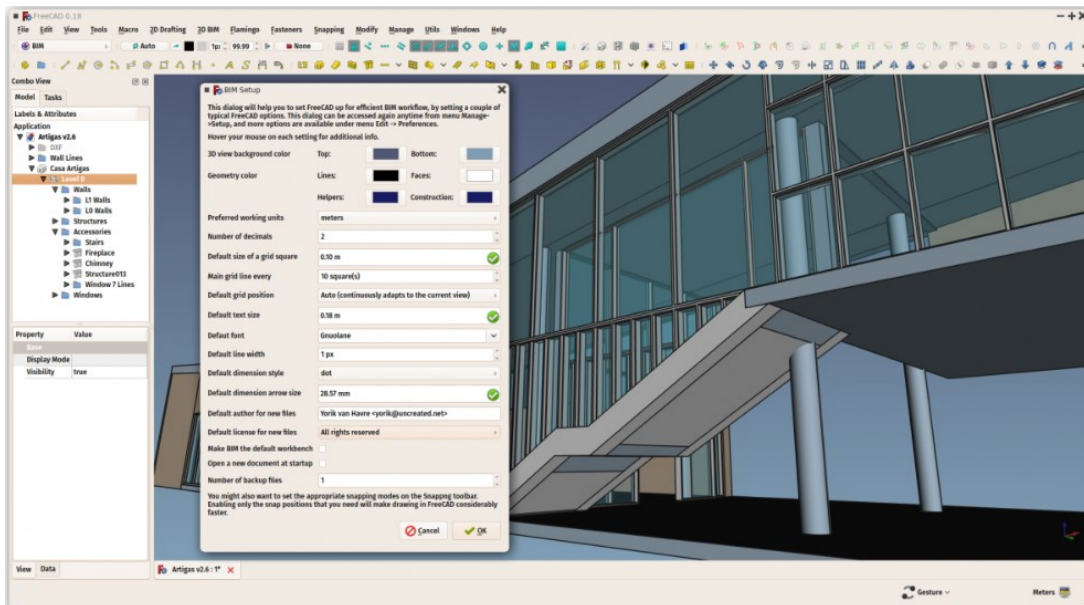


Figure 24: FreeCAD interface (FreeCAD, n.d.)

3.2.2 Speckle

Speckle is an open-source digital infrastructure for collaboration, versioning, data management, and automation of 3D data. This provides seamless integration and interoperability between BIM and CAD software making this platform great for BIM workflows.

Speckle comprises 3 main parts, Speckle Web App (*Figure 25*), Connectors (*Figure 26*), and Speckle Manager. While web apps allow you to manage the data through a browser, connectors provide essential plugins for industry-leading software such as Revit, Rhino, Grasshopper, Blender, AutoCAD, Sketchup, Civil3D, and Unity. These connectors will handle BIM data, geometry, and metadata mainly. Connectors make the data model speckle's neutral form that makes it fast to access data through another plugin or software. Speckle manager handles all the account management and server management tasks (Speckle, n.d.).

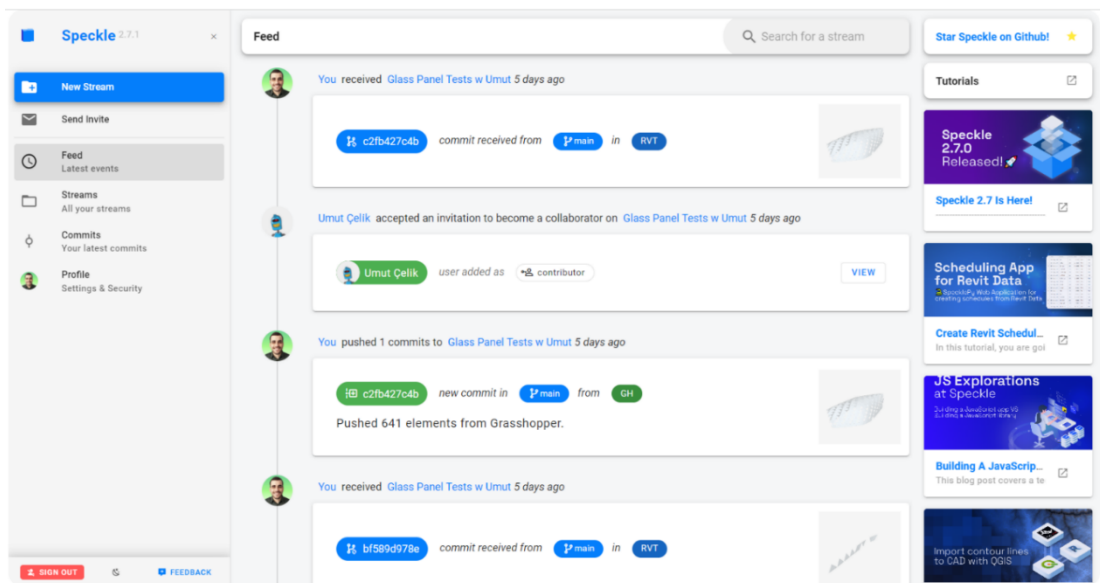


Figure 25: Speckle Web interface (Speckle, n.d.)

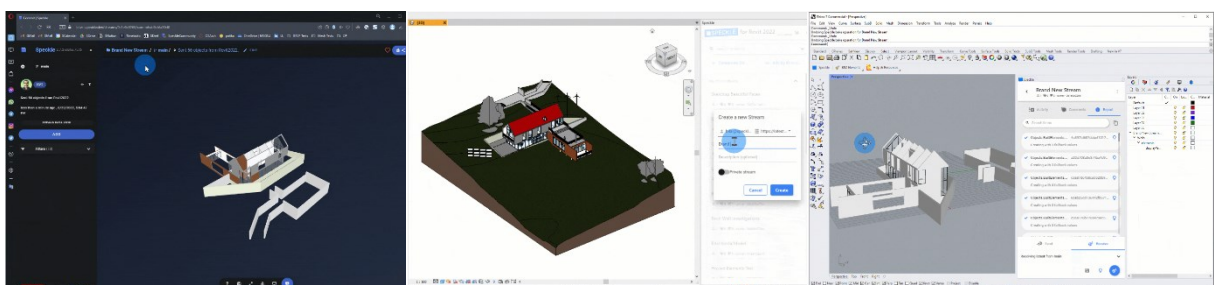


Figure 26: Speckle connectors for Revit, Rhino, and web viewing (Speckle, n.d.)

3.2.3 Blender and BlenderBIM

Blender is an open-source 3D creation suite with the capability to employ its Application Programming Interface (API) to customize the application and write tools as required with Python scripting. It is free software with no cost added and released under GNU General Public License (GPL). It is also a community-driven project expanding releases with minor and major changes to the code base (Blender, n.d.).

The geometry node option in Blender is useful in modifying geometry with node-based operations (*Figure 28*). A specific geometry could pass through a modifier and output with the group output node. Geometry nodes could modify meshes, curves, point clouds, volumes, and instances (Blender, n.d.).

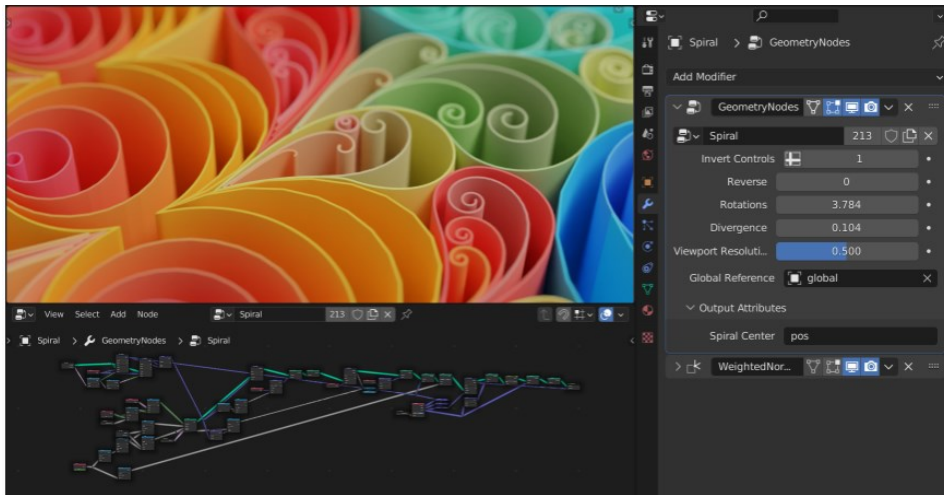


Figure 28: Geometry node examples in Blender (Blender, n.d.)

BlenderBIM Addon is a free-to-use addon that can be used to create, analyze, and modify openBIM. It allows users to author IFC models, create geometry, insert metadata, and other useful BIM workflows. Currently, BenderBIM allows users to create an IFC project or load an existing IFC to the software. When creating an IFC object, the addon lets the user assign it to a class and then to the proper place in the spatial tree (*Figure 27*).

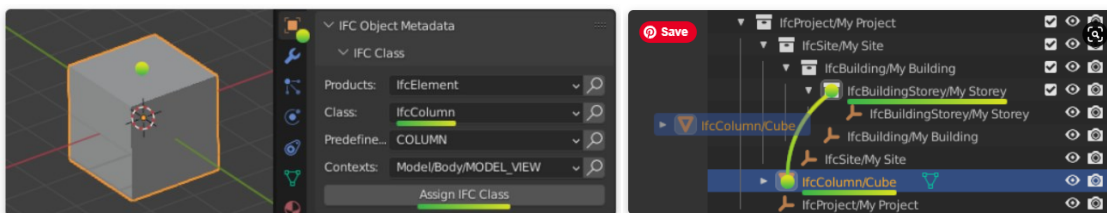


Figure 27: Assigning class and placing in spatial tree in BlenderBIM (BlenderBIM, n.d.)

Ifcopenshell holds a core low-level functionality in the BlenderBIM add-on to read and write the data. The add-on allows to natively author IFCs rather than import and export IFC-based collaboration works in other software (BlenderBIM, n.d.).

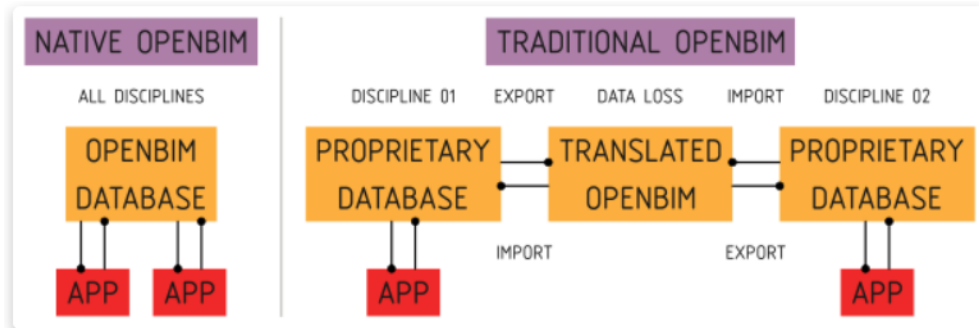


Figure 29: Native IFC authoring of BlenderBIM vs traditional openBIM (BlenderBIM, n.d.)

Operations that write or read data are always represented in IFC (Figure 29). Blender is just the interface between the user and IFC where it gets stored (BlenderBIM, n.d.). BlenderBIM is developing rapidly with large community backing and stable developer models of the add-on can be accessed through GitHub. Using BlenderBIM users can generate a range of BIM tasks as shown in Figure 30.

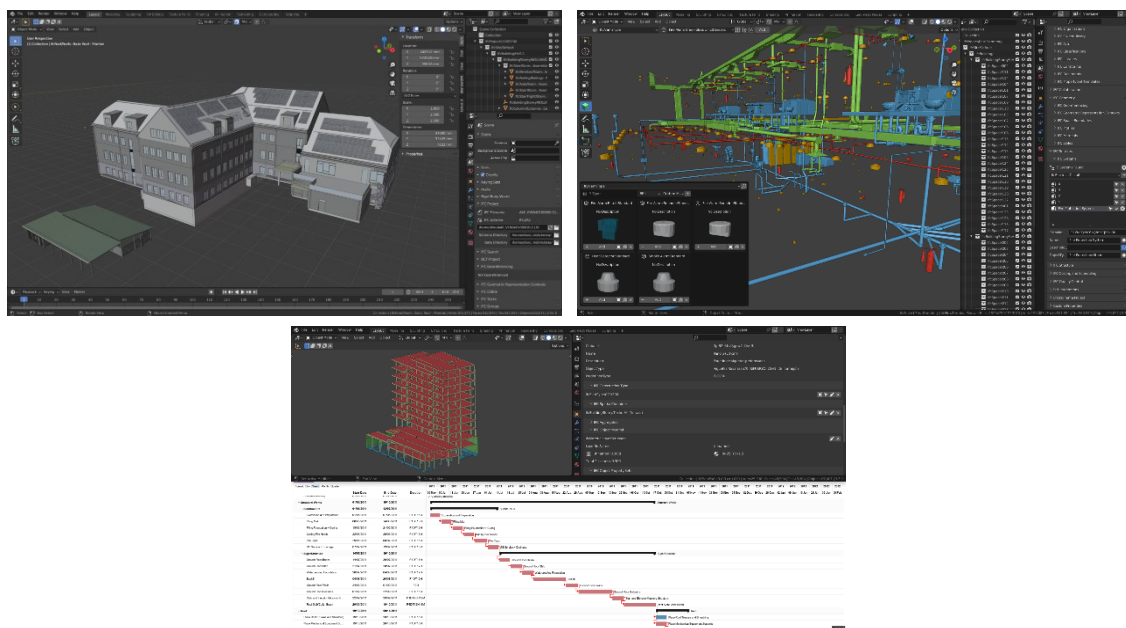


Figure 30: Geometry in IFC and scheduling in BlenderBIM (BlenderBIM, n.d.)

Another promising add-on for Blender for BIM is, IfcSverchok, an add-on capable of generating parametric objects.

3.3 Useful tools and software

In the operations on point cloud files and IFCs, this section will explain useful Python libraries and open-source software.

3.3.1 CloudCompare

CloudCompare is an open-source tool dedicated to point cloud operations such as viewing, converting, registration, segmentation, and meshing (*Figure 31*). It is free to use software for educational or commercial purposes defined by GNU General Public License (GPL). (CloudCompare, n.d.)

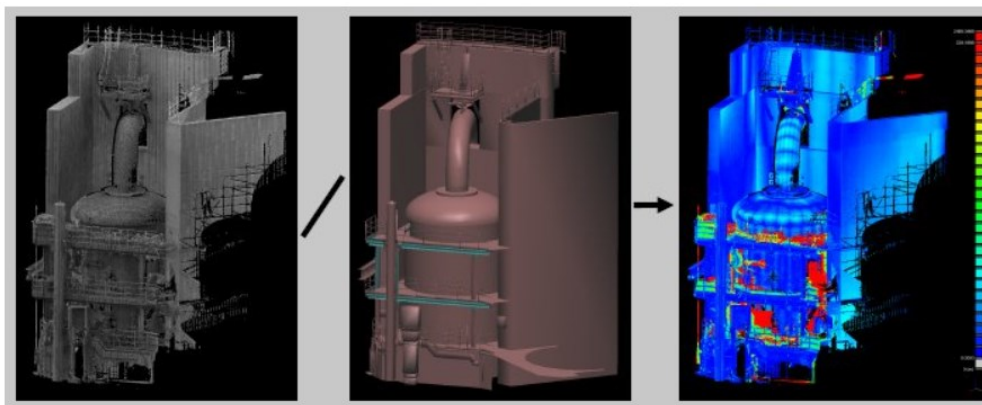


Figure 31: Meshing and distancing operations in CloudCompare (CloudCompare, n.d.)

3.3.2 Ifcopenshell

Ifcopenshell is an open-source toolkit for IFC operations such as reading, writing, or modifying BIM (*Figure 32*). The library is offered in C++ and Python languages. The latest Ifcopenshell version 0.7.0 has geometric supports IFC releases IFC2x3 TC1 and IFC4 Add2 TC1 while parsing support is provided for IFC4x1, IFC4x2, and the IFC4x3.

```
import ifcopenshell

model = ifcopenshell.open('model.ifc')

for wall_type in model.by_type("IfcWallType"):
    print("The wall type element is", wall_type)
    print("The name of the wall type is", wall_type.Name)
```

Figure 32: Sample code to get all wall types with Ifcopenshell (IfcOpenShell, n.d.)

Since Python scripting is possible in Blender, Ifcopenshell could be imported to Blender (Figure 33) and run as well. (IfcOpenShell, n.d.).

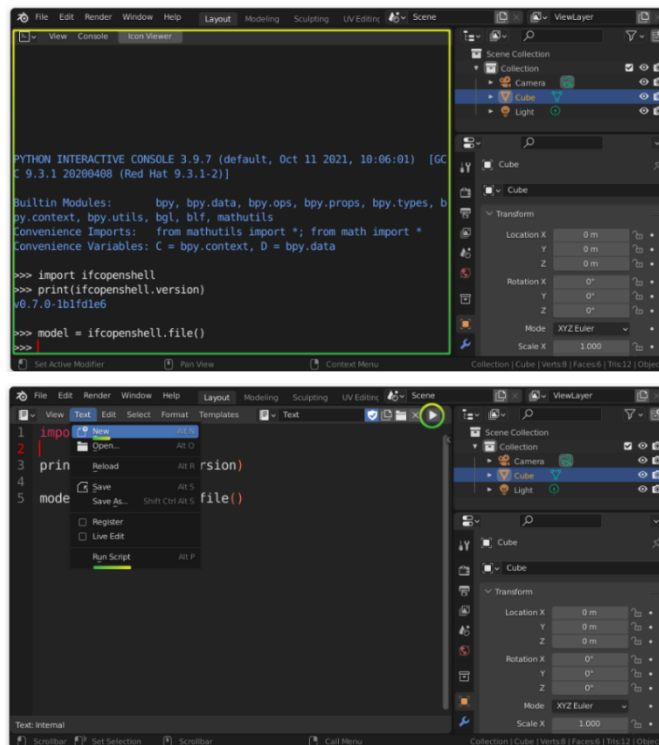


Figure 33: Python console and text editor functions in Blender to run Ifcopenshell scripts (IfcOpenShell, n.d.)

3.3.3 Open3D

Open3D is a library that supports software dealing with 3D point cloud data. Its data structures and algorithms are both in C++ and Python with an open-source community and released under MIT license. Core features of Open3D include 3D data processing, scene reconstruction, surface alignment, and 3D visualization (Figure 34, Figure 35). The pre-built package supports Python 3.6-3.10 with Ubuntu 18.04+, macOS 10.15+, and Windows 10+ (64-bit). Some operations of Open3D would require NumPy and Matplotlib as well (Zhou et al., 2018).

```
print("Load a ply point cloud, print it, and render it")
ply_point_cloud = o3d.data.PLYPointCloud()
pcd = o3d.io.read_point_cloud(ply_point_cloud.path)
print(pcd)
print(np.asarray(pcd.points))
o3d.visualization.draw_geometries([pcd],
                                   zoom=0.3412,
                                   front=[0.4257, -0.2125, -0.8795],
                                   lookat=[2.6172, 2.0475, 1.532],
                                   up=[-0.0694, -0.9768, 0.2024])
```

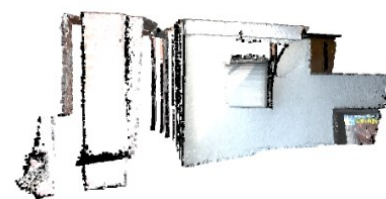


Figure 34: Example of code to visualize a point cloud (Open3D, n.d.)

```
print("Downsample the point cloud with a voxel of 0.05")
downpcd = pcd.voxel_down_sample(voxel_size=0.05)
o3d.visualization.draw_geometries([downpcd],
                                   zoom=0.3412,
                                   front=[0.4257, -0.2125, -0.8795],
                                   lookat=[2.6172, 2.0475, 1.532],
                                   up=[-0.0694, -0.9768, 0.2024])
```

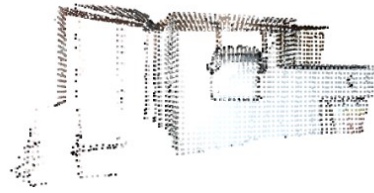


Figure 35: Example of voxel down sampling operation on a point cloud (Open3D, n.d.)

Open3D only allows file formats XYZ, XYZN, XYZRGB, PTS, PLY, and PCD. Some common point cloud formats such as LAS or E57 are not supported.

3.4 Summary

This chapter provides a concise overview of factors and components that hold potential value in the case study application.

Regarding IFC geometric representation, it could be useful to consider `IfcExtrudedAreaSolid` and `IfcFacetedBrep` to generate geometry in IFCs and use classification to assign metadata and hierarchical relationships.

Even though FreeCAD and Speckle have a lot of potential to be helpful for BIM workflows and collaboration, BlenderBIM's capability to author IFC and Blender's built-in scripting could be more advantageous.

CloudCompare is useful for operations in a point cloud as a standalone software and it is useful in converting files and pre-processing of point clouds.

`Ifcopenshell` Python library lets the user do operations to IFCs with scripting knowledge. Open3D is also a promising tool to read, modify, and write point cloud files. BlenderBIM's ability to author IFC and Blender's capability for scripting that allows `Ifcopenshell` and Open3D integration makes a promising combination of openBIM and open-source approach for the case study application.

4 CASE STUDY

This chapter lays out the aspects of the proposed workflow in a hierarchical way, from the TO-BE model to practical application in software with the case study application. The following sections comprise the case study overview, proposed TO-BE model, digitization strategy, and investigation of the proposed workflow using openBIM and open-source tools.

4.1 Introduction

This section will explain the structure of the following sections. Considering the AS-IS model presented in Chapter 2, the proposed TO-BE model is presented (*Figure 36*). After that, the digitization strategy will follow laying the foundation for the proposed workflow. The subsequent section explains the challenges and benefits while the chapter will close out with results and discussion. *Figure 36* further explains this information as a methodology.

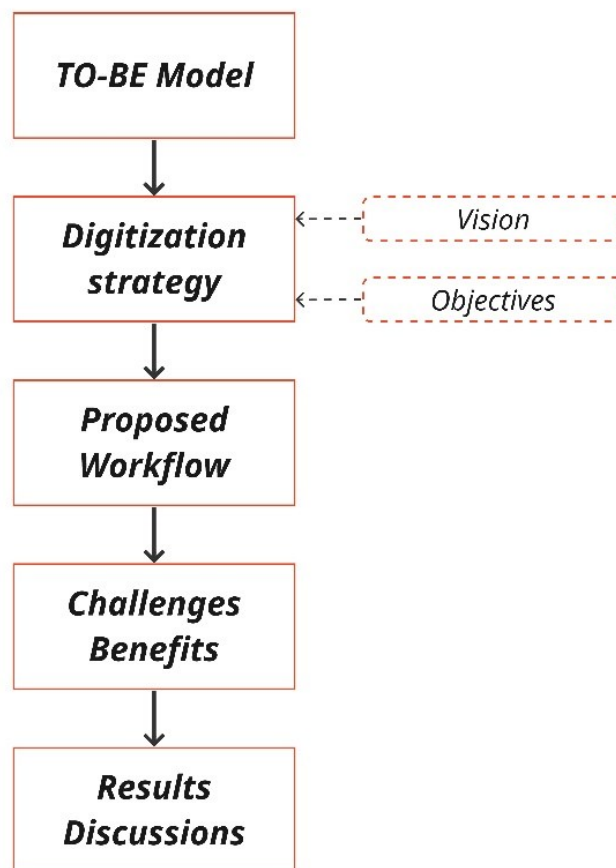


Figure 36: Methodology for case study application (Author)

4.1.1 Case Study Overview

The case study comprises a point cloud file provided by the Fraunhofer Italia Company. This Point cloud is surveyed using Faro 3D laser scans and scan registration has already been done. It is a survey of a facade containing the ceiling of a two-story house in Italy. The point cloud showcases the necessary details of building elements of the façade needed for the BIM model (*Figure 37*).

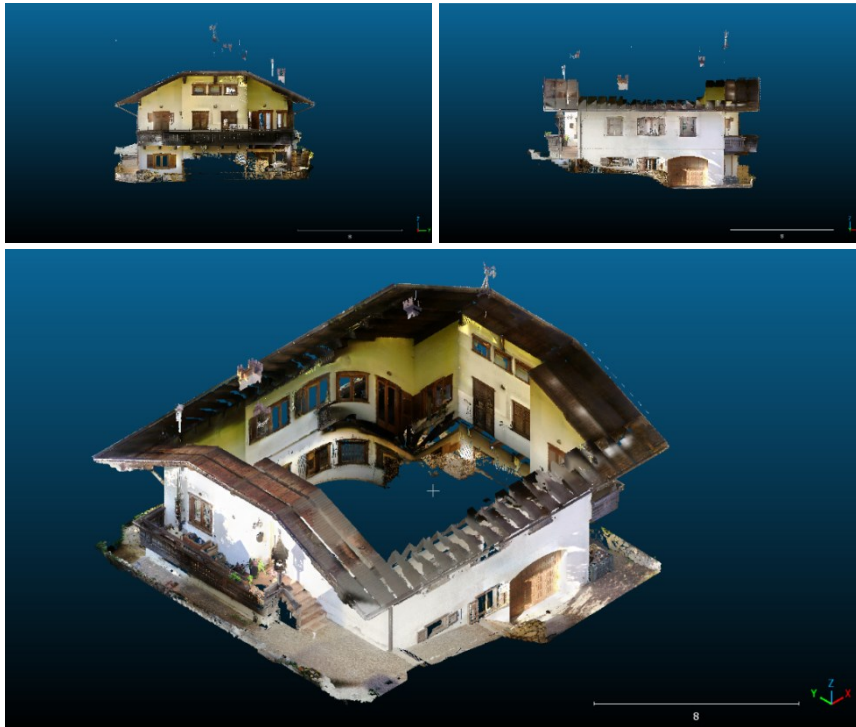


Figure 37: Case study point cloud viewed in CloudCompare (Author)

4.1.2 Software and Tools

Following the summary of Chapter 3, the digitization strategy is focused on generating IFCs and using the capabilities of Blender with BlenderBIM addon as an authoring platform for IFCs. Visual scripting in Blender as well as its scripting integration is useful for the proposed workflow. As showcased in Chapter 3, CloudCompare and Python libraries Ifcopenshell and Open3D are also included in the workflow (*Figure 38*).

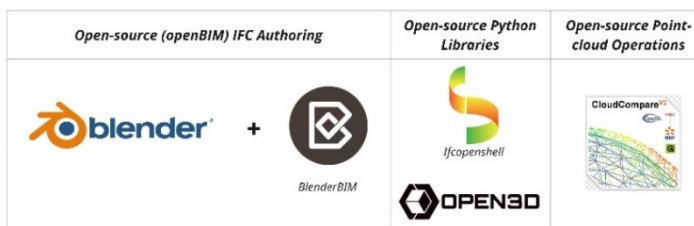


Figure 38: Open-source and openBIM software and tools for case study (Author)

4.2 Developing Framework

This section will elaborate on the TO-BE model for the proposed workflow that identifies the digitization strategy. Based on the AS-IS model in Chapter 2, the proposed TO-BE model introduces new subprocesses, actors, and information exchanges. The objective for developing the TO-BE model is to increase efficiency and collaboration in the workflow, ensure accurate and unhindered information flow, and timely completion of projects.

4.2.1 TO-BE Model

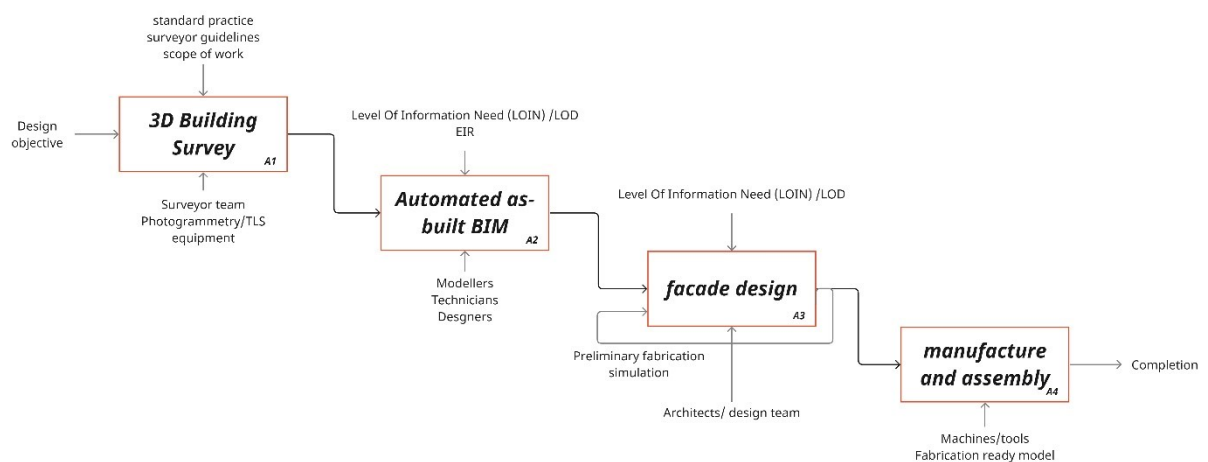


Figure 39: Proposed TO-BE process model (Author)

The proposed To-BE model (Figure 39) comprises four stages A1-A4, covering 3D building survey to assembly onsite. The highlights of the model are automated as-built BIM from point cloud data and preliminary fabrication simulation in the early design stage to ensure fabrication-aware design. The following sub-process models further investigate the information requirements and exchanges in each main process.

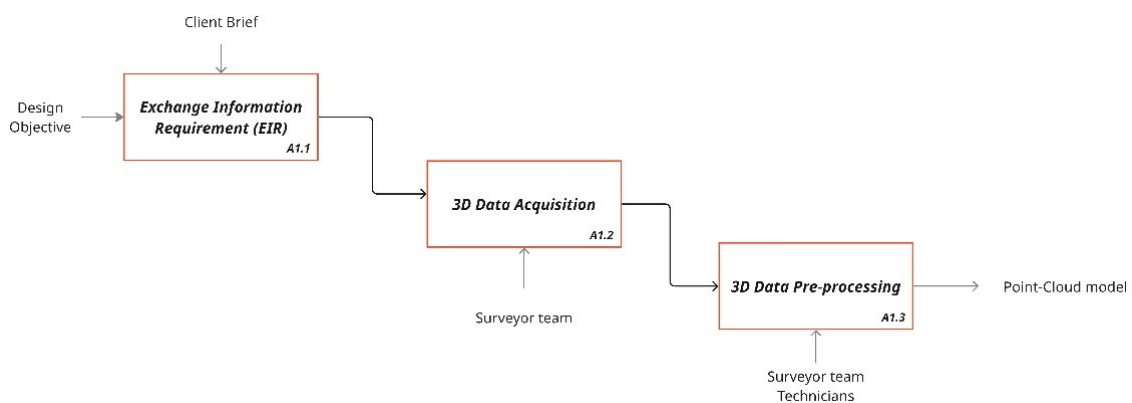


Figure 40: A1 (3D building survey) sub-process model (Author)

A1 (Figure 40), the 3D building survey process comprises 3 sub-processes as shown in the above diagram. After the design objective is received to start the workflow, generating EIR would be the first step. An EIR would explain in detail the information requirement and exchange for each main process and sub-processes if needed. Following this initiative at the beginning of the workflow ensures cohesiveness of information, and efficient data management and helps to define the targets and deliverables. 3D data acquisition and data pre-processing activities such as denoising, registration of clouds, and decluttering could be the next steps to output a useful point cloud model to complete the A1 stage.

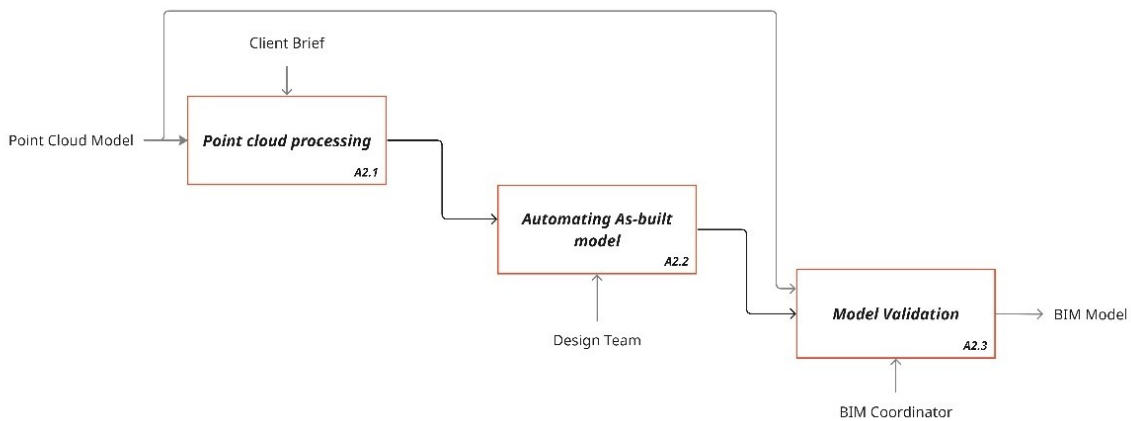


Figure 41: A2 (Automated as-built BIM) sub process model (Author)

Figure 41 denotes the sub-processes of the A2 stage, automating the as-built model. The proposed model suggests point cloud processing as its first step which states actions on the cloud file to extract information to build the BIM model. Point cloud normal estimation, slicing, object detection, plane detection, and poison surface reconstruction are a few operations that are useful to generate information based on point cloud files. The next step would be automating the as-built model creation with a robust process that would eliminate the tedious labor-intensive process of manual BIM modeling. As for the facade renovation task, the BIM model would only require some elements in the building, namely, external walls, windows and doors, extrusions on the facade such as slabs, equipment and installation,

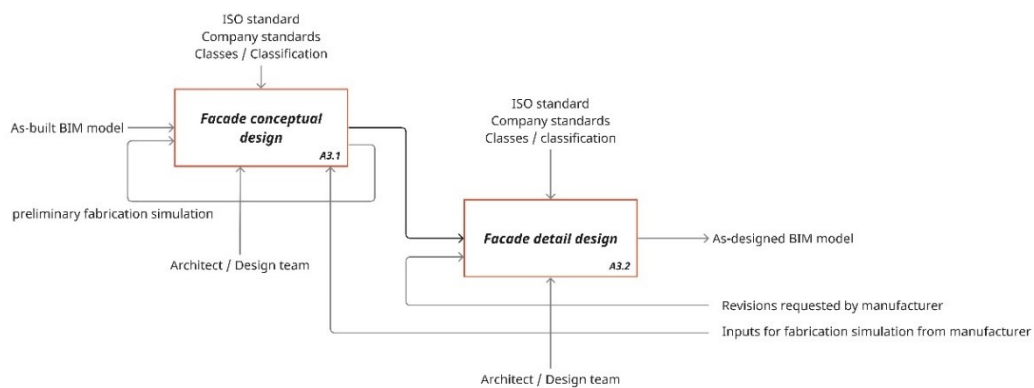


Figure 42: A3 (Facade design) sub process model (Author)

stairs or steps, and ceiling or roof. The model validation step would take the previous point cloud model to guarantee the accuracy of the generated BIM model to ensure the information exchange.

The third stage of the main TO-BE process map, A3 (Figure 42), could be identified as a combination of two stages for ease of understanding. The proposed method suggests that a preliminary fabrication simulation within the facade conceptual design stage would significantly increase efficiency while providing a fabrication-aware design. A simulation could effectively prevent a significant design alteration, guaranteeing that manufacturing input leading to design adjustments remains confined to a minor modification, manageable within a short amount of time.

For ease of understanding, the subprocesses in the manufacture and assembly stage are explained in two process maps. The manufacturing industry already has robust workflows and processes that drive efficiency, sustainability, and timely project completion. Hence, the proposed model denotes the current information flow majorly with the highlight of inputs for fabrication simulation for the design stage.

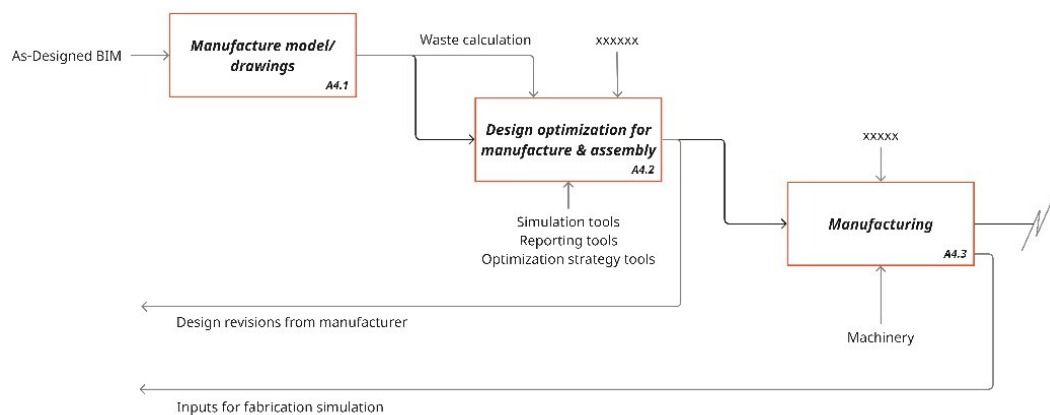


Figure 43: A4 (Manufacture and Assembly) sub processes part 1 (Author)

Figure 43 showcases the inputs that would be required from the manufacturing to the design stage. This information flow and exchange is crucial to increase efficiency and achieve the objectives of the TO-BE model.

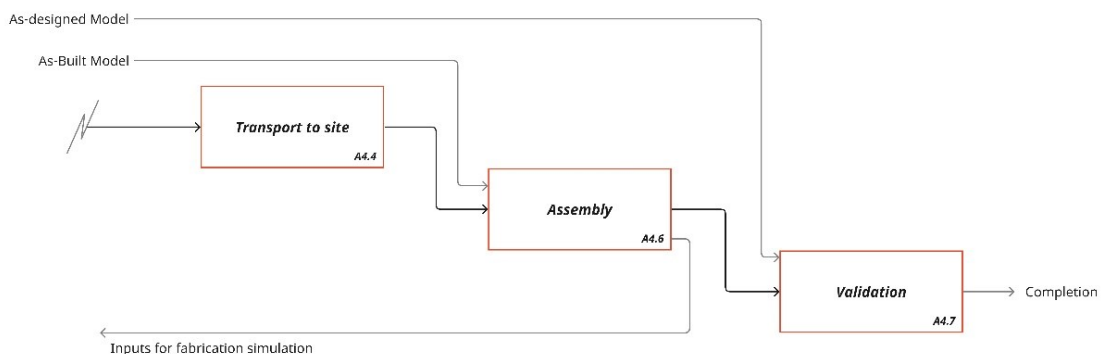


Figure 44: A4 (Manufacture and Assembly) sub processes part 2 (Author)

The rest of the subprocesses in the manufacture and assembly stage shown in *Figure 44* would have the as-designed model and as-built BIM models as inputs for the validation and assembly process respectively. As-built BIM and point cloud could be useful for installation activities on-site with the use of AR/VR technologies as well and validation of the assembly could be done using the as-designed model.

4.2.2 Digitization Strategy

It is useful to create a strategy for the whole workflow to summarize and identify key aspects of the above section. The discussed strategy in *Figure 45* is based on a vision for the workflow and lists down key objectives of the proposed digitization.

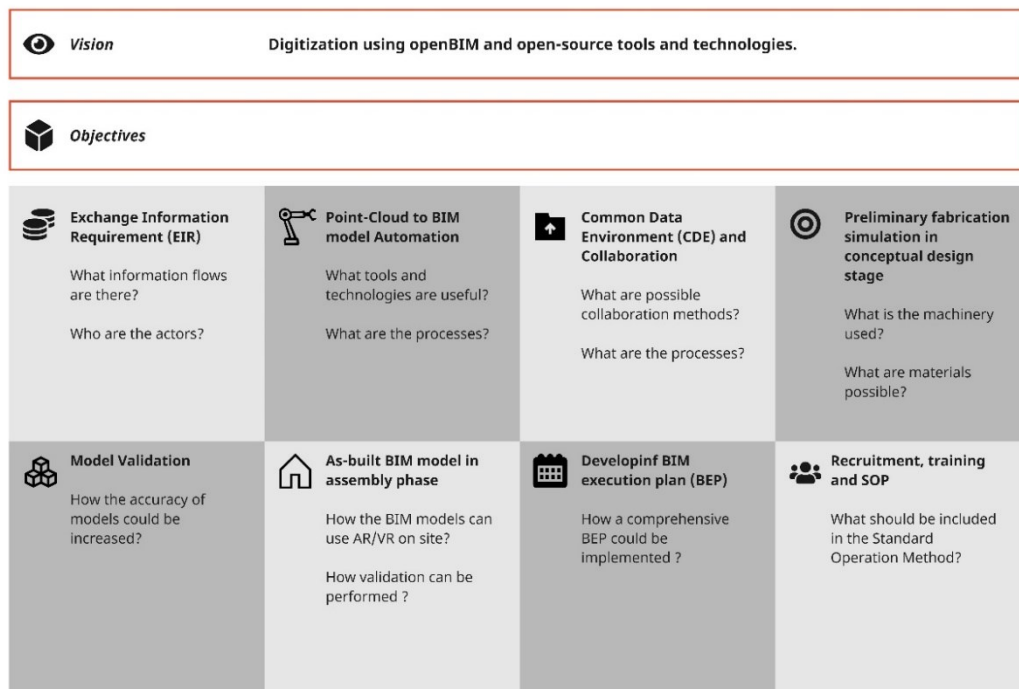


Figure 45: Digitization strategy for the case study (Author)

The vision for digitization is to open-source and openBIM tools and technology. This specifies the approach by which the objectives should be fulfilled. It sets out a clear direction for the application of the workflow. Typically this strategy is based on the OIR and PIR from the appointing party.

The above objectives comprise management, technological, and people aspects in their execution as well as they are connected to achieving digitization. EIR at the beginning of the project would provide useful insights into information requirements and management of later steps. Preparing BEP will clearly state how these objectives would be achieved, by whom, and at which point in the project. Point cloud to BIM automation, CDE and collaboration, simulation, and validation would require technological interventions. In the assembly phase, the as-built BIM could be useful for accurate installation with visualization technologies and cross-platform collaboration.

These objectives could be further explored when used in its application. Identifying technology, actors, processes, and information on each objective to fulfill the vision is the key takeaway from the digitization strategy explained here.

4.3 Proposed Workflow

The following section of this research focuses on developing EIR, point-cloud to BIM automation, and collaboration through the case study. These three objectives are priorities in the implementation of the workflow capturing information requirement and exchanges with technology and tools for automation of as-built BIM and collaboration based on openBIM and open source.

4.3.1 Exchange Information Requirement

The section identifies the PIR as a derivative of the digitization strategy. Based on the strategy, research proposes an EIR comprising all information requirements concerning all appointments across the project, a single coherent and coordinated series of information requirements. The following EIR diagram captures the information exchanges between parties. All these exchanges ideally would happen in the Common Data Environment set forth.

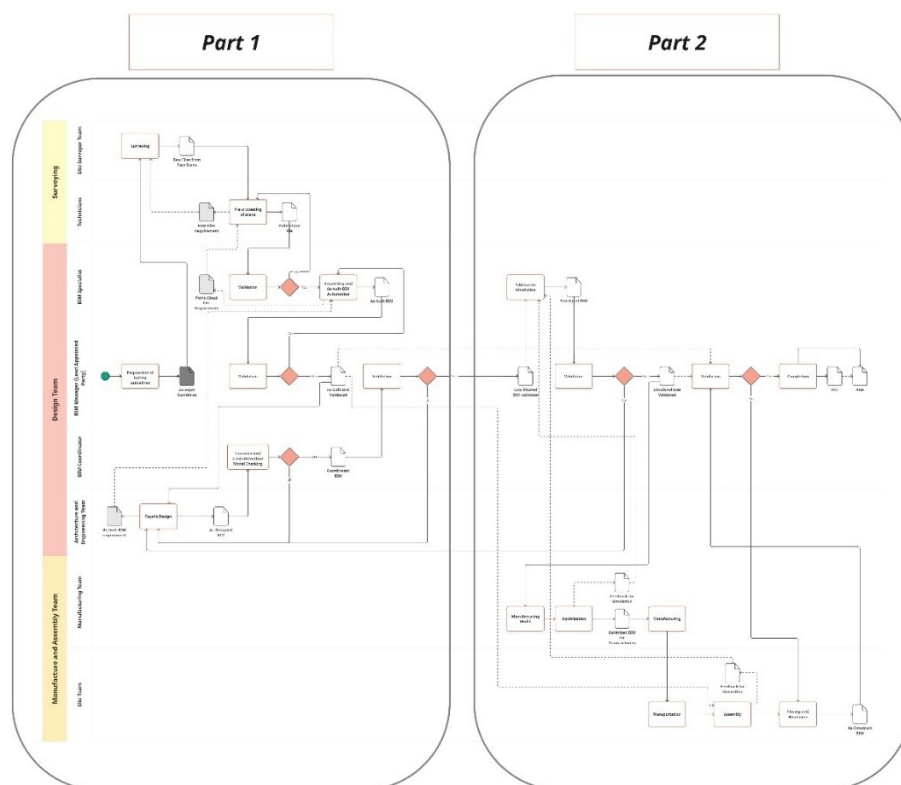


Figure 46: EIR for the proposed workflow divided to two parts (Author)

Figure 46 shows the full picture of EIR for this workflow and is divided into two parts. Those two parts are explained in detail therein.

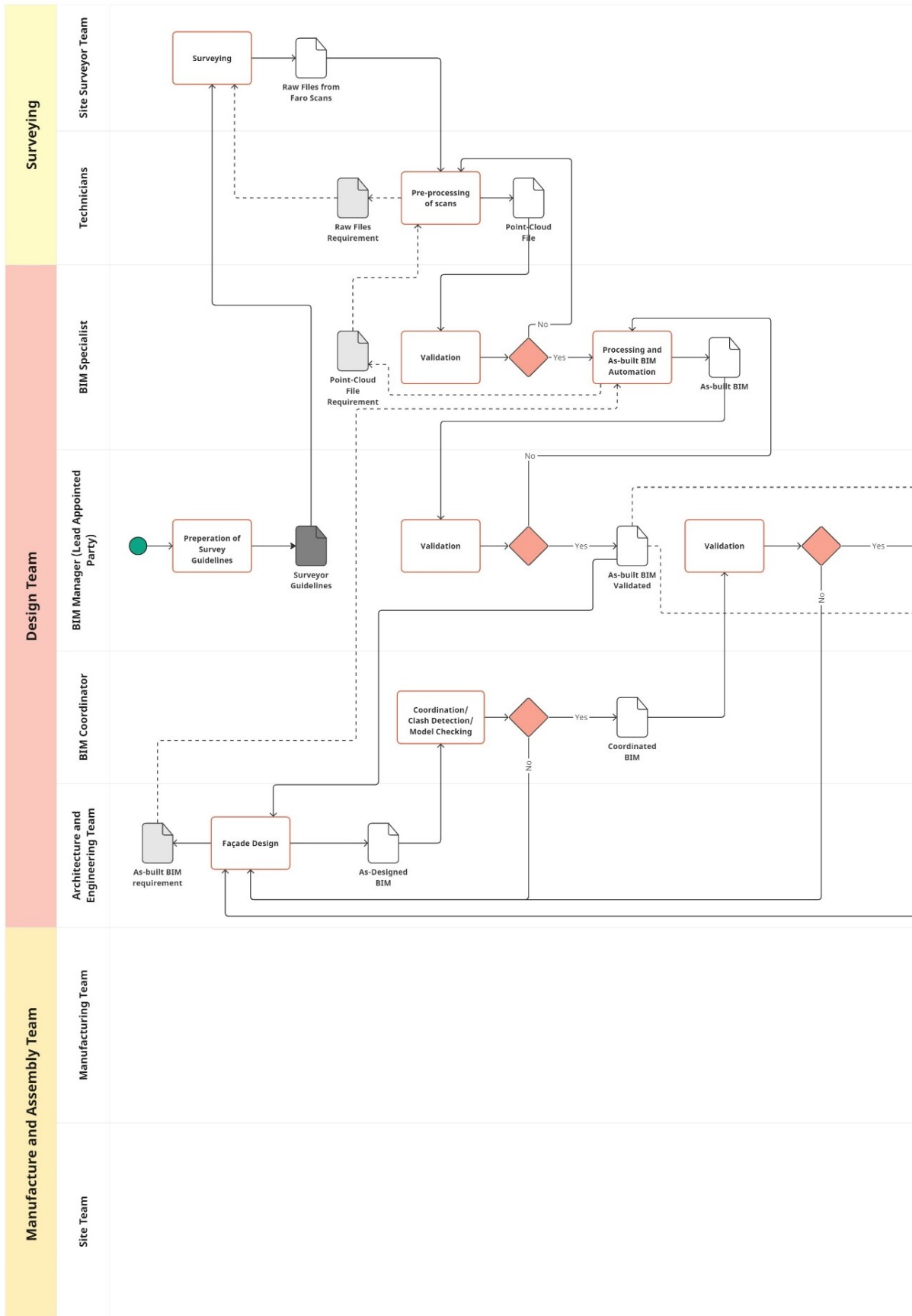


Figure 47: Part 1 of the EIR for the proposed workflow (Author)

Part 1 of the information exchange diagram (*Figure 47*) explains the information requirements and exchanges from the inception of the survey guideline until the validation of the coordinated BIM model by the BIM manager. According to the existing workflow, the teams are categorized into three parts mainly, the Surveying team, the Design team, and Manufacture and Assembly team. The surveying team comprises the site staff carrying out the 3D building surveying and technicians who do the preprocessing of data. The design team is led by the lead-appointed party, the BIM manager, and includes the BIM specialist, BIM coordinator, and architecture and engineering team. The manufacturing team and site team are included in the last category.

The surveyor guideline put forward by the BIM manager mentions the requirement for a surveying team on the ground and it includes details such as LOD/LOIN (Level of Information Need), and accuracy of data. Raw files from the survey are transferred to technicians for preprocessing tasks such as registration, and noise filtering, and the requirements of these raw files are put forward by the technicians to the site survey team before the survey on site. Technicians issue the point cloud file to the BIM specialist for validation.

BIM specialist therein does the processing of files and automation of as-built BIM. Here the BIM specialist should put forward the required information on point cloud to technicians before issuing the point cloud. The as-built BIM is then validated by the BIM manager and the validated as-built BIM is issued to facade design. Here the requirement for an as-built BIM is put forward by the architecture and engineering team to the BIM specialist before issuing of as-built BIM.

The Facade design team works with the BIM coordinator for model-checking activities with the as-designed BIM and the BIM coordinator issues the coordinated BIM for validation by the BIM manager.

Figure 48 shows the information exchanges from the coordinated BIM model validated by the BIM manager until the issuing of the Project Information Model and Asset Information Model.

The validated BIM model is put forward to the BIM specialist for fabrication simulation by the BIM manager and the feedback for simulation is to be provided by the manufacturing and assembly team. The simulated model is then validated and then put forward to the manufacturing team where they run the manufacturing model and optimization. The optimized BIM model for manufacturing proceeds onto manufacturing, transport to site location, and assembly on-site processes. In the assembly, the site team requires a validated as-built BIM model for installation and assembly. After closing out the site team issues as-construct BIM to be validated by the BIM manager. Here the BIM manager requires validated as-built BIM and validated simulated BIM.

The project completion denotes all information exchanges occurred accordingly and the BIM manager issuing the PIM and AIM to the appointing party.

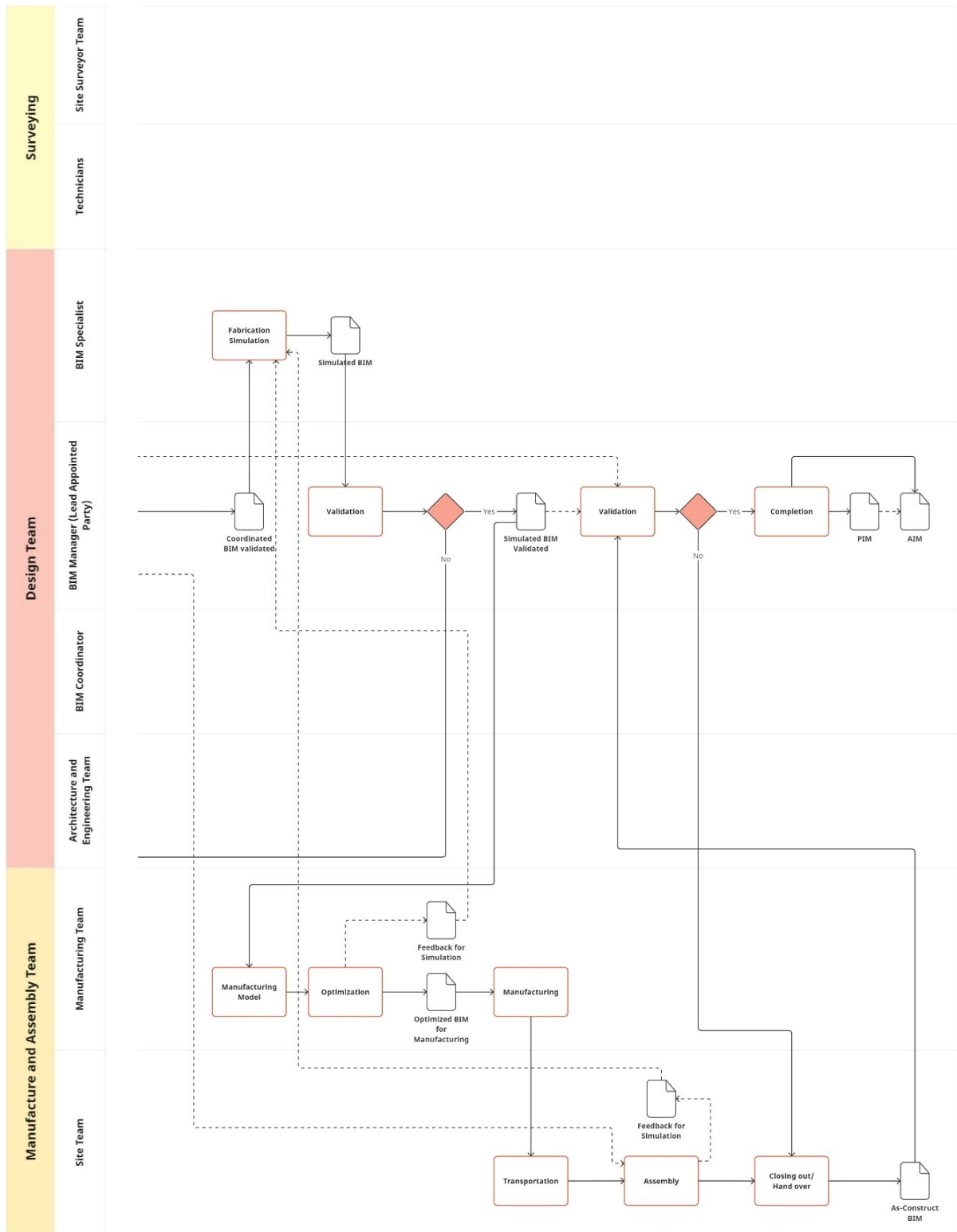


Figure 48: Part 2 of EIR for the proposed workflow (Author)

Each information requirement and exchange guarantees the issuing of AIM and PIM as ultimate goals which fulfill the objectives of the digitization strategy. In this case study, activities until the issuing of the point cloud file are already carried out and the appointed party has validated the point cloud as well. As the next step in the workflow, the following sections will focus on the BIM specialist activity of processing and as-built automation activity to issue the as-built BIM.

The point cloud to BIM operation is carried out using Blender, BlenderBIM, and IFC standards.

4.3.2 Point Cloud to BIM

First, the point cloud of the case study needs to be imported to Blender. Blender natively supports the import of PLY format, FBX format, or OBJ format file types but more common file formats of point cloud data such as XYZ, LAS, or E57 currently cannot be imported using native import (*Figure 49*).

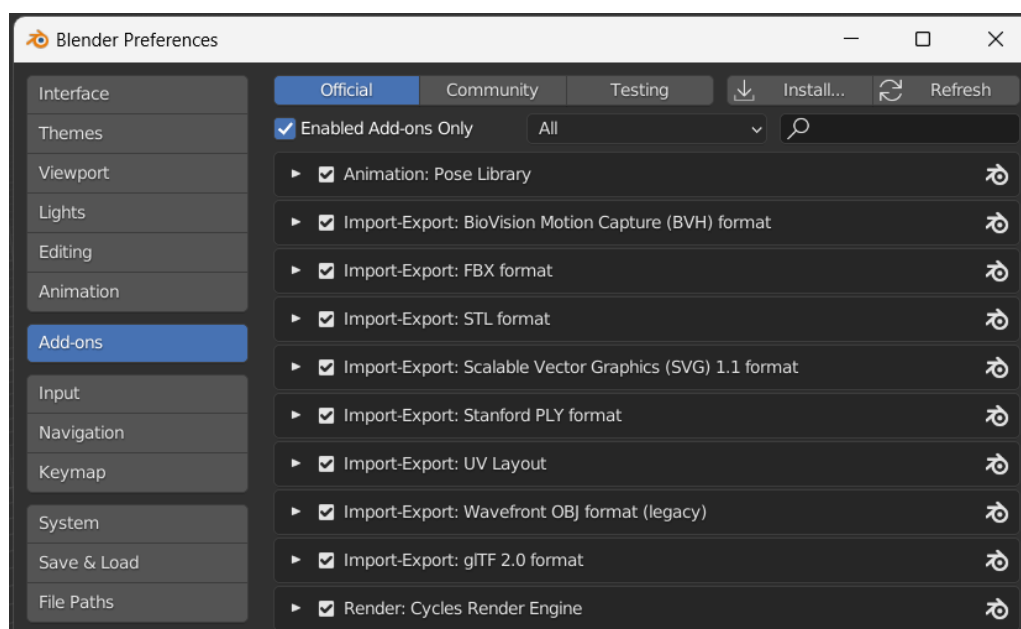


Figure 49: Natively supported file formats for Blender such as PLY, SVG, OBJ (Author)

Some add-ons support common point cloud file formats with free and paid versions. The most notable paid add-on is Point Cloud Visualizer by Jakub Uhlík. It offers operations such as visualizations, editing, translations, cleaning, simplifying, Boolean operations, meshing, downsampling, and surface reconstruction to point clouds. It also offers importing of LAS and E57 file formats which makes it a versatile add-on (*Point Cloud Visualizer - Documentation*, n.d.).

Several free-to-use addons enable point cloud operations in Blender put forward by the Blender community. The research investigates two of those add-ons that support the import of point cloud files.

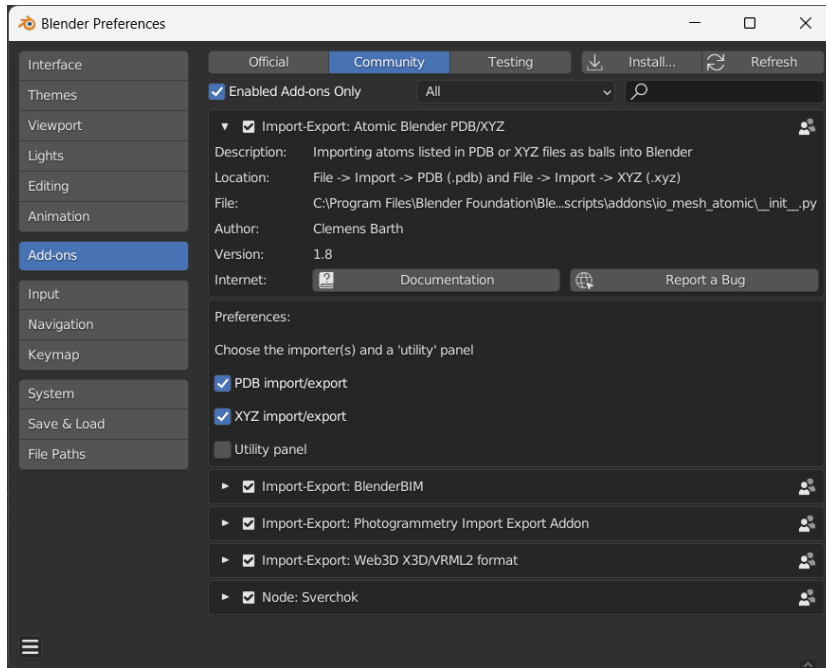


Figure 50: Atomic Blender addon details (Author)

The Atomic Blender addon shown in Figure 50 supports the import of PDB and XYZ file types but when put into work with Blender the import did not work since primarily this addon is focused on visualizing atomic structures and their data.

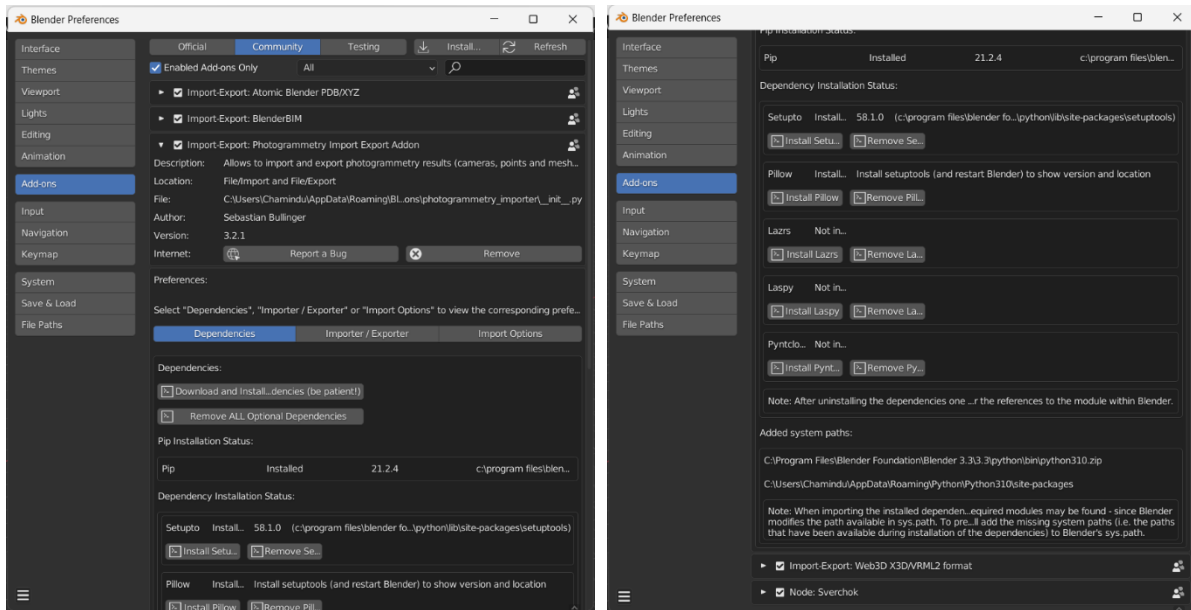


Figure 51: Details of Photogrammetry Import Export addon for Blender (Author)

The second addon is the Photogrammetry Import Export addon (*Figure 51*) which enables a range of point cloud file type imports and their visualizations (Bullinger et al., 2021). This addon requires many dependencies and for a new user, it comprises easy-to-follow guidelines to set up the addon. But this addon is limited to visualization of point clouds.

In this case, open-source software CloudCompare is used to process the case study point cloud LAS file and convert it to PLY format. In CloudCompare, a cloud sub-sampling operation is carried out to reduce the file size from 38, 087, 385 points (944MB) to 6, 650, 000 points (*Figure 52*). This value for the number of points is a guideline to keep the file size around 100MB.

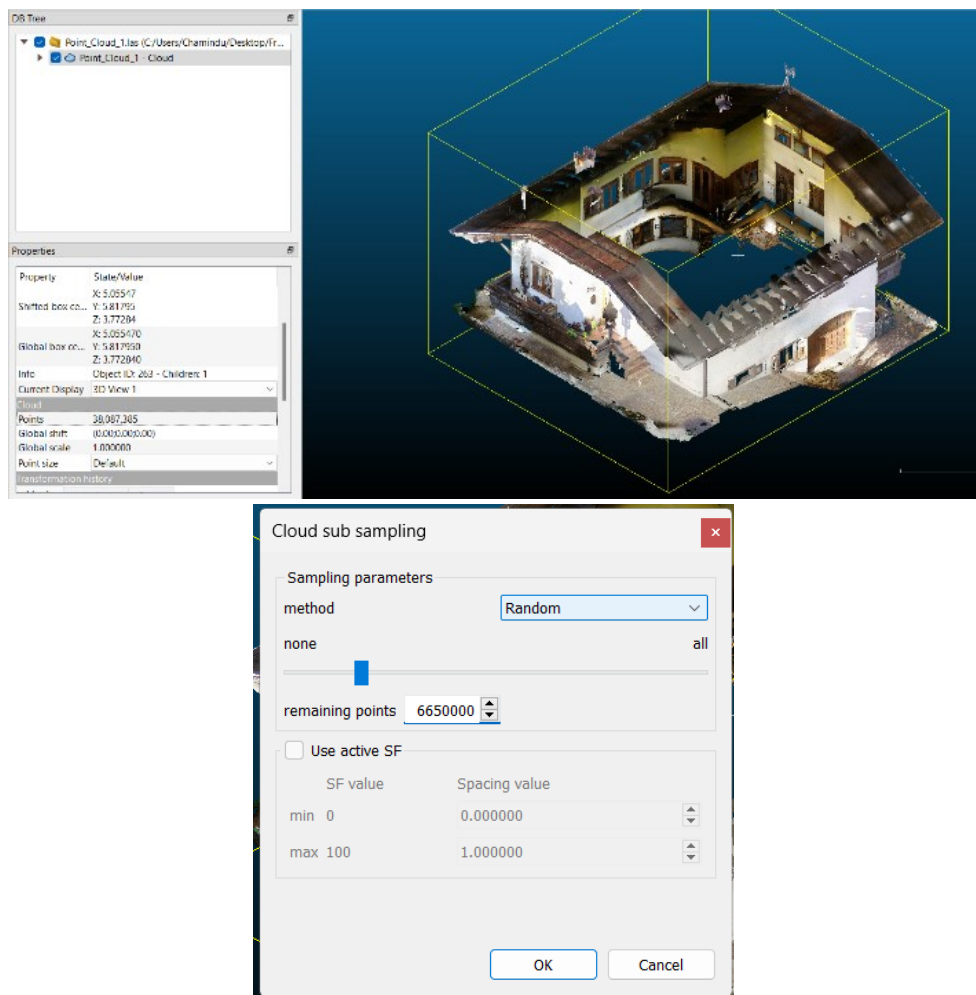


Figure 52: Cloud sub sampling in CloudCompare (Author)

Then using the segment tool noise could be removed manually by drawing regions around them and segmenting from the point cloud. This results in multiple files containing those segments. Next, the point cloud is processed for normal calculation and exported to PLY. The normal calculation is used for object recognition and surface calculations in point clouds.

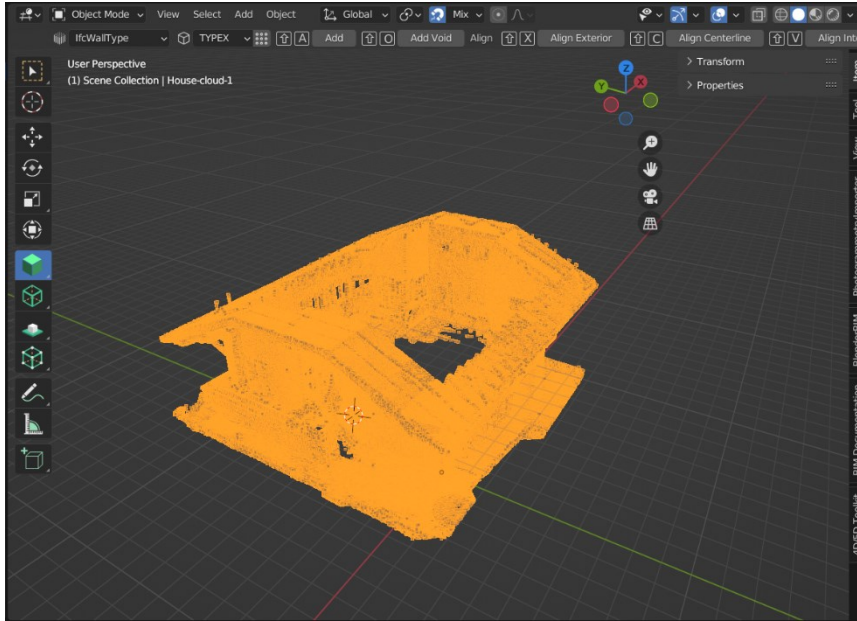


Figure 53: Converted point cloud imported to Blender (Author)

The points could be exported to Blender (Figure 53) and information could be extracted using Geometry Nodes in Blender (Figure 54, Figure 55). The operation would be to use points to create a volume, merge the volumes, and convert to mesh. Since the point operations in Blender are currently limited to operations such as converting to/from meshes (Blender, n.d.). Below geometry node tree below shows the operation and Boolean intersection with a plane in a specific height, so the points in a specific could be extracted. The model changes from the initial point cloud view to extracted points using the Boolean intersection shown.

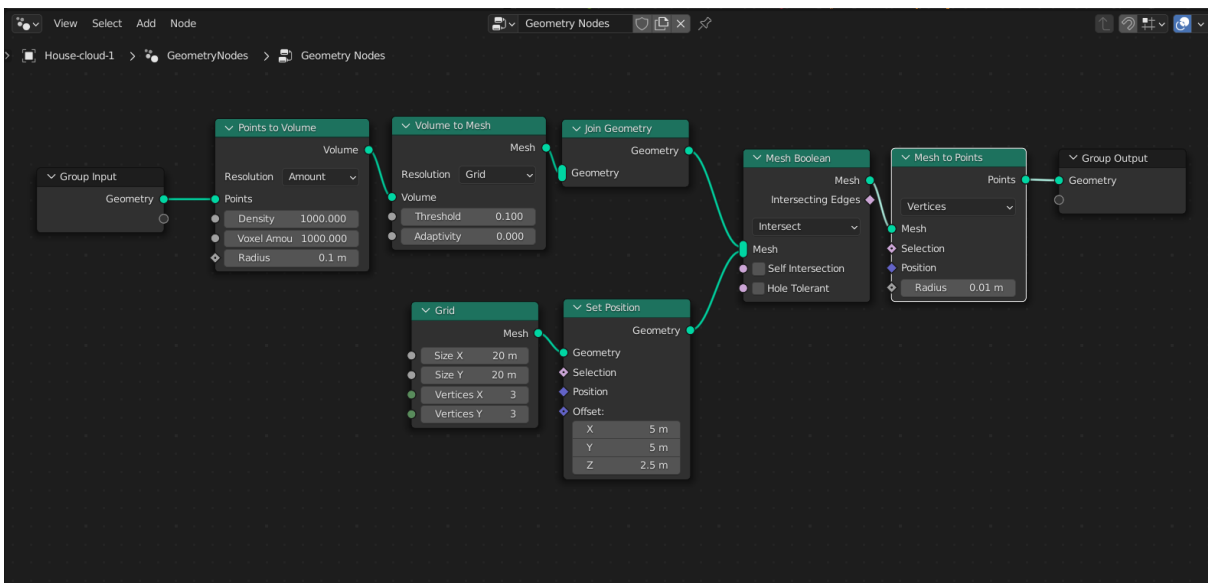


Figure 54: Geometry node tree to create a mesh from points in Blender (Author)

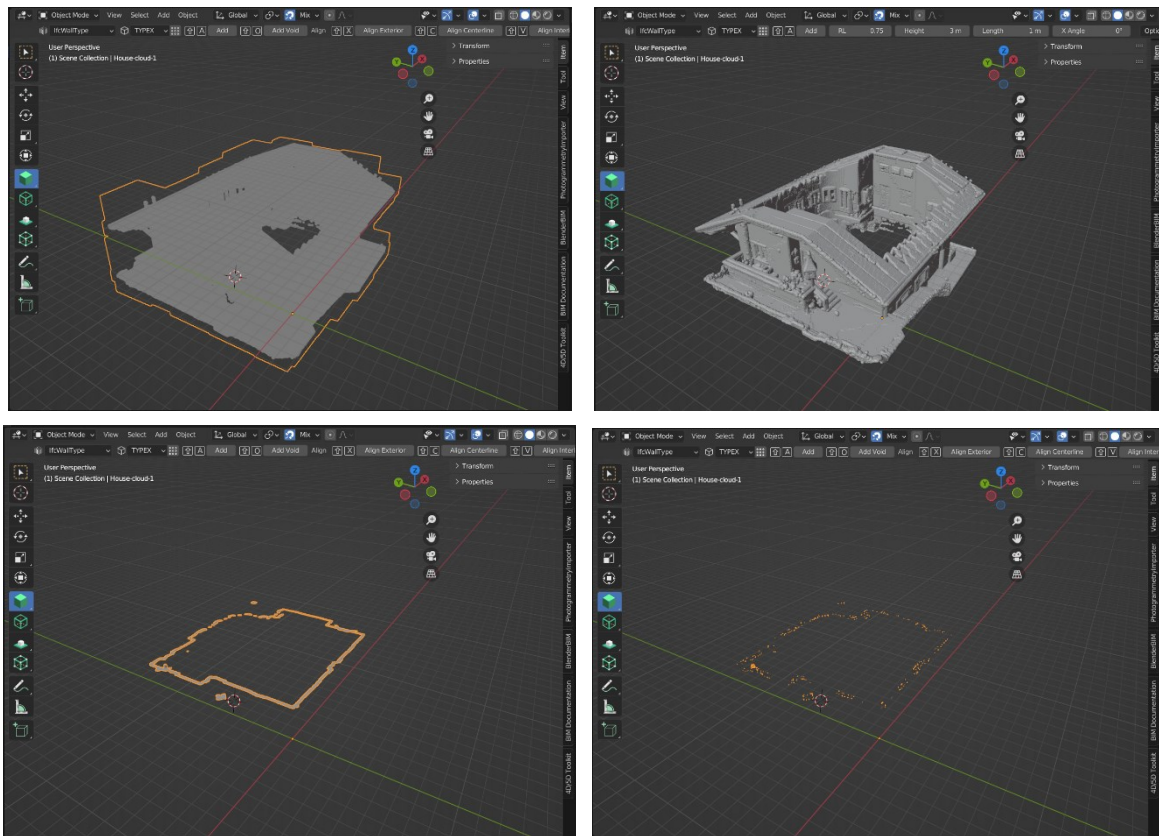


Figure 55: From top left to right and bottom left to right - 1. Points to volume, 2. Volume to mesh, 3. Boolean intersection with plane, 4. points containing the intersection (Author)

For BIM operations in Blender, BlenderBIM and Ifcopenshell are the main two methods to use. This research will use the developer version “blenderbim-230731” from Github for the next sections. First, the research investigates the project setting up and BIM operations in BlenderBIM and shows a workflow of manually creating BIM objects from the point cloud.

Figure 56 shows the starting window of Blender with the addon enabled where the project overview tab on the right allows the user to create a new IFC project or from the menu bar also it is possible. This allows the user to familiarize themselves with using BIM authoring software. The user can choose the IFC schema, units, and template when creating a project. When a project is created the outliner on the left shows the IFC spatial tree. On the right side, project setup, object information, and geometry information can be seen.

With the familiarity of the IFC schema, the tools are useful to create, modify, analyze, and include metadata to IFC. The addon is in the development stage because of this it does not show the full potential of the IFC schema with its objects and relationships.

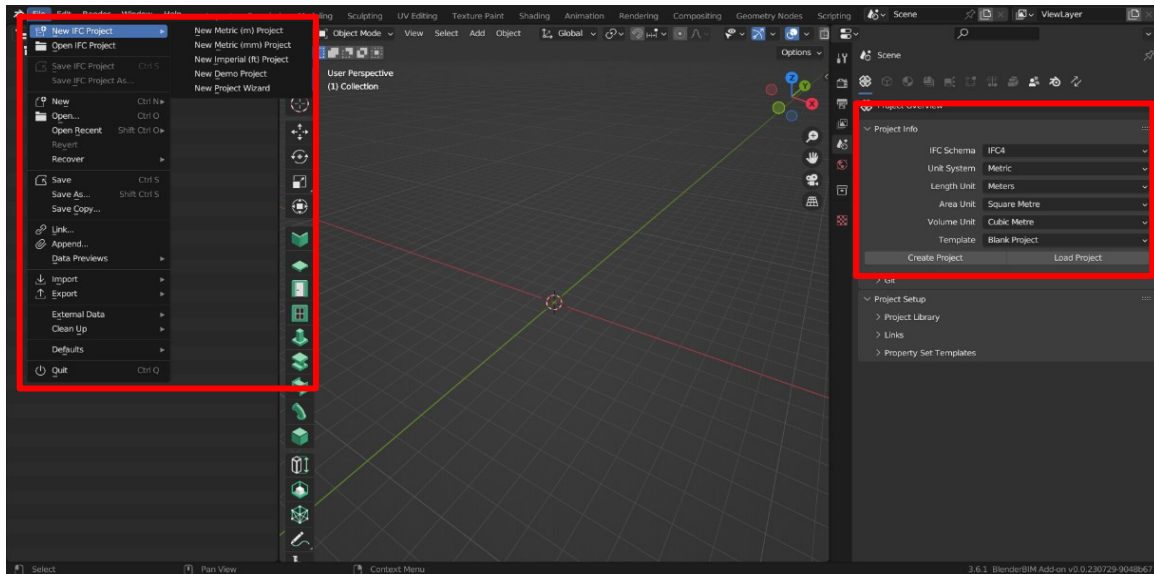


Figure 56: Creating an IFC in Blender with BlenderBIM (Author)

In IFC spatial tree Types under Ifcsite contains the BIM objects (Figure 57). The software allows the loading of past IFC projects from BlenderBIM or IFC exports from other applications and the use of the material or types/objects in the current IFC project. Thus, every IFC type created for a specific project could be used in future projects automatically creating a BIM object library. All the property sets, classification data, COBie data, and material data come along with the loaded types.

A project element such as levels, grids, and objects will appear in the tree as per the spatial relationship in IFC and any parametric objects could be easily categorized into levels as well. Figure 58 shows the created grids on a two-level comprising project. The software allows the user to enrich the authored IFC file with metadata and geometric information for specific uses.

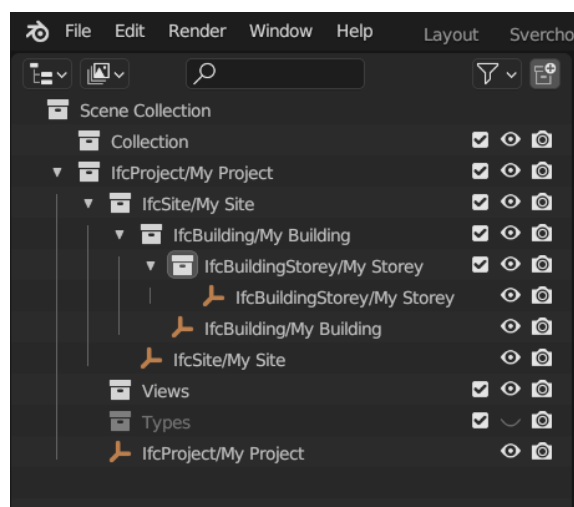


Figure 57: IFC spatial tree in BlenderBIM (Author)

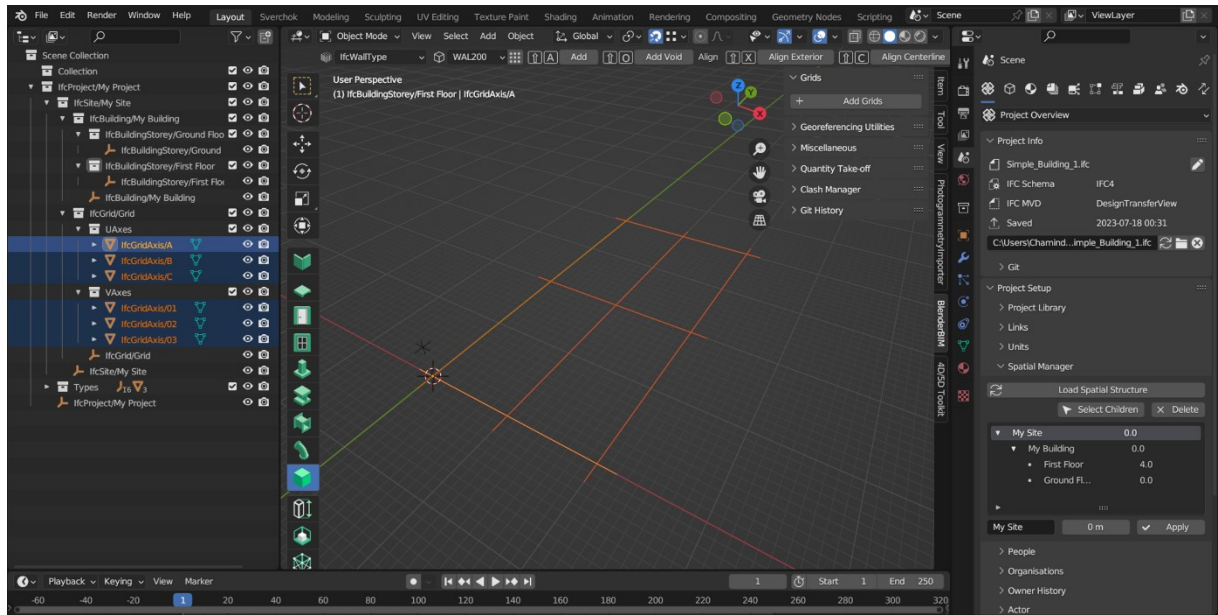


Figure 58: IFC project with grids and levels (Author)

Now with the information extracted from the point cloud, a wall is manually drawn with two windows (Figure 59). The project levels are created manually, and wall and window types are assigned to relevant levels. The saved IFC file is then viewed in Open IFC Viewer to ensure information exchange.

The geometrical representation of the wall is accurate. Quantity and property sets need to be assigned in BlenderBIM for a more informatic model.

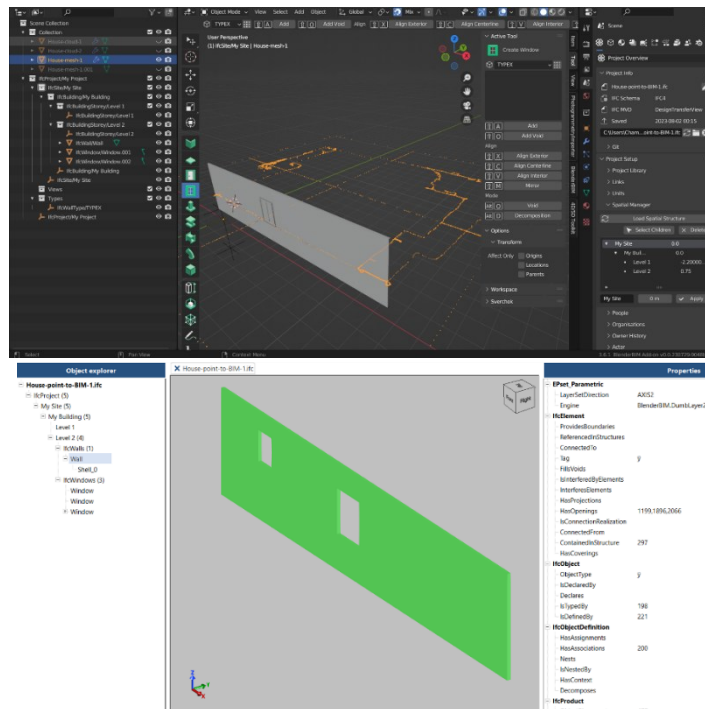


Figure 59: Manually drawn wall and windows in BlenderBIM and viewed in Open IFC Viewer (Author)

Blender holds the ability to use Python scripting and it is useful to look at the possibility of Ifcopenshell integration with Blender for BIM operations. *Figure 61* and *Figure 60* scripts are examples from C-Claus in GitHub. These open-source codes are run in Blender with the text editor function that allows to running of long scripts. Some scripts needed some work to make it run but the codes below run smoothly without any modification.

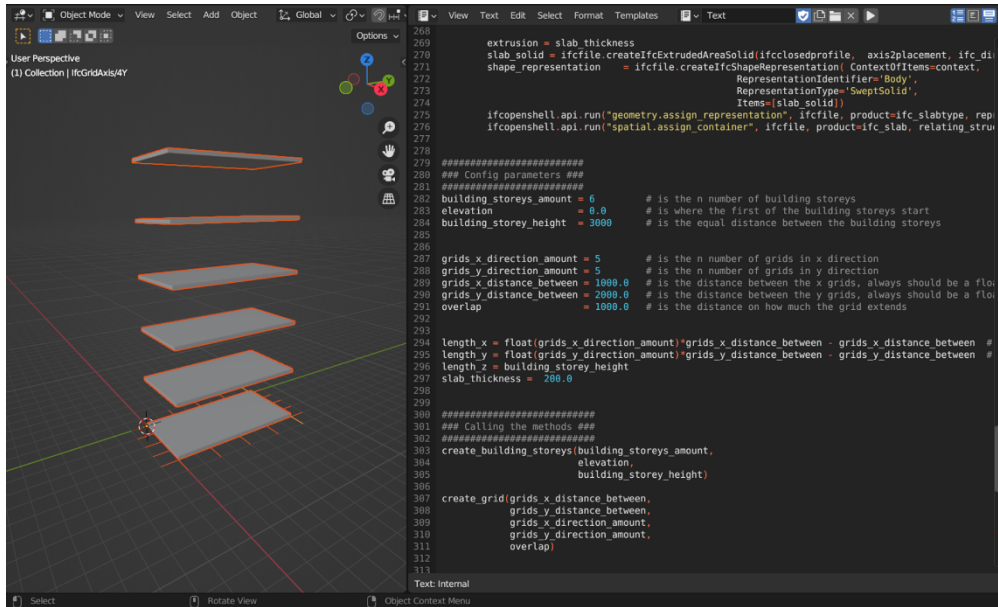


Figure 61: IFC project containing slabs, grids and levels created using scripting (Author)

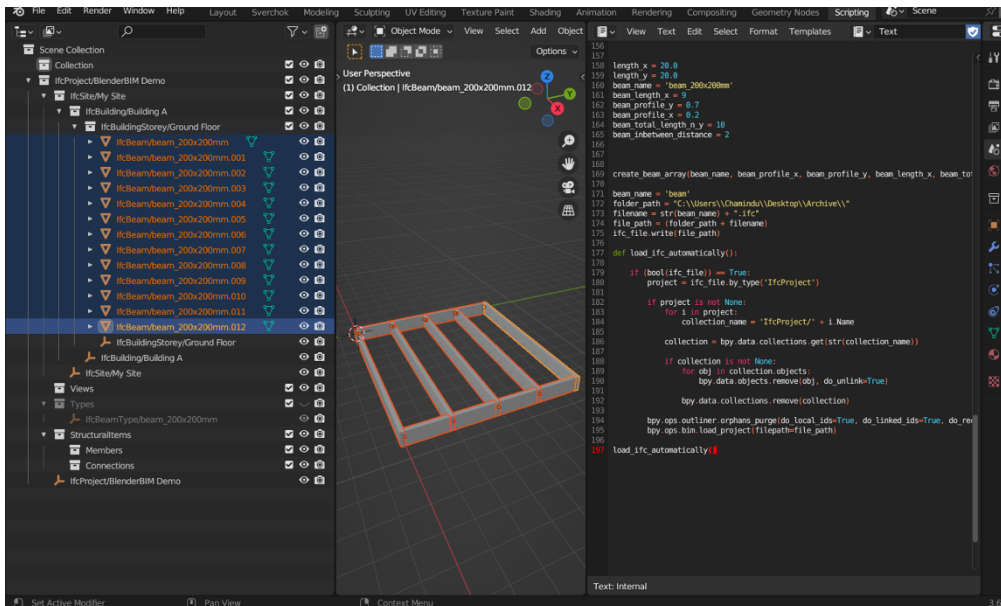


Figure 60: IFC project containing parametric beam system created using scripting (Author)

These two scripts investigate Ifcopenshell in Blender used to create a series of slabs with grids and parametric beam layout. Changing values in the number of levels, grid dimensions, or beam length triggers the script to act as the user wants and maintain the parametric behavior of each object and with each other. The script first writes an IFC file with Ifcopenshell and then imports it to Blender to make a real-time visualization of the script.

Open3D library is versatile in point cloud operations. Processed case study point cloud file from CloudCompare is used with the Open3D library in Visual Studio Code (*Figure 62*).

```
boxes = o3d.detect_plane_patches(  
    normal_variance_threshold_deg=30,  
    coplanarity_deg=75,  
    outlier_ratio=0.75,  
    min_plane_edge_length=3,  
    min_num_points=0,  
    search_param=o3d.geometry.KDTreeSearchParamKNN(knn=30))
```

planar segmentation

```
o3d.visualization.draw_geometries(geometries,  
    zoom=0.62,  
    front=[0.4361, -0.2632, -0.8605],  
    lookat=[2.4947, 1.7728, 1.5541],  
    up=[-0.1726, -0.9630, 0.2071])
```

visualization

```
for i in range(len(geometries)):  
    o3d.io.write_triangle_mesh(r"C:\Users\Chamindu\Desktop\Fran\House-cloud-1-plane-" + str(i) + ".ply",  
        geometries[i], write_ascii=False, compressed=False, write_vertex_normals=True, write_vertex_colors=True,  
        write_triangle_uvs=True, print_progress=False)  
  
for i in range(len(geometries)):  
    line_set = o3d.geometry.LineSet.create_from_triangle_mesh(geometries[i])  
    o3d.io.write_line_set(r"C:\Users\Chamindu\Desktop\Fran\House-cloud-2-line-" + str(i) + ".ply", line_set,  
        write_ascii=False, compressed=False, print_progress=False)
```

writing to mesh and point files

Figure 62: Open3D code used to visualize, planar segment, and write both line and mesh files with planes (Author)

The code in *Figure 62* is used to extract planar patches from the point cloud, visualize them, and write them into line and point files. Firstly, Numpy and Matplotlib libraries must be imported, and the relevant saved location of the point cloud is to be mentioned. The planar patches detection is carried away with default values of each six arguments.

The `normal_variance_threshold_deg` specifies the angle between normals of the points, `coplanarity_deg` sets allowed distribution from plane to points, `outlier_ratio` specifies maximum ratio of outlier in a plane before it is being rejected, `min_plane_edge_length` sets out the value for false positives, `min_num_points` specifies number of points needs to be there to fit a plane and how deep the octree is (Zhou et al., 2018).

The visualized point cloud is shown in *Figure 64*. The script detects 28 patches and output files in PLY formats are viewed in Blender as well.

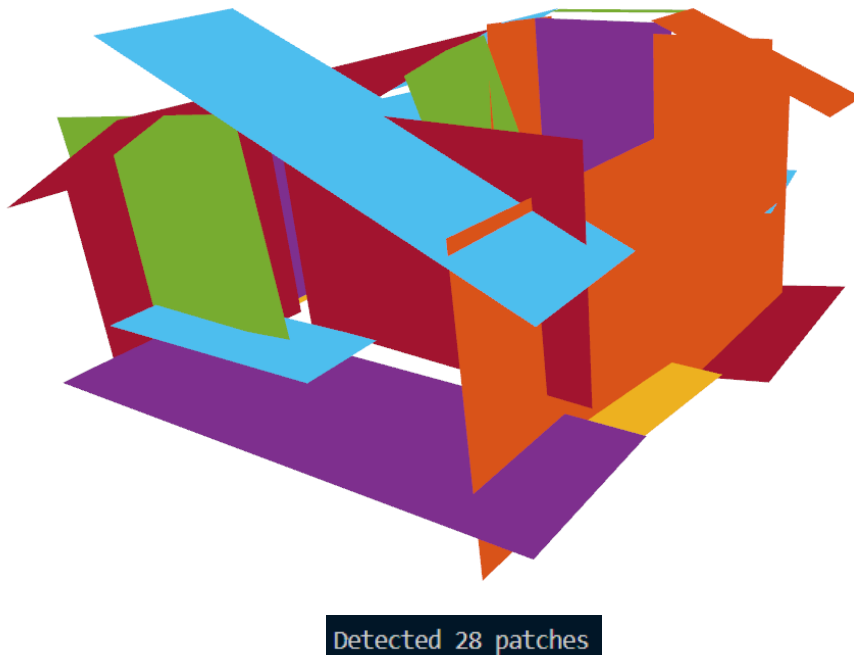


Figure 64: Planar patch visualization of case study (Author)

With the loop function in the Python script, planar patches' mesh and point files were written down as individual files. When imported to Blender, all the files are in their accurate location relative to each other as the visualized point cloud (*Figure 63*). Information on these files corresponds to floors, walls, and ceilings as planes with a multitude of angles.

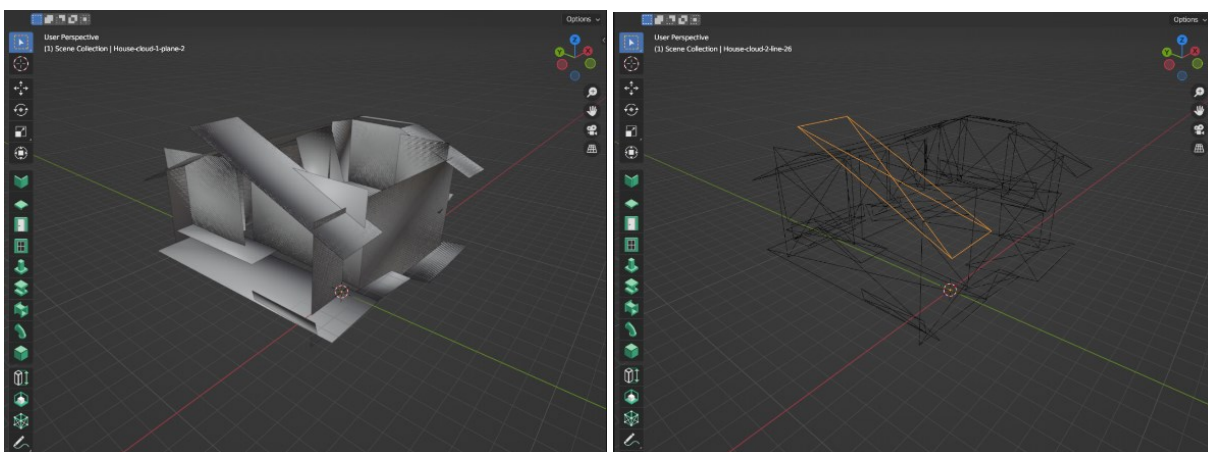


Figure 63 : Mesh (top) and point files (bottom) viewed Blender, contains information on 24 patches (Author)

4.3.3 Collaboration

This section investigates the IFC-based collaboration using BlenderBIM addon Blender. The latest developer releases use IFC Git incorporation into BlenderBIM and this feature will be used in the next stable release as well. IFC Git is a version control system and framework that uses remote repositories for collaboration.

The collaboration workflow can start after creating the IFC project in BlenderBIM and the IFC is saved locally. Then it allows the creation of a local Git repository in the Git panel. The created IFC file could be added to the repository, and it creates a master branch with saved IFC (*Figure 65*).

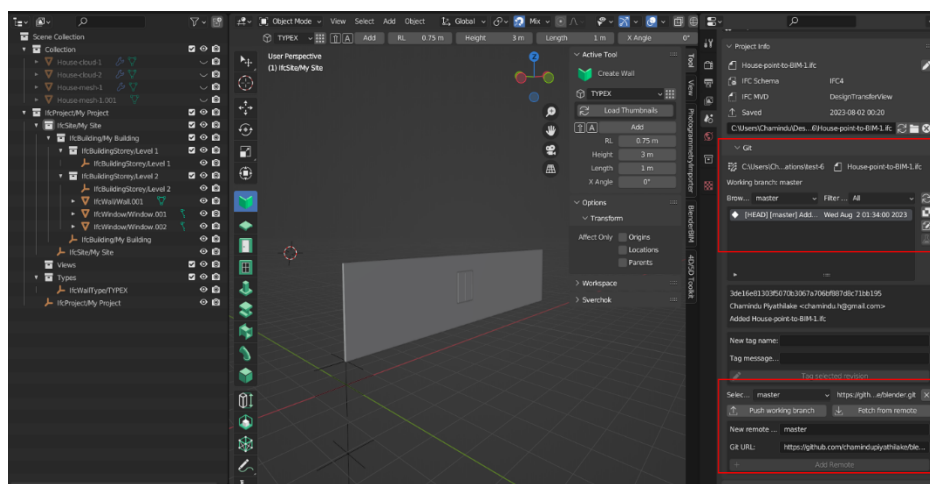


Figure 65: IFC Git in BlenderBIM and linking to remote repository (Author)

When a change is committed to the IFC and saved locally, the Git panel shows the option to commit changes (*Figure 66*). When clicking the 'show uncommitted changes', it can show differences from the original file.

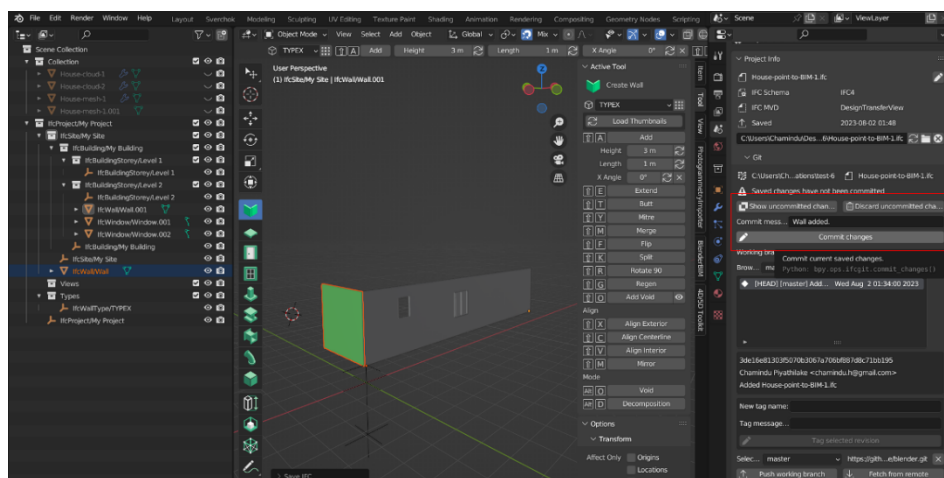


Figure 66: Showing changes to IFC (Author)

To save change, a commit message should be typed, and click commit changes. It adds changes into a branch and later merges with the master to include all changes. ‘Merging conflicts’ could happen if changes are made to the same geometry such as adding two windows in the same location of a wall.

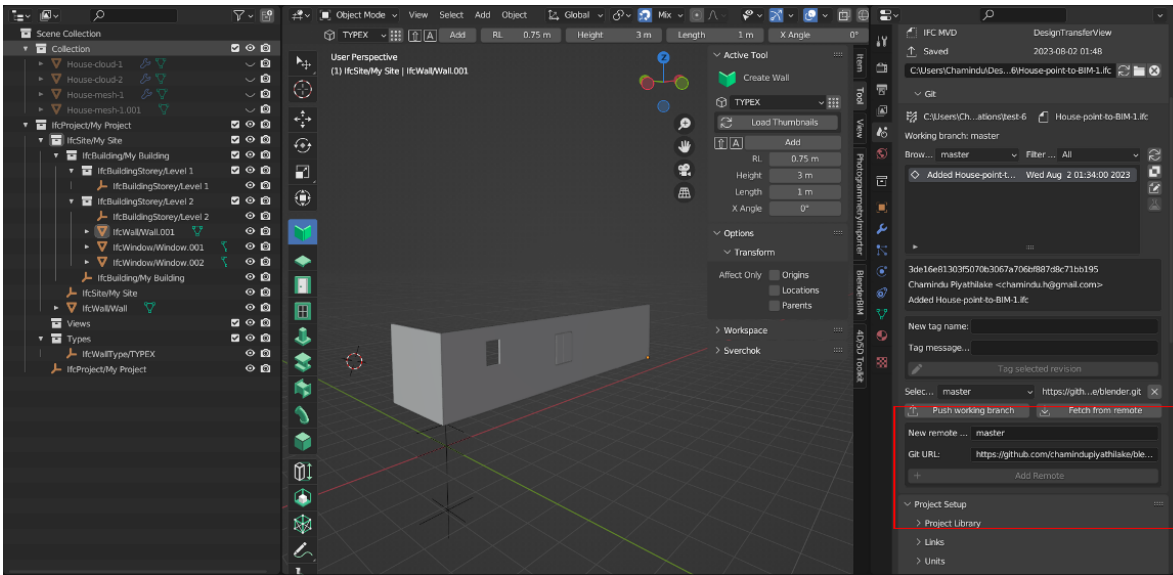


Figure 67: Adding remote repository information (Author)

The local repository is also capable of linking to a remote repository (Figure 67). As an example, a GitHub repository is created. In the GitHub repository, the https link is copied, and in Blender, a new remote name is given as ‘origin’ to say this is the main repository. The copied link is pasted and ‘add remote’ is clicked. Now the like-to remote repository is created.

Then any selected branch, as in master, for example, could be pushed into the panel. The working branch’s IFC commits, and history is updated in the remote repository. Now in GitHub, all the commits and changes in the IFC can be viewed in text format (Figure 68).

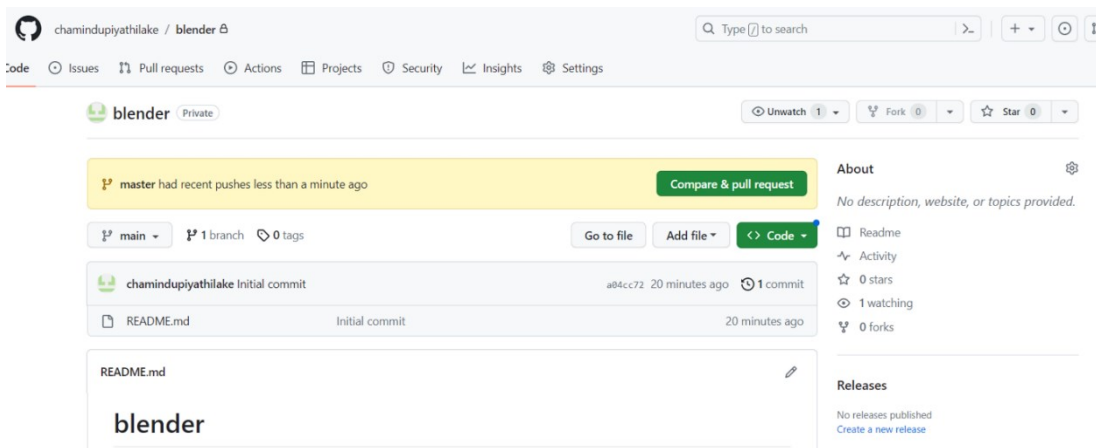


Figure 68: Remote repository in GitHub (Author)

4.4 Results

The provided point cloud lacked pre-processing operations namely data cleansing and normal estimation. For the scan to BIM process to take place these operations could have been performed. However, using the open-source software CloudCompare, point cloud was pre-processed for data cleansing and normal estimation.

Figure 39 to Figure 44 presented the TO-BE process model and its subprocesses that identified the processes and crucial data flows in the workflow. The digitization strategy developed using these models captured the requirement aligning to the vision of using openBIM and open-source approach.

EIR explained in Figure 46 to Figure 48, captured all instances of information requirements and exchanges in the workflow with their relationship to the actors involved.

Point cloud to BIM automation investigation revealed open-source software capabilities. Even though Blender allowed the import of point cloud, the file had to be converted to PLY format using CloudCompare first. Geometry nodes in Blender were limited to operations in the point cloud to convert it to mesh.

BlenderBIM was capable of drawing manually and creating an IFC file. The file is, however, limited in the information as property sets and quantity sets were not set within the software. Ifcopenshell was integrated with Blender using the text editor function and to further refine the scripts more specific knowledge of Python was required on Blender operations and the Ifcopenshell library.

Open3D extracted planar patches from the point cloud and wrote them into line and mesh files. Detected patches contain necessary panes corresponding to walls as well as the ceiling and floors.

IFC Git for collaboration worked with local and remote repositories. All the changes to the IFC file, history of changes, and commit messages are also integrated with both repositories.

4.5 Discussion

The proposed workflow comprising EIR, point cloud to BIM automation, and collaboration specified the base for the digitization strategy. This is helpful to future studies to carry on the tasks of further refining point cloud to BIM automation and fulfilling the rest of the objectives.

Software and programming literacy is one of the key areas highlighted in the case study application. Using open-source software was straightforward but using Python libraries required specific knowledge. This could be further strengthened by the need to reference coordinate points from planes in Ifcopenshell for walls.

To create a wall from these coordinates geometri creation in IFC schema and hierarchy is vital to consider. It is useful to consider `IfcCartesianPoint` in the `Ifcopenshell` script to incorporate the coordinate points from the exported point files of the point cloud planes (*Figure 69*).

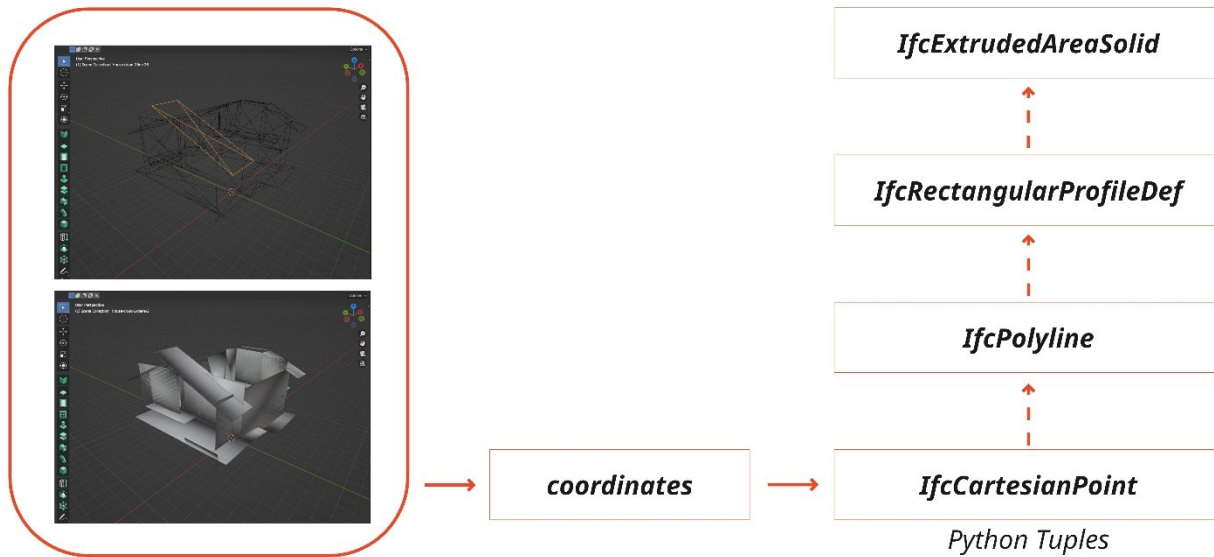


Figure 69: From detected planes of point cloud to wall creation (Author)

In IFC Git collaboration, it is possible to clone an existing repository from another user to BlenderBIM if it is permitted. Even if there is no permission, you can fork it, get the link, in BlenderBIM copy it create a remote git repository, select an empty local folder destination, and then click clone repository to open a copy of the repository through the cloned remote repository. By clicking refresh user can see all the history, commits, and comments.

Now the user can change the IFC, save the model, and push the working branch to the repository. Then in the remote repository, the user can make a pull request, where the other party can confirm the request. Then if the user again moves to Blender and selects 'fetch from remote', the history will be updated. Now in the branch browser, by selecting the origin master from a remote repository, the user can see revisions made by other users. If the user wants to see a specific revision, select the revisions, click the switch, and reload the IFC button, then the model gets updated with revisions by another user. It will also show if there are conflicts when confirming pull requests. By tagging the revisions and proper naming conventions, we can keep track of major file revisions as well as tags for phases, stages of the project, design stage, construction stage, etc.

With this approach, it's possible to do tasks such as scheduling, energy analysis, specification writing, and costing, separately with IFC files in the remote repository and modified access. Also, in BlenderBIM, when many revisions are merged, the user can go down to a certain point in revisions and see what has changed from that point to the current model, with highlighted objects in the model.

5 CONCLUSION AND FUTURE DEVELOPMENT

This research offers useful insights into a prevailing workflow in the AEC industry. Façade renovation is a common and well-established practice within this industry characterized by advanced technologies and a wide range of professionals integrated. Major components of this workflow encompass 3D point cloud surveys, parametric façade design containing complex geometries, manufacture, and assembly-oriented façade elements.

The digitization strategy and the proposed TO-BE process model envisioned a workflow comprising information exchanges across multiple professionals. This approach was the foundation for implementing operations in the case study. Lacking the information requirement and exchange was a major issue and the envisioned EIR captured the information flow within the appointed party. It laid the groundwork for individuals in the teams to identify their roles, the processes they are involved in, and the data flow they are part of.

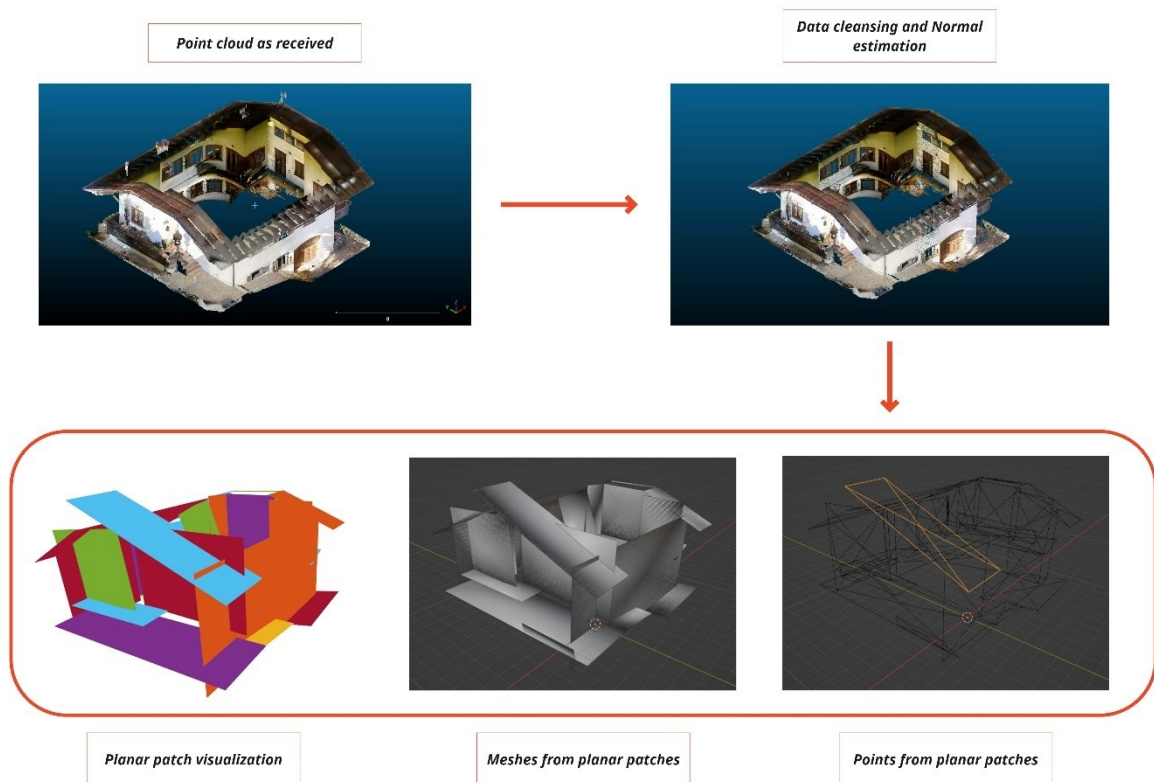


Figure 70: Point cloud operation process followed in the research (Author)

Point cloud to BIM operation envisioned using multiple workflows and the workflow comprising Open3D was successful. As shown in Figure 70, the received point cloud was first put into operations such as data cleansing and Normal estimation. The resulting point cloud was used in the Open3D script, which identified the planar patches and exported them as point and mesh files.

A possible complete workflow could be explained as shown in *Figure 71*. This method could be semi-automatic, and all tasks could be done in Blender with BlenderBIM add-on and text editor function. This approach leads to future research and generates a BIM model from detected planar patches.

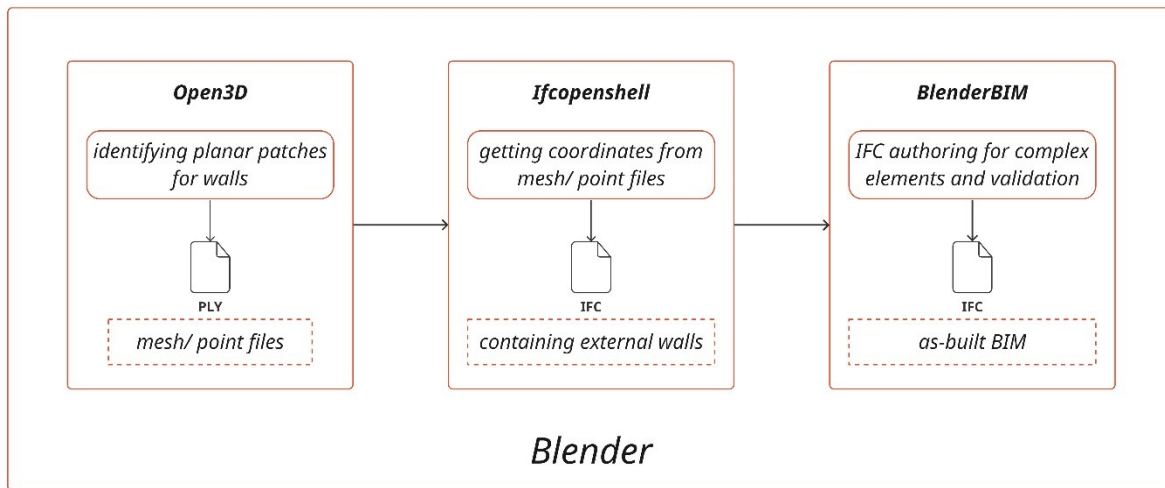


Figure 71: Workflow for point cloud to BIM using open-source and openBIM for future research (Author)

The workflow identified in this research provides valuable insights into achieving improved collaboration levels and interoperability through the use of openBIM and open-source approaches. Employing an IFC authoring environment in open-source software such as BlenderBIM represents a promising approach to improve BIM maturity as well. Although IFC Git collaboration provides a distinctive approach to collaboration, the integration of BCF would further ensure a holistic and comprehensive openBIM workflow.

Nevertheless, the notable challenges and obstacles in this workflow primarily centered around software literacy and programming knowledge as specialized skills. Furthermore, AEC personnel must know how to use BlenderBIM as a native IFC authoring platform as well as the git workflow for collaboration in a successful BIM implementation. Resources from open-source communities are useful to address these challenges, with a particular focus on technical knowledge and training resources. There are responsive and informative forums from these communities that are focused on many topics and explanations of specific tasks. Furthermore, there is open-source software to make a user-friendly interface for git workflow as well.

The research provides useful information on EIR, point cloud to BIM operation, and git-based collaboration which could be insightful to strengthen the manufacture and assembly of integrated façade renovation workflow in the industry.

6 ACRONYMS AND APPLICATION

2D	Bidimensional
3D	Three Dimension
AEC	Architectural, Engineering and Construction
AECO / FM	Architectural, Engineering, Construction, Operations and Facility Management
AIM	Asset Information Model
AIR	Asset Information Requirements
BCF	BIM Collaboraion Format
BEP	Building Execution Plan
BIM	Building Information Modelling
Blender	open-source software for 3D data
BlenderBIM	Blender software addon for BIM authoring
BPMN	Business Process Model Notation
bsDD	buildingSMART Data Dictionary
C++	a programming language
CAD	Computer-aided-Design
CAM	Computer-Aided Manufacturing
CDE	Common Data Environment
CloudCompare	open-source software for 3D data processing
CNC	Computer Numerical Control
COBie	Construction Operations Building Information Exchange
DfMA	Design for Manufacture and Assembly
E57	a vendor-neutral format for exchanging and storing 3D data
EIR	Employer Information Requirements

FBX	format used to exchange 3D geometry and animation data
FreeCAD	open-source CAD software
Git	a distributed version control system
GNU GPL	GNU Public License
IDM	Information Delivery Method
IFC	Industry Foundation Classes
Ifcopenshell	a Python library for IFC operations
ISO	International Organization for Standardization
ISO-STEP EXPRESS	ISO for Standard for the Exchange of Product Model Data
Las	LASer format exchanges and stores 3D data
LiDAR	Light Detection and Ranging
LOD	Level Of Development
LOIN	Level Of Information Need
MEP	Mechanical, Electrical, and Plumbing engineering
MIT	Massachusetts Institute of Technology
MLS	Mobile Laser Scanning
MVD	Model View Definition
MVS	Multi-view Stereo
NC	Numerical Control
OBJ	a standard 3D format
OIR	Organizational Information Requirement
Open3D	Python library for 3D data processing
Open IFC Viewer	an open-source software for IFC viewing
PCD	Point Cloud Data format supports 3D data not supported by other formats

PDB	format of file containing atomic coordinates
PIM	Project Information Model
PIR	Project Information Requirement
PLY	Polygon File Format
PTS	a 3D points file format
Python	A programming language
Revit	BIM Software by Autodesk
RGB-D	Red, Green, and Blue - Depth
SfM	Structure from Motion
Speckle	Collaboration platform for 3D data
TLS	Terrestrial Laser Scanning
UAV	Unmanned Aerial Vehicle
VS Code	Visual Studio Code
XYZ	a point data file format
XYZN	a point data file format with the normals
XYZRGB	a point data file format with color

7 REFERENCES

- Achakir, F., El Fkihi, S., & Mouaddib, E. M. (2021). Non-Model-Based approach for complete digitization by TLS or mobile scanner. *ISPRS Journal of Photogrammetry and Remote Sensing*, 178, 314–327. <https://doi.org/10.1016/j.isprsjprs.2021.06.014>
- Alfieri, E., Seghezzi, E., Sauchelli, M., Di Giuda, G. M., & Masera, G. (2020). A BIM-based approach for DfMA in building construction: framework and first results on an Italian case study. *Architectural Engineering and Design Management*, 16(4), 247–269. <https://doi.org/10.1080/17452007.2020.1726725>
- BIM Workbench - FreeCAD Documentation*. (n.d.). https://wiki.freecad.org/BIM_Workbench
- Borrmann, A., König, M., Koch, C., & Beetz, J. (Eds.). (2018). *Building Information Modeling: Technology Foundations and Industry Practice*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-92862-3>
- Bullinger, S., Bodensteiner, C., & Arens, M. (2021). A Photogrammetry-based Framework to Facilitate Image-based Modeling and Automatic Camera Tracking. *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 106–112. <https://doi.org/10.5220/0010319801060112>
- Chen, Z., Zhang, W., Huang, R., Dong, Z., Chen, C., Jiang, L., & Wang, H. (2022a). 3D model-based terrestrial laser scanning (TLS) observation network planning for large-scale building facades. *Automation in Construction*, 144, 104594. <https://doi.org/10.1016/j.autcon.2022.104594>
- Chen, Z., Zhang, W., Huang, R., Dong, Z., Chen, C., Jiang, L., & Wang, H. (2022b). 3D model-based terrestrial laser scanning (TLS) observation network planning for large-scale building facades. *Automation in Construction*, 144, 104594. <https://doi.org/10.1016/j.autcon.2022.104594>
- CloudCompare - Open Source project*. (n.d.). <https://www.cloudcompare.org/>
- Daniotti, B., Bolognesi, C. M., Lupica Spagnolo, S., Pavan, A., Signorini, M., Ciuffreda, S., Mirarchi, C., Lucky, M. N., Andersson, B., Andersson, P., Valra, A., Madeddu, D., Chiappetti, J., Farina, D., O'Regan, B., O'Leidhin, E., Tahir, F., Mould, K., O'Donovan, S., ... O'Sullivan, D. (2021). An Interoperable BIM-Based Toolkit for Efficient Renovation in Buildings. *Buildings*, 11(7). <https://doi.org/10.3390/buildings11070271>
- Dong, Z., Yang, B., Hu, P., & Scherer, S. (2018). An efficient global energy optimization approach for robust 3D plane segmentation of point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 137, 112–133. <https://doi.org/10.1016/j.isprsjprs.2018.01.013>

Foundation, B. (n.d.). About. In *blender.org*. <https://www.blender.org/about/>

Funding & tenders. (n.d.). <https://ec.europa.eu/info/funding-tenders/opportunities/portal/screen/opportunities/horizon-results-platform/39659;resultId=39659;needList=7>

Hello, world! - BlenderBIM Add-on 0.0.220504 documentation. (n.d.). https://blenderbim.org/docs/devs/hello_world.html

IfcExtrudedAreaSolid. (n.d.). <https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD1/HTML/schema/ifcgeometricmodelresource/lexical/ifcextrudedareasolid.htm>

Industry Foundation Classes (IFC) - buildingSMART International. (2019). <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/>

Installation - IfcOpenShell 0.7.0 documentation. (n.d.). <https://blenderbim.org/docs-python/ifcopenshell-python/installation.html>

International Organization for Standardization. (2018). *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles (ISO 19650-1:2018)*. <https://www.iso.org/standard/68078.html>.

Introduction — Blender Manual. (n.d.). https://docs.blender.org/manual/en/latest/modeling/geometry_nodes/introduction.html

Joblot, L., Paviot, T., Deneux, D., & Lamouri, S. (2017). Literature review of Building Information Modeling (BIM) intended for the purpose of renovation projects. *IFAC-PapersOnLine*, 50(1), 10518–10525. <https://doi.org/10.1016/j.ifacol.2017.08.1298>

Kamari, A., Laustsen, C., Peterson, S., & Kirkegaard, P. H. (2018). A BIM-based decision support system for the evaluation of holistic renovation scenarios. *J. Inf. Technol. Constr.*, 23, 354–380.

Lu, W., Tan, T., Xu, J., Wang, J., Chen, K., Gao, S., & Xue, F. (2021). Design for manufacture and assembly (DfMA) in construction: the old and the new. *Architectural Engineering and Design Management*, 17(1–2), 77–91. <https://doi.org/10.1080/17452007.2020.1768505>

Martinez, A., Patterson, M., Carlson, A., & Noble, D. (2015). Fundamentals in Façade Retrofit Practice. *Procedia Engineering*, 118, 934–941. <https://doi.org/10.1016/j.proeng.2015.08.534>

- Moult, D. (2022). Native OpenBIM, and the rise of open source in AEC. In *AEC Magazine*.
<https://aecmag.com/collaboration/native-openbim-and-the-rise-of-open-source-in-aec/>
- OpenBiM Definition*. (2020, December 3). BuildingSmart International.
<https://www.buildingsmart.org/about/openbim/openbim-definition/>
- openBIM Definition - buildingSMART International*. (2020).
<https://www.buildingsmart.org/about/openbim/openbim-definition/>
- Point Cloud — Blender Manual*. (n.d.).
https://docs.blender.org/manual/en/latest/modeling/point_cloud.html
- Point Cloud Visualizer - documentation*. (n.d.). <https://jakubuhlik.com/docs/pcv/docs.html#libraries-compatibility>
- Pottman, H. (2009). Geometry and New and Future Spatial Patterns. *Architectural Design*, 79(6), 60–65. <https://doi.org/10.1002/ad.980>
- Pottmann, H. (2010). Architectural Geometry as Design Knowledge. *Architectural Design*, 80(4), 72–77. <https://doi.org/10.1002/ad.1109>
- Pottmann, H. (2013). Architectural Geometry and Fabrication-Aware Design. *Nexus Network Journal*, 15(2), 195–208. <https://doi.org/10.1007/s00004-013-0149-5>
- Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *BIM Handbook*. John Wiley & Sons, Inc.
<https://doi.org/10.1002/9781119287568>
- Said, H. M., Chalasani, T., & Logan, S. (2017). Exterior prefabricated panelized walls platform optimization. *Automation in Construction*, 76, 1–13. <https://doi.org/10.1016/j.autcon.2017.01.002>
- Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010a). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7), 829–843. <https://doi.org/10.1016/j.autcon.2010.06.007>
- Tang, P., Huber, D., Akinci, B., Lipman, R., & Lytle, A. (2010b). Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7), 829–843. <https://doi.org/10.1016/j.autcon.2010.06.007>
- Wang, C., Cho, Y. K., & Kim, C. (2015). Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. *Automation in Construction*, 56, 1–13. <https://doi.org/10.1016/j.autcon.2015.04.001>

Wang, Q., Guo, J., & Kim, M.-K. (2019). An Application Oriented Scan-to-BIM Framework. *Remote Sensing*, 11(3), 365. <https://doi.org/10.3390/rs11030365>

Wang, Q., Tan, Y., & Mei, Z. (2020). Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications. *Archives of Computational Methods in Engineering*, 27(2), 479–499. <https://doi.org/10.1007/s11831-019-09320-4>

Welcome! *Speckle Docs*. (n.d.). <https://speckle.guide/#speckle-web-app>

What is open source software? *IBM*. (n.d.). <https://www.ibm.com/topics/open-source>

What is Open Source Software (OSS)? (n.d.). In *GitHub Resources*. <https://resources.github.com/open-source/what-is-open-source-software/>

Xie, Y., Tian, J., & Zhu, X. X. (2020). Linking Points With Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geoscience and Remote Sensing Magazine*, 8(4), 38–59. <https://doi.org/10.1109/MGRS.2019.2937630>

Yuan, Z., Sun, C., & Wang, Y. (2018a). Design for Manufacture and Assembly-oriented parametric design of prefabricated buildings. *Automation in Construction*, 88, 13–22. <https://doi.org/10.1016/j.autcon.2017.12.021>

Yuan, Z., Sun, C., & Wang, Y. (2018b). Design for Manufacture and Assembly-oriented parametric design of prefabricated buildings. *Automation in Construction*, 88, 13–22. <https://doi.org/10.1016/j.autcon.2017.12.021>

Zhou, Q.-Y., Park, J., & Koltun, V. (2018). *Open3D: A Modern Library for 3D Data Processing*.