Hochschule Karlsruhe
University of
Applied Sciences

Fakultät für Architektur und Bauwesen
STUDIENGANG BAUINGENIEURWESEN

BACHELOR-THESIS

Development of a software module for the optimization of openBIM models for additive manufacturing

von

Fernanda Lourenzi

Matrikel-Nr. 157732

Referent:In:	Prof. Dr.-Ing. Reinhard Wimmer
Korreferent:In:	Dipl.-Ing. Andreas Geiger
Arbeitsplatz:	Institute for Automation and applied Informatics Group Semantic Data Model
Betreuer:In am Arbeitsplatz:	Dipl.-Ing. Karl-Heinz Häfele
Zeitraum:	06.05.2023 – 05.09.2023

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Bachelor-Thesis ohne unzulässige fremde Hilfe selbständig verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie alle wörtlich oder sinngemäß übernommenen Stellen in der Arbeit gekennzeichnet zu haben.

Ich erkläre hiermit, dass ich beim Einsatz von IT-/KI-gestützten Schreibwerkzeugen diese Werkzeuge des im Rahmen dieser Studienarbeit genutzten Funktionsumfangs vollständig aufgeführt habe. Davon ausgenommen sind diejenigen IT-/KI-gestützten Schreibwerkzeuge, die vom zuständigen Prüfer / von der zuständigen Prüferin zum Zeitpunkt der Abgabe meiner Studienleistung als nicht anzeigepflichtig eingestuft wurden („Whitelist").

Ich erkläre hiermit, dass ich alle Textpassagen, die durch den Einsatz von KI-gestützten Schreibwerkzeugen generiert wurden, unter Nennung des eingesetzten KI-gestützten Schreibwerkzeug gekennzeichnet habe.

Bei der Erstellung der vorgelegten Prüfungsleistung habe ich durchgehend eigenständig und beim Einsatz IT-/KI-gestützter Schreibwerkzeuge steuernd gearbeitet. Ich habe keine anderen als die zugelassenen Hilfsmittel verwendet.

Karlsruhe, den 31. August 2023

(Unterschrift)

Acknowledgements

The realization of this thesis has been an incredible experience that would not be possible without the support, collaboration and guidance of many people I am deeply grateful for:

Prof. Dr.-Ing. Reinhard Wimmer, my supervisor from the Hochschule Karlsruhe, who guided me from the beginning of this project while giving me a sense of security and providing precious insights, helping to shape this project. I am grateful for his thoughtful feedbacks and for working under his supervision.

Dipl.-Ing. Karl-Heinz Häfele and **Dipl.-Ing. Andreas Geiger,** my supervisors and mentors from IAI, whom I would like to express my deepest appreciation for giving me an opportunity to expand my knowledge and to work not just with this thesis but also in the past years as a student. Working with them made me gain experience and awaken an interest and excitement that helped me guide my future choices. I am grateful for the patience, opportunities and innumerous lessons given through this time. I extend my thanks to the staff of **IAI** for continuous assistance and resources who helped me in many parts of this thesis.

Shuchen Di, who I have found not just a perfect match for a work project, with countless discussions and intensive cooperation, but also a dear friend, helping the work to remain a constant source of joy and accomplishment.

My **friends and family**, who regardless of the distance, I was confident I could always rely for support in any manner they could.

Foreword

The structure of this work unfolds with the following chapters:

Chapter 1 – Introduction:

Presentation of the theme as well as problem statement, showing motivation and goals of the thesis.

Chapter 2 – Fundamentals:

An initiation into the foundational concepts for this thesis. The basic knowledge of BIM is presented, going from its dimensions to how data and information exchange is handled, and presenting the standards. Additive Manufacturing (AM) foundation is also explored and how it merges to the Architecture, Engineering and Construction (AEC) industry. It proceeds to elucidate the typical workflow of this dynamic relationship.

Chapter 3 – Conceptual Design:

The methodology and conception of the preparatory module takes center stage; as preliminary tests are conducted to detect potential errors in the conversion of data for individuals lacking 3D printing expertise. Exploring analogous modules to the proposed solution, the study navigates their pros and cons to find a well-defined approach for the interface, guided by objectives that look for an enhanced workflow and outcomes.

Chapter 4 – Implementation:

Moving forward, the implementation of the module is dissected, presenting foundational insights into BIM modeling before delving into an examination of the module itself, illustrated through a chosen use-case scenario while showing the challenges faced.

Chapter 5 – Final Tests and Evaluation:

Once the application is implemented, the module can be put into practice by presenting the final tests and showcasing a juxtaposition of the outcomes against the initial tests results. An evaluation of the findings and how it compares to the initial goals is made and future plans for the application is also discussed by exploring the improvements that can be made outside the time of the thesis.

Chapter 6 – Conclusion:

The thesis will be concluded with an overview of all the steps, evaluation and outlook.

Both “AM” and “3DP” will be used interchangeably in the text for smoother reading, as they refer to the same process.

Kurzfassung

Entwicklung eines Softwaremoduls zur Optimierung von open-BIM für Additive Manufacturing-Prozess

Im Bereich des Bauwesens formt die Integration neuer Technologien unsere Herangehensweise an Design und Konstruktion neu. Diese Arbeit erkundet das innovative Feld des Additive Manufacturing (AM) im Bereich der Architektur, Ingenieurwissenschaften und Konstruktion (engl. Architecture, Engineering and Construction) (AEC), wo es seine Vorteile auf der Baustelle in Konstruktionen im Maßstab 1:1 oder als visuelles Werkzeug in kleinskaligen Modellen entfaltet. Diese Verbindung kann zu Innovation, Nachhaltigkeit und Wirtschaftlichkeit führen. Allerdings bieten die üblichen Building Information Modeling (BIM) Softwarelösungen keinen reibungslosen Übergang vom digitalen Zwilling zum 3D-Drucker, insbesondere für kleine Skalenmodelle. Dies führt zu Modellen, die nicht mit der ursprünglichen Idee vergleichbar sind. Diese Arbeit, die in Kooperation mit dem Institut für Automation und Informatik (IAI) am Karlsruher Institut für Technologie (KIT) entstand, stellt einen neuartigen Ansatz im Arbeitsablauf für den 3D-Druck von kleinskaligen BIM-Modellen vor: Ein automatisiertes Modul zur Identifizierung und Behebung von Fehlern vor dem Druck. Das Modul wurde im KITModelViewer erstellt, einem BIM-Viewer, der am KIT entwickelt wird. Die Methodik beginnt mit der Erzeugung von 3D-Druckmodellen auf Basis von Dateien in openBIM Format IFC (Industry Foundation Class), die in gängigen BIM-Softwarelösungen erstellt wurden, um häufige Fehler zu identifizieren, und durch die Analyse vergleichbarer Module, den besten Ansatz für den Arbeitsablauf vom BIM- zum AM-Prozess zu bestimmen. Im Anschluss an die Methodik wird die Umsetzung detailliert beschrieben, wobei die aufgetretenen Herausforderungen hervorgehoben und die Ergebnisse inklusive Vergleiche mit und ohne Optimierung gezeigt werden.

Abstract

Development of a software module for the optimization of openBIM models for additive manufacturing

In the field of contemporary construction, the integration of new technologies is reshaping how we approach design and construction. This study explores the innovative domain of Additive Manufacturing (AM) within the field of Architecture, Engineering and Construction (AEC) universe, where it finds its benefits on site in real scale constructions or as a visual tool in small scale models. This union can lead to innovation, sustainability, and cost-effectiveness. However, the standard BIM Software does not offer an effective transition from the digital twin to the 3D printer, especially for the small scale models, resulting in a model that lacks the full representation of the original idea.

This thesis, which was made in partnership with the Institute for Automation and Informatics (IAI) from Karlsruhe Institute of Technology (KIT), introduces a new approach in the workflow for 3D printing of small-scale BIM models: an automated module for identifying and rectifying errors prior to printing.

The module was created within the KITModelViewer, a BIM viewer created by the KIT. The methodology starts by making 3DP models from IFC files created in commonly used BIM Software with the goal of identifying common errors, and by analyzing analogous modules to determine the best approach for the workflow of the BIM to AM process. Following the methodology, the implementation is explored highlighting the encountered challenges while showing its results and comparing it to the initial tests made without the application.

Table of Contents

1 Introduction

In the dynamic of modern construction, the fusion of technology and creativity is reshaping the way we conceptualize, design, and build structures. Enter the innovative domain of Additive Manufacturing (AM), where the marriage of digital precision and tangible creation has opened doors to unprecedented possibilities. This revolutionary approach not only streamlines construction processes but also fosters sustainable practices and reduces production costs.

However, this progression has not been without its challenges. As we delve into the realms of 3D printing (3DP) within the Architecture, Engineering and Construction (AEC) landscape, a tapestry of complexities and considerations emerges. Focusing on the visualization use case, printing small scale models results in one central concern in striking a balance between the intricacies of design and the limitations imposed by 3D printing technology. Issues such as thin sections, oozing, stringing, and loss of details illuminate the critical need for strategies to ensure design fidelity throughout the printing process. Moreover, the transition from digital design to physical reality poses unique hurdles, especially when dealing with more extensive and complex models that must adhere to the confined spaces of 3D printers. The revelation that certain post-printing issues cannot be previewed during the design phase underlines the significance of anticipatory measures.

This thesis introduces an innovative approach to the 3D printing workflow, unveiling a step of automation that identifies errors and facilitates rectifications before the printing process begins. This shift is realized through the implementation of a preparatory module. Envisioned as a plugin within the KITModelViewer (successor to FZKViewer), the newly introduced module envisions simplifying the process of preparing digital models for 3D printing. At its core, it promotes the alignment of design intent with tangible realizations, aiming for a “what you see is what you get” application for AM in the Building Information Modeling (BIM) environment.

2 Fundamentals

2.1 Building Information Modeling: Digitalization in the AEC Industry

BIM has emerged over the past couple of decades as a pivotal approach in the design and construction industry. Originally proposed by researchers in the 1980's, but only reaching maturity in recent years, BIM is a method that consists of the continuous use of digital information across the entire lifecycle of a built facility, including its design, construction and operation (Borrmann, 2018).

This technology implementation profoundly changed the way architects and engineers work, by creating a digital twin of a model derived from the planning and construction phase containing important databases that serves as a crucial foundation for the operational phase.

According to the VDI 2552, the digital model of BIM consists of geometric and non-geometric structured data, which can be composed of several partial models. Digitalization is understood here as the capture and processing of graphical and alphanumeric building data as well as adaptation of organization and processes, describing digital transformation as a collaborative method (Krämer, 2023). BIM facilitates cross-disciplinary collaboration and provides the necessary IT tools and platforms, empowering faster and more accurate planning, resulting in expedited construction processes for buildings. Employing BIM allows for enhanced efficiency and reduced errors, ultimately benefiting the overall construction industry.

2.1.1 Data and Information in BIM

The shift from the traditional workflow to the new model-based approach necessitated the establishment of novel procedures and requirements. To facilitate this transition without disrupting established workflows, an approach involving different levels of BIM implementation was created, focusing on both internal and cross-company procedures (Borrmann, 2018).

The simplest differentiation lies between "big BIM" and "little BIM". Little BIM refers to the use of specific BIM software to fulfill discipline-specific design tasks. The building model remains isolated within a single software package and is not shared with other stakeholders. Whereas big BIM consists of model-based communication among all stakeholders throughout the entire lifecycle of a facility (Krämer, 2023).

The extent of BIM usage also raises the question on how this data should be handled. When project participants utilize the same BIM authoring tool, model-based data exchange can occur based on the proprietary data format of that specific tool, representing a closed BIM approach. The problem with this approach is that when architects, structural engineers, and MEP engineers utilize different software products optimized for their respective planning tasks, interoperability issues can arise. The lack of compatibility between proprietary data formats may lead to data loss, conversion issues or information discrepancies, compromising the overall efficiency and accuracy of the project (Borrmann, 2018).

Having these issues in mind, openBIM comes as a collaborative process that is vendor-neutral. It facilitates interoperability to benefit projects and assets, and empowers stakeholders to develop new ways of working, while enhancing communication and industry standard exchange methodologies. OpenBIM enables an accessible digital twin, which provides the core foundation to a long-term data strategy for built assets. This provides better sustainability for projects and for more efficient management of the built environment (Petrie, 2023) .

2.1.2 BIM Standards

To improve data exchange between software products in the AEC industry, the international non-profit organization buildingSMART was created, succeeding in defining a vendor-independent data format for exchanging comprehensive digital building models. The resulting object-oriented data model named Industry Foundation Classes (IFC) provides very rich data structures covering almost all aspects of built facilities (Borrmann, 2018).

IFC, which was ISO certified in 2013, is an open standard usable across a wide range of hardware devices, software platforms, and interfaces for many different use cases. Its schema can describe how a facility is used, constructed and operated, while also defining physical components, manufactured products, target systems or analysis. It achieves that by codifying in a logical way (buildingSMART, 2023):

- The identity and semantics (Name, identifier, object type...),

- characteristics or attributes (Material, color, properties...),
- relationships (location, connections, ownership, …),
- objects (like columns or slabs),
- abstract concepts (performance, costing),
- processes (installation, operations) and
- people (designers, contractors, suppliers, etc.).

This schema can be encoded in various formats, such as XML, JSON, and STEP, and transmitted over web services, imported/exported in files, or managed in centralized or linked databases (buildingSMART, 2023).

For data exchange processes, IFC-compatible applications implement the required subset of the IFC data model. These subsets are referred to as Model View Definition (MVD) (Krämer, 2023). A set of MVD defines each Information Delivery Manual (IDM), which defines the workflow and information exchange specifications and requirements for needed use cases. The MVD identifies the portion of the IFC model which is needed for the information exchange (Noardo, Arroyo Ohori, Krijnen, & Stoter, 2021). BuildingSMART has specified MVDs for common use cases, such as coordination view, structural analysis view, basic facility management handover view, reference view and design transfer view (VDI 2552-11, 2020).

IFC Specifications databases are listed by the buildingSMART as well as their components. In the specifications it is possible to find terms, concepts and data specification that originate from use within disciplines, trades and professions. For this work and its implementations, the specifications from IFC4x2 found in the buildingSMART website (buildingSMART, 2023) were used. The following conventions were applied:

- The prefix "Ifc" followed by the English word specifies names for types, entities, rules and functions.
- The attribute names within an entity follow the name convention without prefix.

The primary IFC element hierarchy is based on the accessing structure:

Project → Sites → Buildings → Stories → Spaces → Elements

The project (IfcProject) is the top-level container made up of one or more sites. A site, is a container of Buildings, that contains one or more stories, that is made up of spaces, that are defined of elements (Eastman, 1999).

2.2 Additive Manufacturing and the Construction Industry

AM, also known as 3D Printing (3DP), is a process of creating three-dimensional objects directly from computer aided designs (CAD), by adding material layer by layer without the need of any tool, jig or fixture (Srivastava, 2019). It was a revolutionary technology that made significant strides to design creativity, digital fabrication and time-compression with varieties of product (Srivastava, 2019).
In the AEC Industry, there are many applications for AM due to its innumerous advantages. 3DP can directly link the 3D model to the final product in concrete, as it does not need formwork in the process (Rangel, 2023). It enables architectural creativity, by allowing greater geometry flexibility impossible to be built by traditional methods; it increases safety in construction sites by automating dangerous tasks and promotes sustainability, since it uses only the necessary amount of material, reducing wastes and transportation needs (Sayegh, 2020). In summary, AM improves the productivity on site, produces sustainable buildings and lower costs of production (Haupt, 2022).
An example for the advantage in welcoming this technology in the construction industry, would be the cloud and data center of Heidelberg IT Management, Europe’s largest 3D-printed building. The construction, with know-how provided by the PERI 3D Construction, used high-speed printers to shape the walls which started in March 2023 and finished in July, finishing the work in just around 140 hours (Römer, 2023).
Another example would be the first 3D-printed office for Dubai Future Foundation. Created by the Chinese company WInsun, the construction took 17 days, with labor cost 50-80% lower than traditional methods and 30-60% less waste generated (Sayegh, 2020).
These examples demonstrate some of the utilization of 3DP in large-scale construction projects. However, this method is also applicable to small-scale projects, where its purpose can be found in visualization and to facilitate the construction process for testing. By visualizing the model on a small scale, it is possible to analyze relationships between different building elements and identify potential design flaws. The level of difficulty and expenses associated with printing BIM models will vary according to the intended use case and the necessary level of details. In a first analysis for viability, like in the case of

conceptual design, costs and prerequisites remain minimal. However, from a marketing standpoint, a more comprehensive approach is needed, involving meticulous detailing and presentation. This entails producing models with well-defined elements, intricate details, and an assurance of flawless outcomes. Points like the quality of the chosen materials and the precision of the printer will undoubtedly assume a crucial role, albeit potentially resulting in slightly elevated costs (Domínguez, Romero, Espinosa, & Domínguez, 2013).

2.3 Workflow for 3DP of a BIM model

The workflow process for 3D printing remains relatively consistent across various applications, with some differences primarily related to machine capabilities and material requirements. The fundamental steps generally involve designing the 3D model, converting it into a printable format, slicing it into layers, and sending the data to the 3D printer (Douglas, 2023). The printer then builds the object layer by layer using the specified material, ultimately producing the desired output. While specific details may vary, the core workflow for 3D printing remains similar, ensuring its versatility and wide-ranging applications.

This process for the construction industry also varies on the application. For real scale, usually include division into different phases with specific methods depending on the structural and functional component (Xu, et al., 2022). In this thesis, the focus will be in the application of AM for small scaled models, that means for the use-cases focused on visualization or simulation. In this context, quality requirements are not always crucial, as long as the relevant elements can be accurately displayed. While an approximate georeferencing that aligns with the right location and orientation can enhance visualization, a tolerance of a few meters typically has minimal impact, varying based on the circumstances (Noardo, Arroyo Ohori, Krijnen, & Stoter, 2021).

For this purpose, the phases of the workflow will be separated into three phases, as shown in the diagram of Figure 1: Modeling, Slicing and printing.

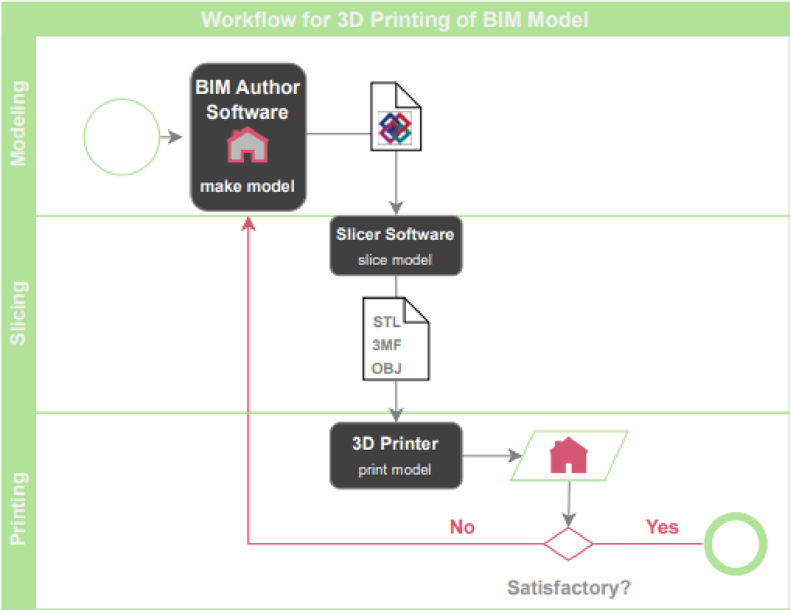


Figure 1: Workflow for 3D Printing of BIM Model

2.3.1 Modeling

The modeling phase constitutes the creation of the model within a suitable Author BIM Software (e.g., Revit, AutoCAD, etc.), designed to meet a specific use-case. It is important to emphasize that the majority of models are initially created for purposes other than 3D printing, such as design, construction, and documentation. Therefore, when preparing these models for 3D printing, certain adjustments and conversions are necessary to ensure their suitability for the printing process. Factors such as printability, scale, and structural integrity must be taken into account during this adaptation process, enabling a smooth transformation from the original design-oriented model to a version compatible with 3D printing.

The resulting file from the CAAD program is a model enriched with essential information concerning its geometry and intricate details. Subsequently, for seamless sharing and accessibility, open-source or neutral formats, such as IFC mentioned in chapter 2.1, are predominantly favored, as they promote widespread compatibility and ease of use.

2.3.2 Slicing

After the CAD-model is created, the file must be converted into a printable format and proceed to a slicing software. The software “slices” the model into 2D layers

and generates instructions in the form of a G-code (Douglas, 2023), which includes information about the printer's size, speed, method, material properties, and setup. The slicer software can differ and may be related to the printer, however, it can take some time to understand all the features, and not everyone who works with BIM models may be familiar with them.
Some of the formats suitable and used for 3D printing are:

- STL

ASCII STL format represents the 3D model as a collection of interconnected triangles, defined by the coordinates of its vertices. However, it is not suitable for high-precision or multicolor 3D printing, as it does not specify information about color, material, scale, or other contextual information. Additionally, STL files can be prone to errors due to non-manifold geometry (Dickson, 2023).

- STL binary

The most used format for 3D Printings, these files are a binary representation of 3D models. Instead of using plain text, binary STL files store the geometric information in a compact format. They are more efficient in terms of file size and can be read and processed faster by software (Dickson, 2023).

- OBJ

OBJ is the second most widely used file format for computer graphics. It uses polygons instead of triangles, creating a smoother geometry texture than STL. Unlike STL, OBJ also encodes color, material, and texture information via material list (MTL) and image (PNG) files, making it a preferred choice for realistic 3D visualizations.

- Additive Manufacturing File (AMF)

Official ISO standard XML-based format designed to be faster, less error-prone, and capable of storing color, material, and texture information. It uses curved triangles that must conform to strict rules that almost completely eliminates issues found in STL files. However, it is not universally supported by all players in the 3DP industry and has poor levels of uptake (Douglas, 2023).

- 3D Manufacturing Format (3MF)

It was developed with the intent of overcoming some of the deficiencies with AMF, sharing a very similar XML-based standard. However, 3MF has more exacting standards for numerical precision and reduction of rounding errors; and has, in a short period, achieved strong industry backing. This allows for more complete and accurate 3D models and simplifies the printing process, by storing information about printer and slicer settings, including supports (Douglas, 2023).

Table 1: Printable file formats for CAD software export and slicer software import

	CAD - Export					Slicer - Import	
Format	ArchiCAD	Revit	Allplan	FreeCAD	Blender	Prusa	Cura
STL	x	x	x	x	x	x	x
STL binary	x	x	x	x	x	x	x
OBJ	x	x		x	x	x	x
AMF				x		x	x
3MF				x	x	x	x

2.3.3 Printing

The resulting G-code from the slicer software is then transferred to the selected printer, and the printing process is prepared. In this process includes the choice of an adequate material, printing plate and post work on the model. The post work consists on cleaning, taking the supports outs and, when necessary, joining the printed parts together (Domínguez, Romero, Espinosa, & Domínguez, 2013).
Among the various types of 3D printers available, this work will primarily focus on Fused Deposition Modeling (FDM). FDM involves the extrusion of a thermoplastic filament through a heated nozzle. The nozzle moves with precision, following a specific pattern dictated by the G-code, and constructs the 3D object layer by layer. As each layer is deposited and subsequently cooled, it seamlessly fuses with the layer below, creating a solid and continuous structure. FDM 3D printers are widely preferred due to their accessibility, user-friendliness, and versatility (Chen, He, Yang, Niu, & Ren, 2017). Furthermore, FDM printers are compatible with various thermoplastic materials such as PLA, ABS, PETG, and more.
While printing may seem straightforward, achieving a satisfactory outcome requires careful consideration of several factors. Proper bed leveling, adjusting print temperature, generating support structures, and optimizing print speed contribute to stable printing and better outcomes as well as using high-quality filament, maintaining a dry storage environment, and ensuring effective bed adhesion prevents common print defects (Singh, 2018).

3 Conceptual Design

The conceptual design for the implementation was based on the bachelor thesis of Shuchen Di “Konzeptionierung eines automatisierten Arbeitsablaufs im skalierten FDM-3D-Druck mit dem Ziel der Generierung von drückfähigen Modellen aus BIM-Daten” for Karlsruhe Institute of Technology (Di, 2023). The two theses complement each other and overlap at certain points.

Envisioning an enhanced workflow for 3D printing of BIM data, the work outlined in this thesis aims to simplify the process of printing the model, even for individuals without extensive expertise in AM. It introduces and implements a plugin for the KITModelViewer, facilitating automatic adjustments and filtering. The new proposed workflow is shown in the diagram depicted in Figure 2.

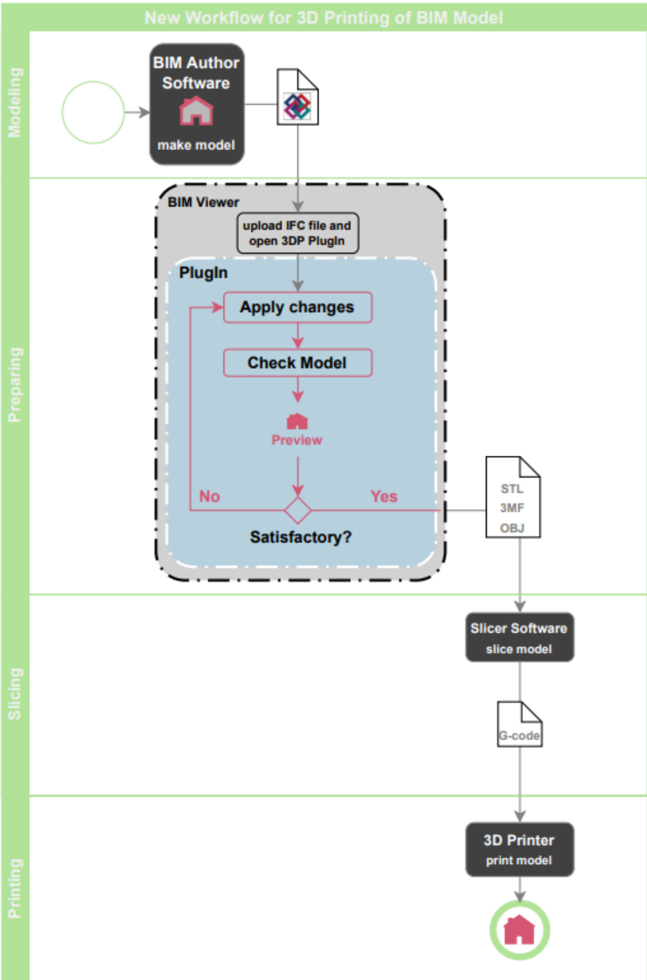


Figure 2: New Workflow for 3D Printing of BIM model

By integrating the plugin into the KITModelViewer, users can experience an intuitive "what you see is what you get" interface. The viewer is adapted to identify problematic aspects of the model relevant to the chosen printer. Additionally, the viewer can attempt to rectify these issues, ensuring a more favorable outcome.
This innovative approach enables users to preview and assess the viability of printing, preventing unnecessary material and time waste. The benefits encompass not only the actual printing time but also the common need for model adjustments.
Before proceeding with the implementation phase, a series of preliminary tests were conducted. These tests aimed to identify potential errors that might occur when individuals lacking 3D printing expertise engage with the process. Additionally, an evaluation of analogous software tools was undertaken to gather insights so that well-defined objectives were established for the development of the proposed interface.

3.1 Initial Tests

The testing was conducted by attempting to print the model from Figure 3 generated in ArchiCAD 22.

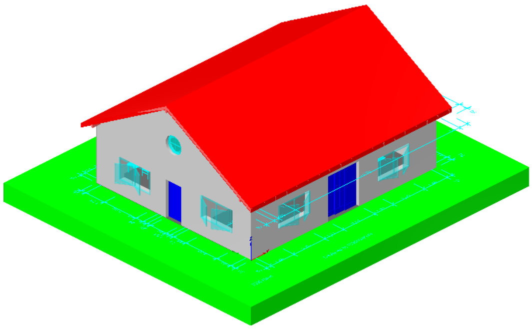


Figure 3: FZKHaus used for printing testing

This will show the different outcomes the printing of a BIM model is able to produce and what kind of errors were found. The tests were conducted through the following cases:

- Test 1: Printing converting the IFC file STL as it is without modifications.
- Test 2: strategic adjustments were implemented to enhance the 3D printing outcome. Void elements such as IfcSpace and IfcOpenings were intentionally excluded from the process. Furniture components were integrated into the printing, and each storey, along with the roof, was printed individually. To ensure a cohesive final structure, Boolean functions were employed to align the two storeys, and supplementary geometry was introduced for the same reason. The additional geometry was created editing the IFC file using the Trimple Navigation 3D Design Software SketchUp, with the task completed in approximately 30 minutes.
- Test 3: The model was printed in its entirety, excluding features like openings, windows, doors, and furniture. This print incorporated the fundamental building elements, resulting in a solid structure with an interior void, albeit with internal support structures.
- Test 4: For the fourth iteration, the same model was printed, but this time without the use of any support structures.

For the initial tests Prusa i3 MK3S printer was used and the model was sliced in the PrusaSlicer Software and obtained the following results:

- **Test 1**: The printing process took approximately 6 hours, with an additional 30 minutes for the removal of support structures. As depicted in Figure 4, all elements from the IFC model were printed, including void volumes such as openings and spaces. Regrettably, these void volumes were printed as solid entities, resulting in unnecessary material consumption and extended production time. Moreover, the doors were not discernible in the printed model; as they were indistinguishable from the walls.

Figure 4: Model with details missing and interior filled

- **Test 2**: The entire printing process extended over 4 hours and 40 minutes, with an additional 1.5 hours dedicated to the meticulous removal of support structures; a task that demanded attention due to its complexity, since the furniture was so thin it had the risk of breaking during removal process.
  In this test, the printing time was reduced, more details were visible and less material used. However, despite these enhancements, challenges persisted. The printing process revealed issues of oozing and stringing as depicted in Figure 5 and 6, primarily stemming from the presence of numerous thin and delicate elements within the design. Upon the removal

of support structures, it became evident that a significant portion of design details and furniture had been compromised or, in some cases, completely omitted.

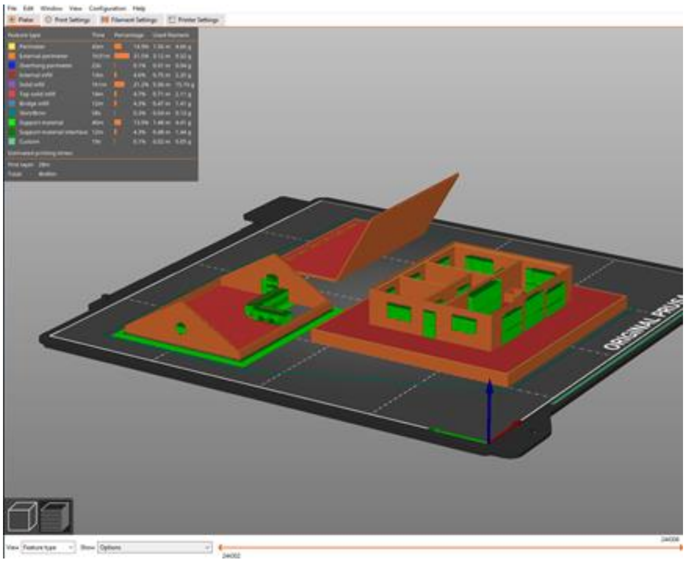


Figure 5: Model separated by Storey in Slicer Software

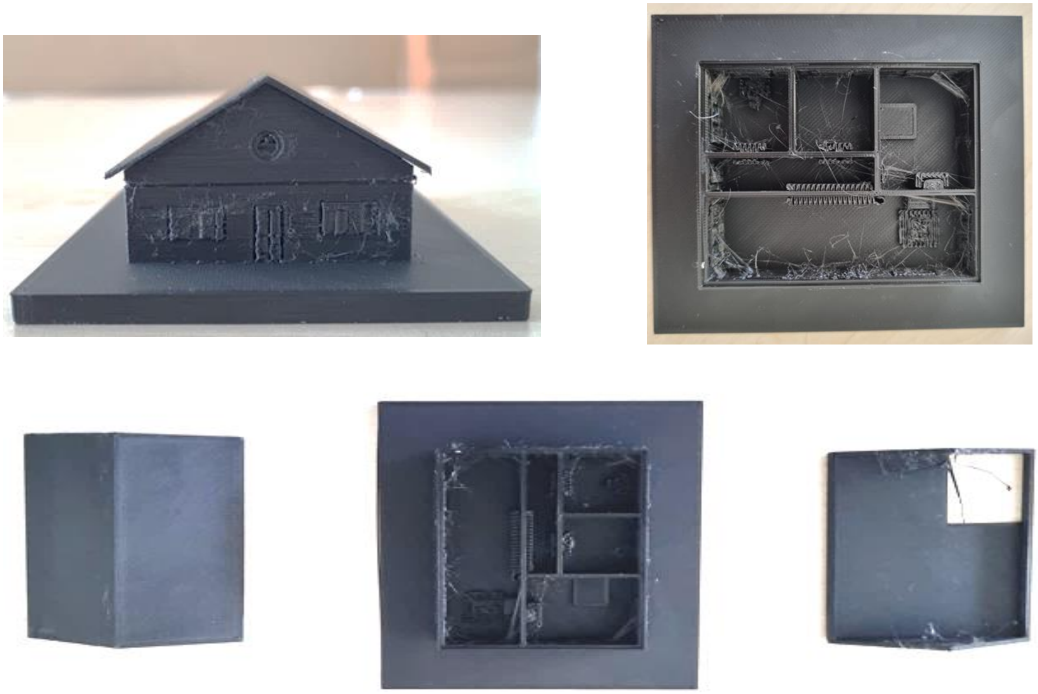


Figure 6: Oozing, stringing and missing parts in model

- **Test 3**: The printing process itself required 5 hours and 50 minutes, and the subsequent removal of support structures demanded approximately 3 hours, notably due to the difficulty to remove support in the interior of the model. This task proved to be quite demanding, emphasizing the importance of carefully planning and executing post-printing procedures. Furthermore, the internal view of the structure was limited, mainly visible through the windows and doors.

Figure 7: Model printed with support and after removal; interior of model view

- **Test 4**: The printing process was notably faster, and the overall material consumption was reduced. However, it is important to acknowledge that this method was not without its challenges. Certain points of the model, particularly the lateral sections, exhibited some failures and imperfections due to printing in an unsupported area.

Figure 8: Model printed without support

Some other models were printed to explore the response to different geometries. While the FZK house served as a suitable initial example for testing, it is important to note that its simplicity does not encompass the level of complexity often encountered in the industry's models. For instance, when dealing with larger structures, the limitations of common 3D printers become apparent. Scaling down a large building to fit within the printer's confined space often leads to walls, slabs (such as floors or roofs), and other elements becoming excessively thin for the printer nozzle. While some parts of these scaled-down models might still be printable, issues begin to show the challenges in this approach.

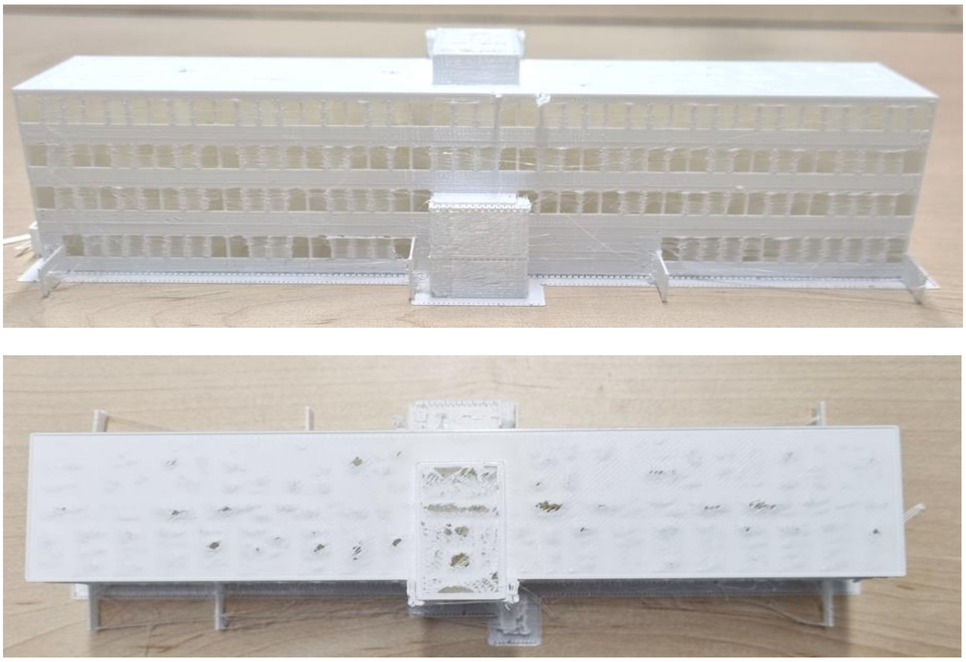


Figure 9: Model presenting layers too thin for printer

An alternative printing approach involved using a different printer for the FZK house model. This printer allowed the utilization of support that could be dissolved in water after printing. This method proved advantageous for several reasons. The removable support simplified the support removal process, eliminating the need for meticulous manual removal and mitigating the risk of damaging other parts. Additionally, it addressed the challenges encountered in supportless models, presenting a viable solution.

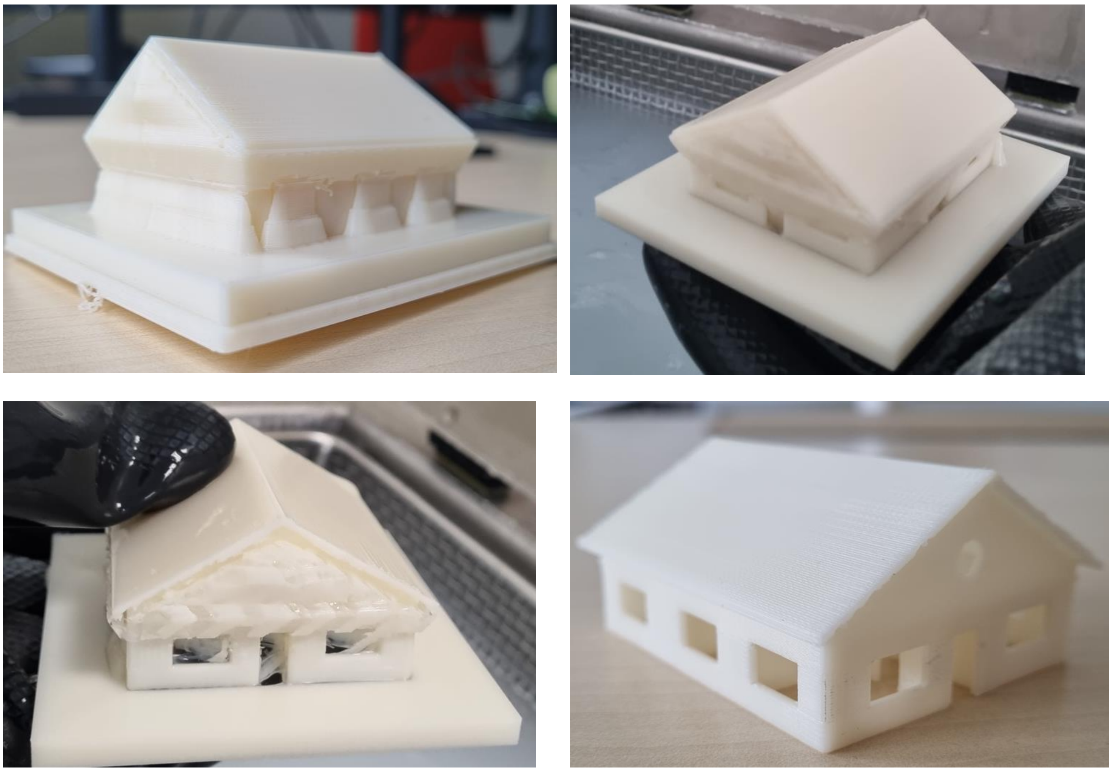


Figure 10: Model with soluble support. Recent printed; after 3 hours dissolving support; after 5 hours dissolving support; and completely dissolved

However, it is important to highlight certain considerations associated with this method. The alternative printer comes at a higher cost than a standard one, and the soluble support material itself is more expensive. Furthermore, this method introduces additional phases to the printing process. Removing the support requires soap water inside of an ultrasonic tank, which provides continuous movement of the water solution as it slowly dissolves the support material. The additional step can extend the overall process by several hours.
While this approach presents satisfactory results, its time-consuming nature and associated costs should be taken into account. As such, it is best suited for particularly complex or special models rather than routine and simple applications.

In summary, several key challenges that emerge in the 3D printing process could be observed. One prominent issue lies in balancing design details with the constraints of 3D printing technology. Thin sections, oozing, and stringing become evident, highlighting the need for better strategies to maintain design integrity. Additionally, the complexity of larger models presents difficulties when attempting to fit within limited printer dimensions, often leading to compromised elements. Many of these post-printing issues cannot be predicted during the design phase, leading to wasted time and resources through multiple iterations. Therefore, the ability to identify errors before printing or swiftly rectify them would greatly improve the overall outcomes, streamlining the 3D printing process and generating superior results.

Table 2: 3D Printing Tests Summary

Initial Tests Summary							
	IFC Model	Modeling Software	Printer	Considerations	Printing Time [h]	Support removal time [h]	Total time [h]
Test 1	FZK-Haus	ArchiCAD 22	**Prusa i3 MK3S**	No changes made to the model	6	0,5	6,5
Test 2	FZK-Haus	ArchiCAD 22	**Prusa i3 MK3S**	IfcSpace and IfcOpening excluded; Storeys separated	4,7	1,5	6,2
Test 3	FZK-Haus	ArchiCAD 22	**Prusa i3 MK3S**	just fundamental building elements printed; support inside	5,8	3	8,8
Test 4	FZK-Haus	ArchiCAD 22	**Prusa i3 MK3S**	As Test 3, but no support	3	0	3

3.2 State of Technology

There are some software applications that currently provide assistance throughout the conversion of IFC files to printable formats, often accompanied by additional features for modification and refinement.

In the Bachelor thesis of Shuchen Di, she evaluates software solutions for 3DP data preparation in modeling and slicing software as well as the ones dedicated for it. The results can be seen as follow (Di, 2023):

3.2.1 Solutions for Modeling Software

For modeling software, two options of commonly used BIM Software were chosen that presents ownership: ArchiCAD and Revit; and one open source: Blender.

- ArchiCAD and Revit: ArchiCAD is a BIM CAD Software developed by the GRAPHISOFT, a company that offers computer aided solutions for handling all common aspects of aesthetics and engineering during the whole design process of the built environment (Graphisoft, 2023). Revit is also a BIM Software provided by Autodesk that allows architects, engineers and construction professionals to model shapes and structures while streamlining the project management with revision of plans, schedules, etc. (Autodesk, 2023) BIM experts familiar with these software environments can efficiently locate required functions. However, there are limited settings for AM in the 3D format export add-on, mainly basic options like STL or OBJ format, color, transparency, and units. Notably, the lack of convenient scale settings evident in the Revit export interface, complicates producing scaled visual models. The lack of properties in this preparation requires labor-intensive manual assessment of modifications.
- Blender: developed by the non-profit Blender Foundation, it is a widely used open-source software tool for 3D computer graphics used for creating diverse art visuals, video games, animation and 3D printed models (blender.org, 2023).

  A dedicated 3DP add-on is available, offering scaling and export capabilities to file formats such as STL, OBJ, and 3MF. This toolbox streamlines modification and adjustment tasks boasting an array of features like volume and area statistics, geometric assessments, and manifold generation.

However, when comparing Blender to specialized BIM software, it lacks certain architectural features, particularly the ability to identify construction-specific element groups. This feature, crucial for efficiently preparing scaled structural building models, is absent. Exporting IFC data to STL can result in errors, requiring additional corrections. Notably, managing IFC files within Blender poses challenges, especially for those dealing with larger files or who are new to the software. While using its manifold issues correction, for example, can result in loss of valuable pieces in the final result.

An important consideration is that BIM authors typically lack familiarity with Blender, requiring additional learning to utilize its modeling. Although Blender offers remarkable artistic potential, its complexity makes proficiency a challenge for the average user.

3.2.2 Solutions for Preparing Software

A range of tools are available for correcting 3D print files, including Materialise Magics 3D and Siemens PLM NX. These tools excel in producing data tailored for AM data preparation.

- Materialise Magics 3D: The software operates at the STL level, offering diverse functions like design alteration, simplification, 3D texturing, and more. It acts as a bridge between CAD files and 3D printers, with features including editing, support generation, measurement, and slicing (Materialise, 2023). It streamlines 3D printing by enabling easy reconfiguration and hardware-software integration. Users can refine designs, prepare models, and establish efficient printing workflows
- Siemens PLM NX: An encompassing CAD/CAM/CAE software solution that offers sophisticated free-form modeling, shape analysis, rendering, visualization, and multidisciplinary simulation features. It empowers the assessment of support and model suitability prior to 3D printing (3DP), incorporating performance predictions through CAE analysis (Siemens, 2023). Furthermore, Siemens PLM NX permits direct output to 3D printers.

However, its primary focus is on the machine manufacturing industry. Several settings within this software might not be conducive for BIM authors and could prove complicated to grasp.

In addition to their potential benefits, these software solutions often require purchasing extra licenses, incurring additional costs. Moreover, using these solutions demands expertise, making them less accessible to everyone. This complexity can hinder widespread adoption, especially for basic use cases where the investment of time and money might not be justified by the outcomes.

3.2.3 Solutions for Slicing Software

As mentioned in the workflow description for 3DP, the slicer software is utilized to convert 3D models into machine-readable instructions for the 3D printer. They are capable of making some model corrections. A couple of examples include:

- Prusa Slicer: The most used slicer for our testing phase, it provides a user friendly interface and offers a layer of printing settings. It also offers support generation and some manifold corrections (Prusa Research by Josef Prusa, 2023).
- Cura: Benefiting from a broad user base and community, Cura has regular updates and an extensive array of plugins and extensions. Noteworthy advanced features include experimental print models and customizable print profiles. Its compatibility extends to various 3D printer models, making it a versatile choice for diverse users (UltiMaker, 2023).

Slicing software has advanced a lot over time, but they concentrate mainly on the slicing process involving software and hardware. In contrast, other methods take a more comprehensive approach to 3D printing data preparation. Slicing software is specialized for accurate slicing and its role comes into play after the data preparation, focusing on the printer output stages.

3.3 Goals

Based on the initial tests and analysis of the workflow within the aforementioned applications, the primary goal of a BIM model preparation module is to optimize the process of converting designs into print-ready 3D models. This module should seamlessly bridge the gap between complex BIM software and the requirements of 3D printing, offering tools and a workflow that focus on simplifying complex designs, ensuring accuracy, and enabling customization for the specific needs of 3D printing. Additionally, the module should facilitate easy integration with various BIM Viewer applications and offer user-friendly features, allowing BIM authors to efficiently transition from detailed designs to high-quality, tangible 3D prints. In other words, to enhance the efficiency, accuracy, and accessibility of preparing BIM models for successful AM processes.

This would be achieved by facilitating the selection and deselection of entire entity types, by printing without IfcSpace or IfcFurnishingElement for example; specific target entities; or the ones based on common properties, such as choosing to printing a model only with the Property isExternal true. The UI should allow for a rapid confirmation of whether the desired elements are present within the viewer, and subsequently, if the model aligns with the printer's capabilities for successful printing.

Three vital parameters emerge for thorough checking of the model's suitability for printing:

- Solidity Check: A fundamental criterion involves confirming whether the model forms a solid structure. Models lacking volume would be incompatible with the printer and could lead to complications during subsequent stages.
- Thickness: The second consideration involves assessing whether the model's edges adhere to the printer's nozzle size parameters. Elements that are excessively thin may result in issues such as oozing or incomplete parts, like the furniture example previously discussed in the testing phase.
- Scaled Dimensions: Another key aspect involves verifying that the scaled model's dimensions align with the printer's capacity. This encompasses both width and height to ensure a proper fit.

While the concept of support checks may be earmarked for future implementation, it holds less urgency given that most slicer tools already incorporate this functionality. Support requirements might be addressed by eliminating hanging elements, although specific parts could potentially be printed as bridges without requiring support. Achieving this bridge functionality would necessitate designing small bridge spaces within the model and adjusting the printer settings accordingly.

Post-checking, the UI should offer swift remedies for identified errors. Initially, the easiest solution might involve excluding problematic elements. However, recognizing that these elements could hold significance for conveying the proposed concept, alternative corrective measures should be available. The ability to adjust element thickness could prove very valuable, particularly for large building models where scaling down becomes essential for optimal outcomes.

4 Implementation

4.1 Modeling base concepts

BIM relies heavily on the digital representation of three-dimensional building designs. To understand modeling tools and exchange formats, knowledge of computer-aided geometric modeling principles is essential.

There are two fundamental approaches to three-dimensional body modeling: Explicit modeling, which describes volumes in terms of their surfaces using Boundary Representation (BRep); and Implicit modeling, which employs a procedural approach through construction steps. Both methods are part of the IFC specification (Borrmann, 2018).

Explicit modeling involves defining a hierarchy of boundary elements (solid, face, edge, and vertex) with topological information represented by their relationship that can be observed in Figure 11. Geometric dimensions are added to describe the body with nodes containing coordinates of the vertices.

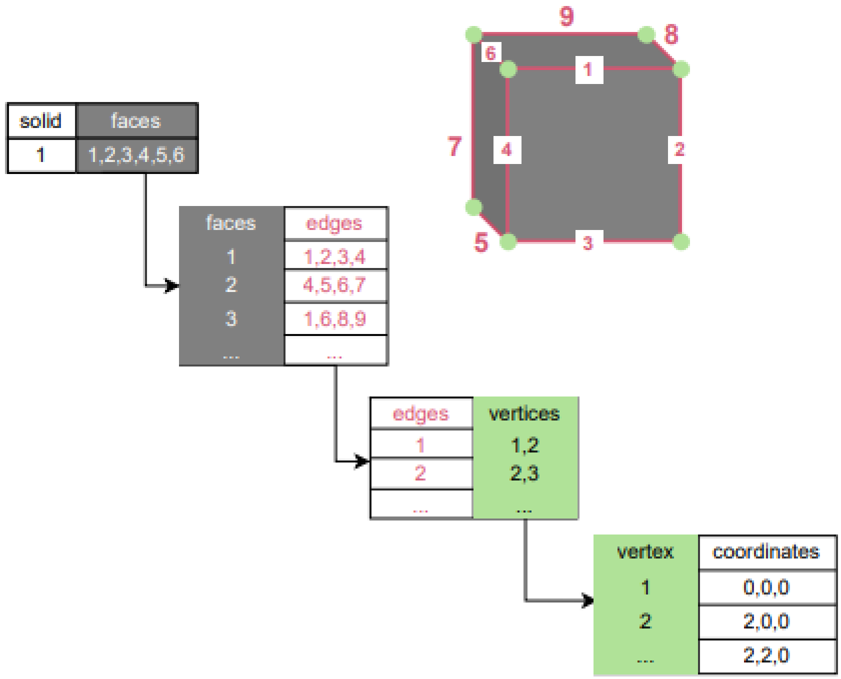


Figure 11: Object geometries relationship

Implicit methods store the history of the creation of a 3D body. They utilize Constructive Solid Geometry (CSG) to combine predefined basic objects using Boolean operators to create more complex objects. The objects can unify, intersect and differentiate to create more complex ones. Subtraction relationships are a key part of the IFC model. They're used to create openings by taking away certain parts of geometry, like making a hole in an IfcWall or IfcSlab. This process ensures consistency in how openings are made according to the data model. These openings, defined using IfcOpening, can then be filled with elements like IfcWindows or IfcDoors, completing the design (Noardo, Arroyo Ohori, Krijnen, & Stoter, 2021).

Many objects are also created by extrusion and rotation, where a 2D geometry is moved along a path to form a 3D solid. These methods are often used in CAD and are included in the IFC data format (Borrmann, 2018).

These objects would be the entities in an IFC project, where the entity type refers it to its class. An entity could be a wall, column or represent a non-physical thing, like load or task, and each is defined by its identity (id) (Borrmann, 2018). The class incorporates entities that hold the same structure or characteristic and it is what in the implemented UI is called “element types”. Moreover, each entity has attributes, that being its properties or data and information associated with it. As the example in Figure 12, entities could have a relation, where a wall entity (IfcWall type) and an opening (IfcOpening) are related to each other. Each having its own attributes.

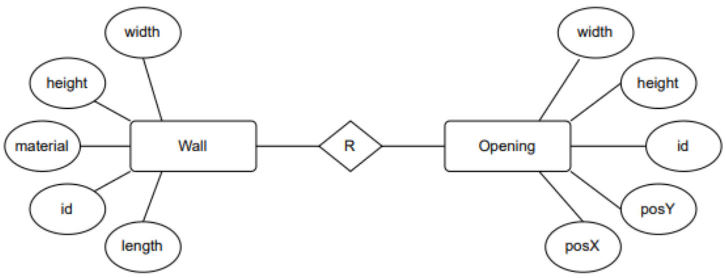


Figure 12: Entities relationship with attributes adapted from Borrmann, 2018

4.2 Software and Libraries

Visioning the goals discussed in the last chapter, the implementations were carried out using Microsoft Visual Studio 2022 in the C++ programming language. The interface dialogues were constructed with the assistance of wxFormBuilder, a tool employed in conjunction with the wxWidgets library.

wxWidgets is an open source C++ library that lets developers create applications for Windows, macOS, Linux and other platforms with a single code base. It has popular language bindings for Python, Ruby, Lua, Perl and several other languages and it uses the platform's native API rather than emulating the GUI (wxWidgets Cross-Platform GUI Library, 2023).

The KITModelViewer, a semantic data model viewer successor to the FZKViewer created at IAI, offers a revised visualization with a PlugIn Interface, housing the planed module within its framework. The application is proficient in exhibiting object properties and relationships, thereby facilitating data (IAI- KIT, 2023). The implementation also took benefits of the libraries from the Viewer by reutilizing methods, such as edges or solid checking, or the export to STL files that was already available in the Software.

Since most of our tests were carried out in Prusa printers, the PrusaSlicer Software was constantly used for preview tests and later for slicing.

4.3 User Interface

The UI flow comprises a series of interconnected steps that guide the user from start to finish. Beginning with loading the IFC file into the viewer, the user then chooses what elements are important for the target case, followed by the filling of the printer's parameters. The printability of the model is then checked and visible to the viewer. Final changes can be made to the model if chosen and finally exported to the preferable printable format.

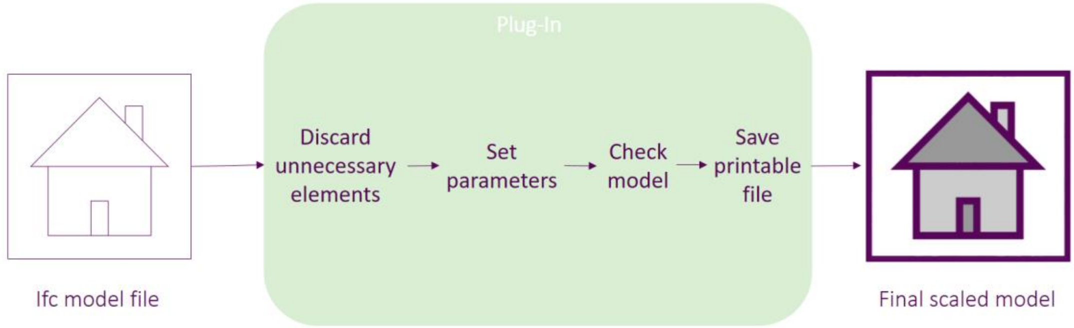


Figure 13: Module simplified workflow

The features will be explained by following the given workflow and using a "building construction" example. This case shows how only the basic building structure is printed, excluding additional elements like windows or furniture. This practical demonstration helps illustrate how the software works in a straightforward manner.

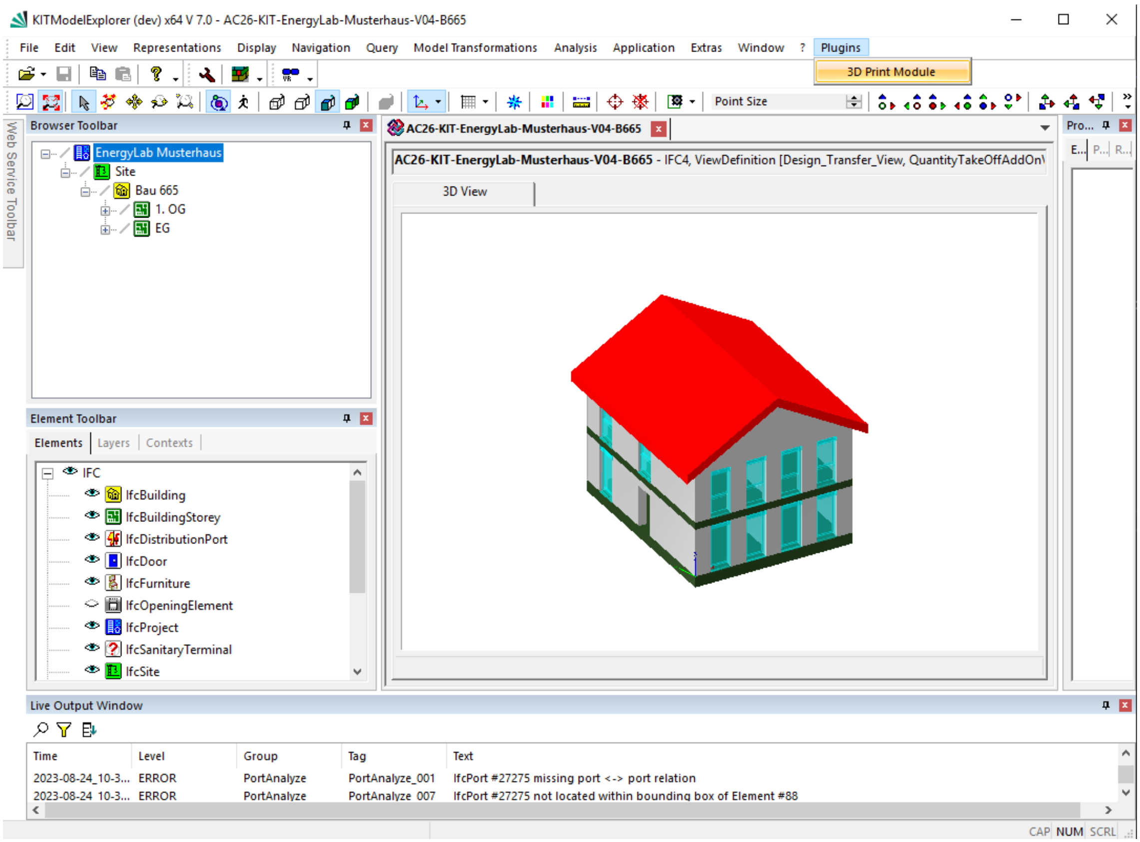


Figure 14: KITModelViewer and plugin on top right

4.3.1 IFC Elements

Upon initiating the plugin, all entities within the viewer are conveyed to the plugin in the form of a list. This list is characterized by the type of element's name and a data structure outlining its properties. This structure encompasses a Boolean value indicating its activation status, i.e., whether the element's type is selected or not. It also incorporates a list comprising elements—entities that pertain to that specific element type. Each element in the structure is identified by its Oid (an application-specific identifier) and possesses an own corresponding data structure. This element structure includes a Boolean value (true or false) representing its selection status, along with the entity's name and its associated ID (attributed to the entity). These elements undergo updates in JSON files[1] as the application is utilized.

[1] Java Script Object Notation: open standard file and data interchange format

At the outset of the process, various model types are presented for selection in the main Dialog, positioned on the top left (Figure 15 No.1). The model type in this case does not carry the meaning defined in the VDI-2552/4 specifically, which divides the types into specific purposes in the BIM process, rather each type is pre-filtered to suit distinct visualizations catering to diverse use cases attributed to a discipline model, such as architecture building shell, facade, structural, etc. The module starts with two options "Default" and "All" types. "All" type will always select all the elements and cannot be edited. "Default" can be changed to fit the users preferability. They have the flexibility to create and customize these types as seen in Figure 16 (on the left) according to their requirements. Once these types are established, a JSON file bearing the chosen name and comprising the selected elements is stored within a configuration file. As the plugin is started, the elements and its types are saved in a list with the name of the model type and all the its elements' names.

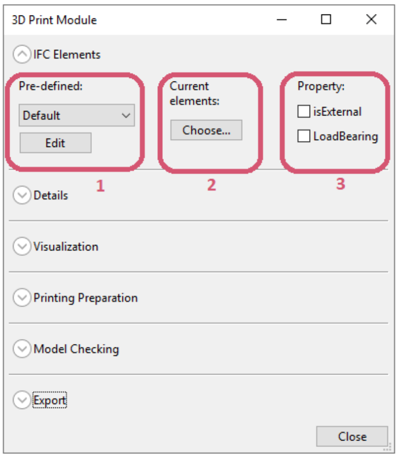


Figure 15: IFC Elements.

Once the foundational elements are chosen, users can then go into finer details (Figure 15 No.2). They have the option in Figure 16 (on the right) to individually filter each entity, either excluding or including them. The "Choose elements" functionality triggers a dialog with a tree control displaying all elements along with their child entities. Upon checking an entity, the Boolean value for the respective element is set to true, signifying its selection status.

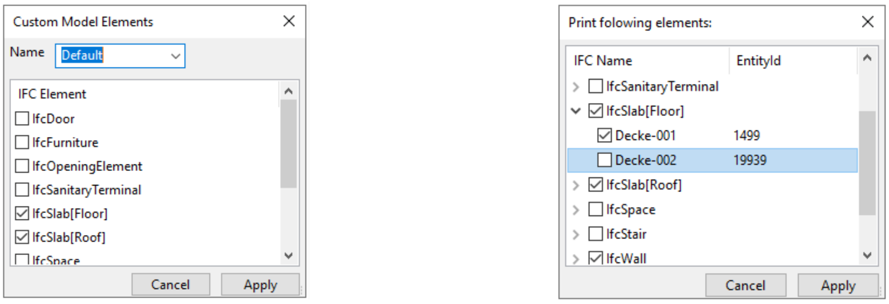


Figure 16: Type of elements (left) dialog and entity choice dialog (right)

In addition to the "Choose elements" dialog, an additional option is the standard property filter designed for commonly tracked properties. This filter offers a selection of checkboxes in Figure 15 No.3, specifically "IsExternal" to display only entities with "external" properties (Figure 17 on left), and "LoadBearing" to reveal entities categorized as loaded structures (Figure17 on right). By activating these filters, the selected elements and entities undergo filtering, resulting in the viewer exclusively displaying elements possessing the specified properties. Naturally, for the function to be effective, the entities must be documented properly. Even if an entity is external but was not documented as so, the application will be unable of filtering it.

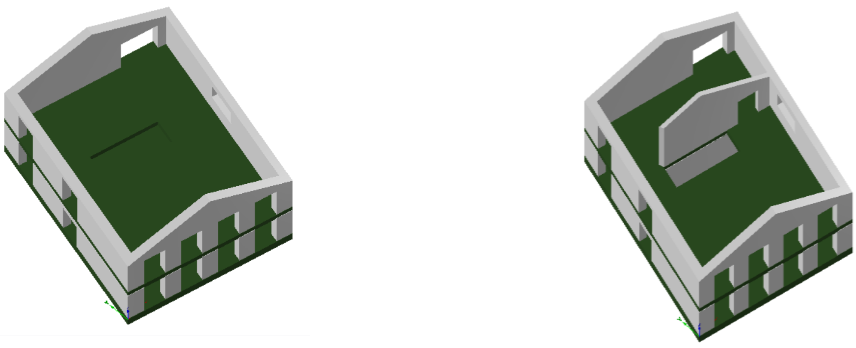


Figure 17: IsExternal property (left) and LoadBearing property (right)

4.3.2 Details

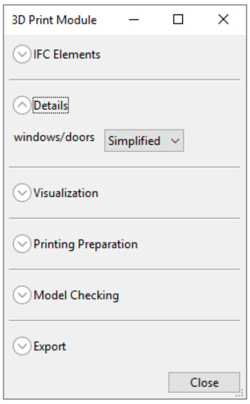


Figure 18: Details

Beneath the IFC elements section, a dedicated "Details" segment is situated. Currently, this section provides a choice solely for window and door details. When opting for "simplified" or "detailed," the visual representation of these elements undergoes adjustment to align with the selected preference. In the "simplified" mode, elements are presented in a streamlined manner, omitting finer details like frames or door knobs (figure 19 on the right). Conversely, the "detailed" mode showcases these distinctive characteristics (Figure 19 on the left).
It is essential to highlight, however, that a majority of models will not retain such intricate details once they are printed. Due to the scale factor involved, most of these finer nuances tend to be lost during the printing process.

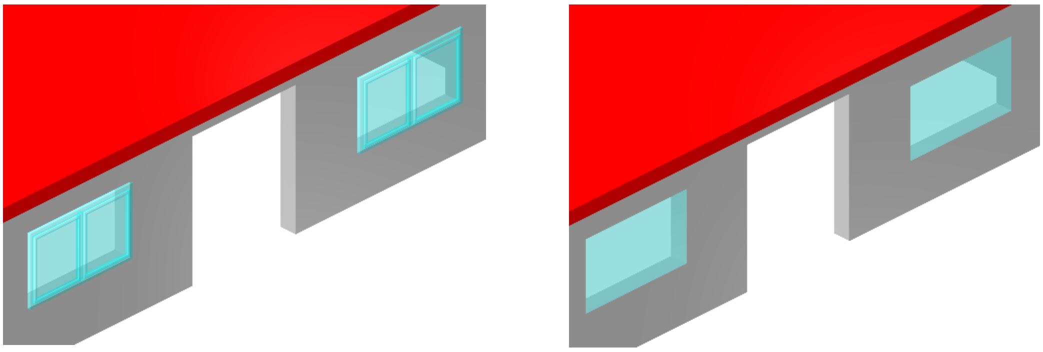


Figure 19: Details - "detailed" (left) and "simplified" (right)

4.3.3 Visualization

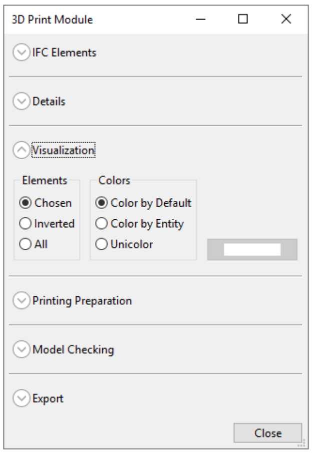


Figure 20: Visualization

The "Visualization" segment provides users with an enhanced overview of the selected and filtered elements within the model. Users can switch between views of chosen entities, those not selected, or all elements collectively. It is important to note that this section is purely a visualization tool and does not directly impact the printing outcome.

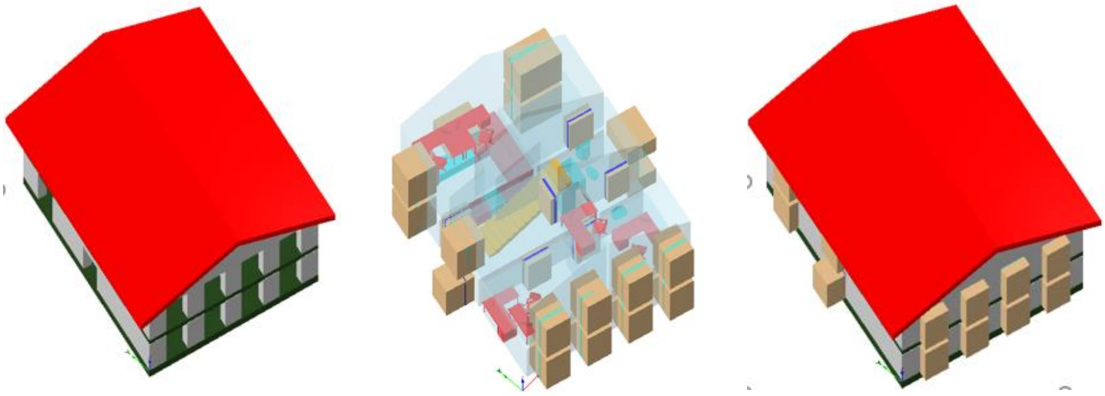


Figure 21: Visualization: Chosen (left), Inverted (middle) or All elements selected (right)

Moreover, within this segment, color views can be modified. These color views can be based on the attributes of the element type, the entity itself, or a user-selected monochromatic color. Opting for a monochromatic view aligns well with the preview of the final model appearance, considering that printers typically produce models in a single color. This feature becomes particularly valuable in cases where there is minimal variation in relief, allowing users to see if it would be possible to clearly distinguish the desired elements.

4.3.4 Printing Preparation

Once the model incorporates all the desired elements and details for printing, the next step involves configuring the printing settings (Figure 22 on the left). This phase holds significance as it precedes the assessment of the model's printability check. These printer options are also stored in JSON files and are open to customization as needed in Figure 22 (on the right). The parameters encompass the following information:

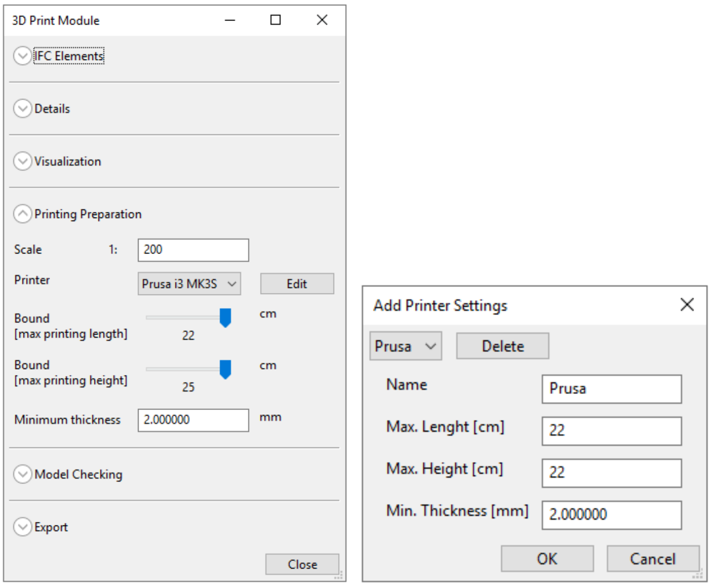


Figure 22: Prepare printing (left) and printer choice settings (right)

- Scale: Users have can select the scale at which the model will be printed. Later, the model will be adjusted to fit this chosen scale so the elements can be measured using the viewer tool. The chosen scale serves as the foundation for evaluating the subsequent parameters.
- Maximum length: Printers possess limitations on the area they can cover during printing. These area boundaries are entered into this field, allowing for later checks to determine if the scaled-down model fits within these boundaries.
- Maximum height: This parameter specifies the maximum vertical dimension that the printer can effectively print for a given model.
- Minimum thickness: Each printer comes with a nozzle size, which determines the minimum thickness for successful printing. Any element with a thickness smaller than this nozzle size is likely to encounter issues.

The minimum thickness can match the nozzle size, adhere to the printer company's recommended minimum, or align with the user's preferred thickness, providing an extra margin for safety.

Configuring these printer parameters is essential to ensure that the model is appropriately prepared for printing and that potential issues related to scaling, dimensions, and structural thickness are considered and addressed in the following step.

4.3.5 Model Checking

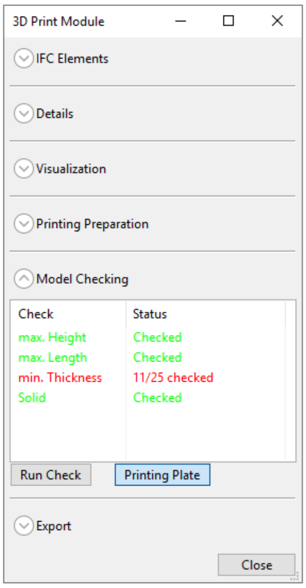


Figure 23: Model Checking

The main function of the UI is to provide users with a realistic preview of their model's appearance. Making geometric checks to identify potential problem areas stands as one of its key features. To achieve this, the plugin concentrates on addressing common errors that arise when converting the original model into a printable file, particularly those relating to scale and printer capacity. As such, the checking process is contingent on the fulfillment of the specific printer parameters to be used. They encompass the maximum height and length that the machine is capable of printing, as well as the minimum thickness. The minimum thickness value can be input as the nozzle size for the filament or an error margin to ensure a reliable outcome. Should the scaled model feature

regions with insufficient thickness for the nozzle, it may not be printed at all or be prone to collapsing during printing. Afterwards, the run check button can be pressed. Some of the checks have to be analyzed case by case because it depends on how the geometry was created.
The following steps outline the checking procedure:

1) Scale Model

The model in the viewer is scaled down to the chosen scale for printing and the measurements are then converted to centimeters. The conversion ensures that regardless of the unit used, users can employ the viewer's measuring tools and view the results at target points if desired. However, it is essential to acknowledge certain challenges that may arise during scaling, especially concerning openings or placements. To mitigate potential distortions, it is advisable to re-initialize the plugin when opting for a smaller scale to ensure error-free processing.

2) Check Model Parts for Solidity and Thickness

The visualization colors in the viewer are reset, and all selected entities undergo a comprehensive examination. A loop is employed to identify the Entity Type, and specific cases are individually analyzed based on it:

a) IfcWall or IfcWallStandardCase
   The extension information from the walls are analyzed by calling the IfcEntityExtension. The width information about the wall is sought, leading to two distinct cases. If the wall width is documented, it is compared against the minimal thickness. In cases where the width is smaller, the thickness check is marked as faulty. Since the width documented is not changed with the scale, the value is converted to the current unit (cm) and scale. Alternatively, if the width is not documented, the process proceeds to case c.
b) IfcSlab (for base, floors and roofs)
   The geometry of the slab is assessed, and its extrusion is extracted. If the magnitude of the extrusion is smaller than the minimum thickness, the thickness check is flagged as faulty. In cases where the magnitude cannot be determined, the process moves on to case c.

c) other Entities

For the other types of entity, the entire geometry undergoes a step-by-step analysis:

i) A loop dismantles the geometry case by case until the base geometry remains.

ii) When a Brep is encountered, the edges are checked using the "checkEdges" method from the viewer, which returns a Boolean value. If the value is false, indicating the geometry is not solid, the solid check is flagged as faulty. Subsequently, the Brep is separated into faces, and each face is put back into the “i” step.

iii) Upon encountering a Face, both the faces and their edges are analyzed. If a face has an area smaller than the minimum area (approximation of the area made by the nozzle with the minimum thickness) or an edge that is too small for the nozzle, the entity is marked as having a faulty face check.

Every time a check is performed, it is counted, along with the number of times it is found to be faulty. Finally, the "isFaulty" Bool method is called to determine if any faults were detected in the entity. If true, the entity is painted in red, and its OID is saved in the list of problematic elements; if false, the entity is painted in green.

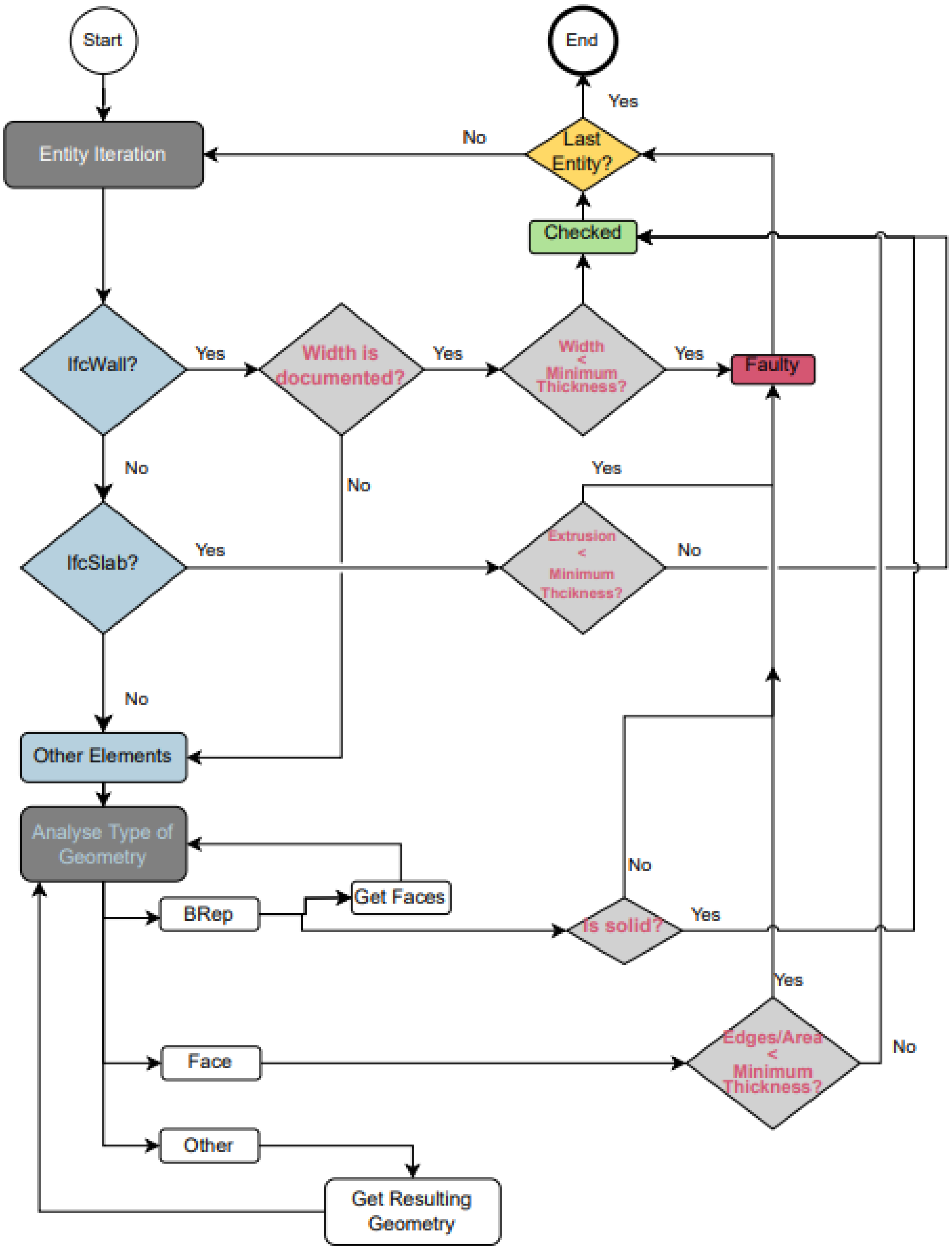


Figure 24: Check Model - Minimum Thickness and Solid Check

3) Maximum length and height Check

To ascertain whether the scaled model adheres to the printer's size capacity, an iteration is conducted for all selected elements. During each interaction, the bounding box of the entity is accessed, and the values for maximum and minimum z, as well as for x and y points, are recorded. Upon analyzing all entities, the absolute minimum and maximum values are calculated. The differences between these values are then compared against the printer parameters of maximum length (x and y) or height (z). If any value exceeds the printer's capacity, the length and/or height check is marked as faulty. Additionally, the middle point of the structure is saved for use as the base point for placing the printer plate.

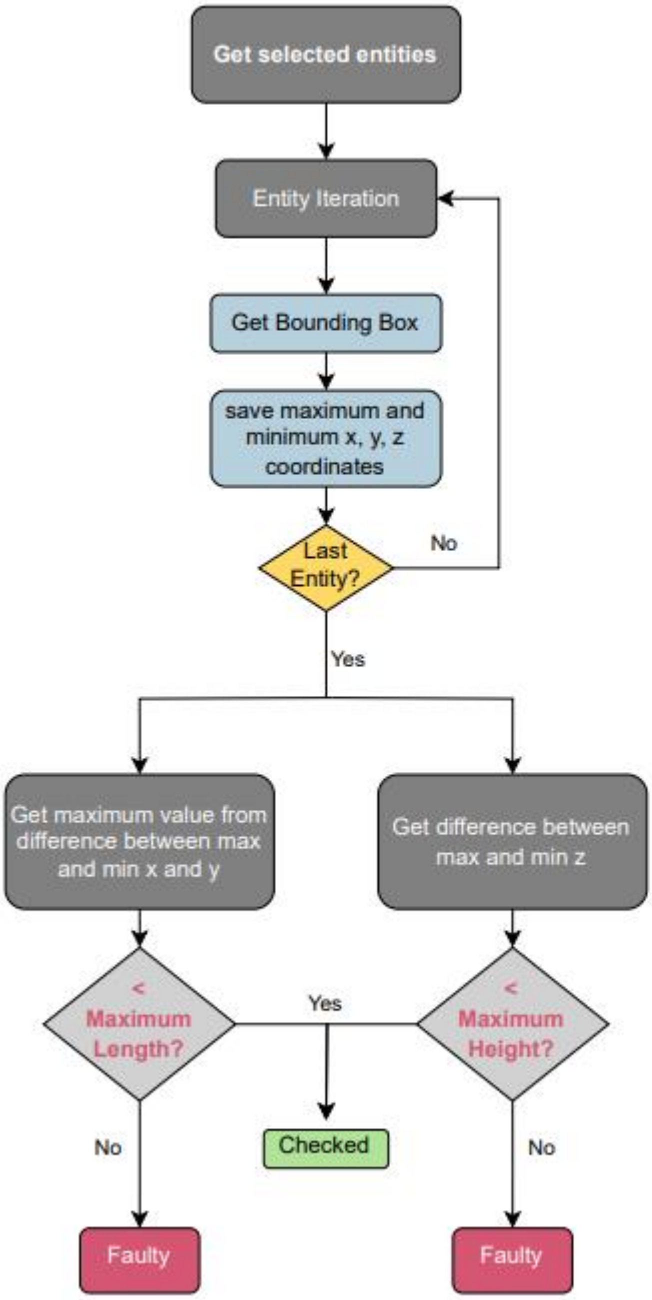


Figure 25: Check Model - Maximum Length and Height

4) Drawing Printer Plate

To provide a clearer visualization of the model's length limits, a plate is drawn in the viewer after the check is performed, with the model positioned at the center of it. For the geometry, a simple rectangular solid is created, with x and y length equal to the maximum printer length and z width of 0.5 cm. The placement matrix is determined, with the origin positioned away from the project origin based on a value calculated from the length parameter and the middle point obtained in the third step. The plate is positioned slightly below the bottom of the model to allow a view from underneath. The geometry is added to the IfcProject as an internal representation with the identifier "Internal_3DPrintingPlate.".

5) Fill Up Check Control

Following completion of all checks, users can visually observe the results in the viewer based on the assigned colors (green for unproblematic and printable, red for problematic) or obtain a report within the plugin dialog. The report provides a detailed overview of each type of check, marked as "checked" when no errors were found. Furthermore, the report highlights messages such as "ERROR: Resulting length too large" or "ERROR: Resulting height too large" when those checks are marked as faulty. For face or solid checks, the report indicates the number of checks made with no error found, along with the total number of checks performed, displayed as "5/20 checked" for instance, representing "number of no errors from/ total number of checks made".

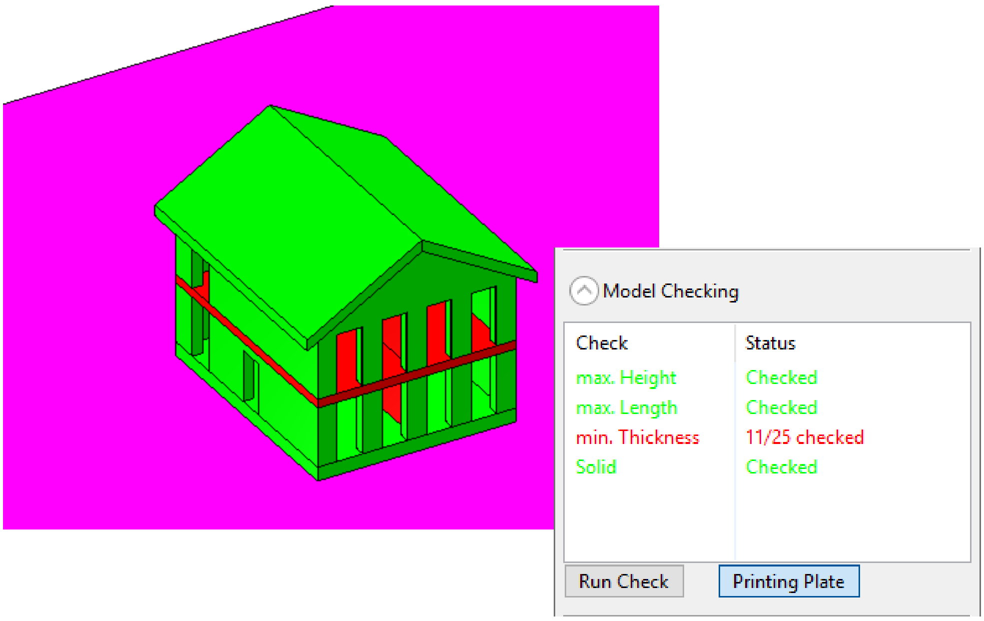


Figure 26: Results of checked model with printable results in green and problematic in red

4.3.6 Export

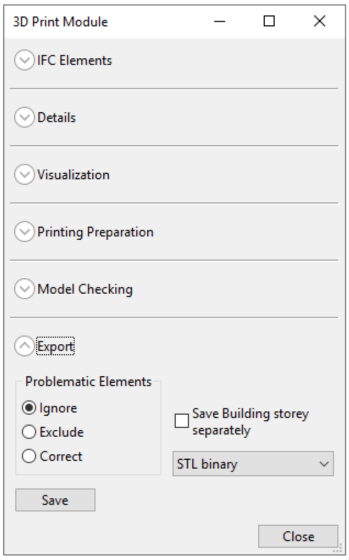


Figure 27: Export

4.3.6.1 Problematic Elements

Any problematic elements identified during the checking phase are recorded within a list, containing their respective OIDs. Subsequently, a decision needs to be made regarding how to address these elements. There are a couple of choices available.

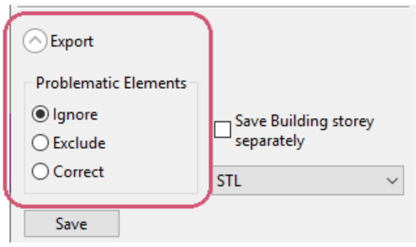


Figure 28: Problematic elements

- Ignore Issues and Proceed: One option is to disregard the identified problems and proceed with attempting to print the model as is.
- Exclude Elements: Alternatively, one can opt for exclusion, meaning the deselection of problematic elements. This course of action guarantees that these elements will be omitted from the printing process. Consequently, any elements highlighted in red within the program will vanish. When this radio button is selected, all elements undergo a filtering process. This filter

operates by identifying the OIDs of problematic subelements, and subsequently marking these subelements or entities as "false”.

- Correct Issues: The last option involves attempting to rectify the issues in question. However, rectifying these elements is not a straightforward endeavor; the approach taken depends on the specific type of error or geometry involved, as well as how these factors impact the overall relationships between the elements. The focus of corrections primarily lies in adjusting the thickness of elements that are too thin for successful printing. Below is the approach taken to rectify such elements:
  - Slabs (roof, floors, etc): Slabs, according to the IFC specification, have the concept of “body SweptSolid Geometry”. That means the “IfcExtrudeAreaSolid” is required, having then profiles that are extruded by certain magnitude in a certain direction, making them relatively straightforward to modify. When a problematic element is recognized as a slab, the extruded geometry is detected, and its magnitude is adjusted to adhere to the required thickness parameter. Here, the IfcOpeningElements must also have the extrusion corrected to adapt to the new thickness.

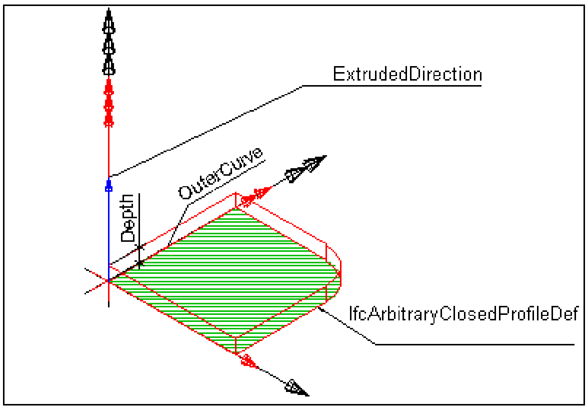


Figure 29: Slabs geometry (buildingSMART, 2023)

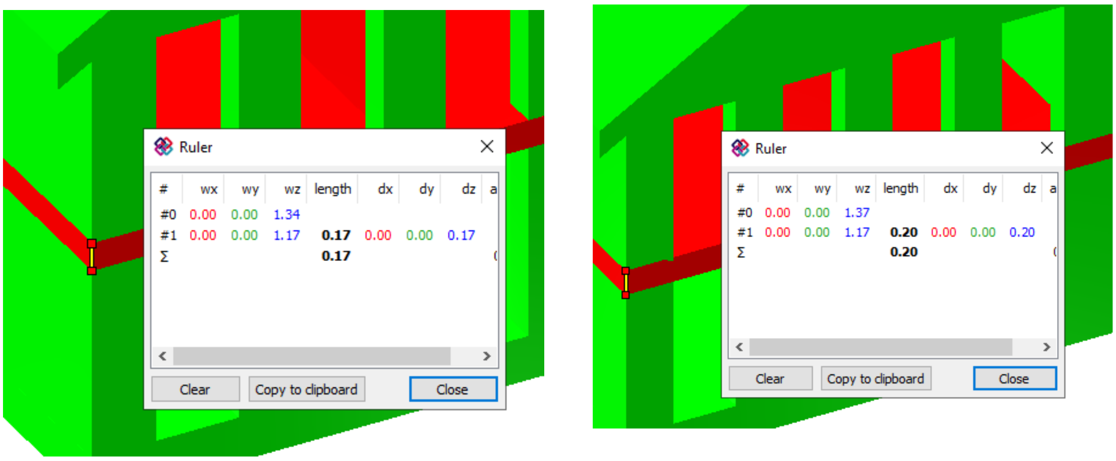


Figure 30: IfcSlab [Floor] before (left) and after (right) corrections

- Walls: Correcting walls is more complex due to variations in their creation and documentation methods. A preliminary approach involves establishing the bounding box of the wall, calculating the disparity between the current thickness and the required minimum thickness, and subsequently scaling the wall's y-axis to meet the specified parameter.

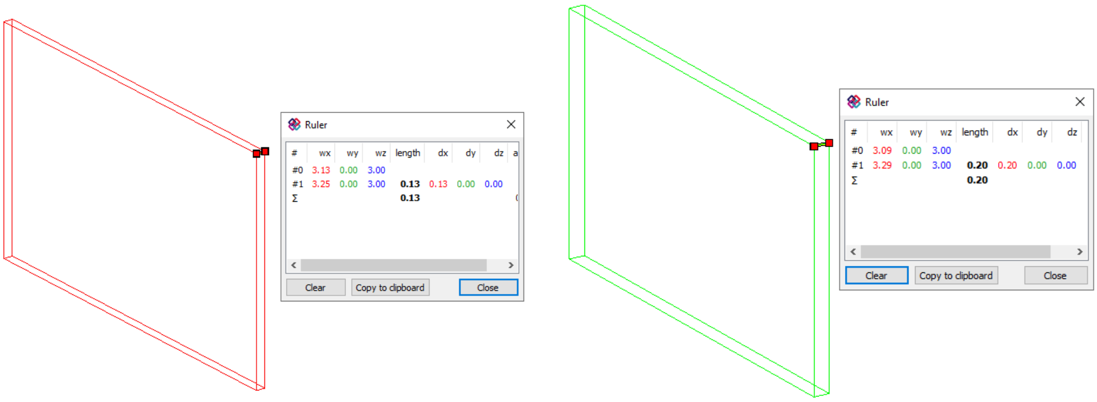


Figure 31: IfcWall thickness before and after correction with minimum thickness parameter of 0.2 cm

4.3.6.2 Separate Storey

One of the goals of the UI was to enable users to print each storey of a model independently, ensuring a comprehensive depiction of both the exterior and interior components. To achieve this, the UI incorporates a geometry aid, facilitating seamless alignment and assembly of each storey.

The first step in achieving this objective was to identify the optimal location for placing the geometry aids without compromising the visual representation. For simple designs featuring distinct corners, a viable approach entailed strategically situating an adequate number of aids at the interior corners of the exterior walls. This was achieved by sorting the storeys from the base upwards and identifying the external walls accordingly. Subsequently, a footprint of each external wall of current storey was generated to establish the foundation for the geometry aids' implementation. By these footprints will be possible to locate the x- and y- points of the placement and this can then be projected at the slab above, where the geometry can be attached.

It is essential to consider the model storey's direction when addressing placement. If a model is not aligned parallel to the x- and y- origin directions, the resulting footprint might be misplaced due to a misalignment between the entity's origin and the storey's origin. To rectify this, the storey placement matrix was inverted and then multiplied by the entity matrix, providing a suitable base for the footprint generation by using the Formula 4.1, where “M” is the transformation Matrix, “A” is the current wall entity and “B “is the storey’s matrix.

$$B = A \times M$$
$$M = B \times A^{-1} \tag{4.1}$$

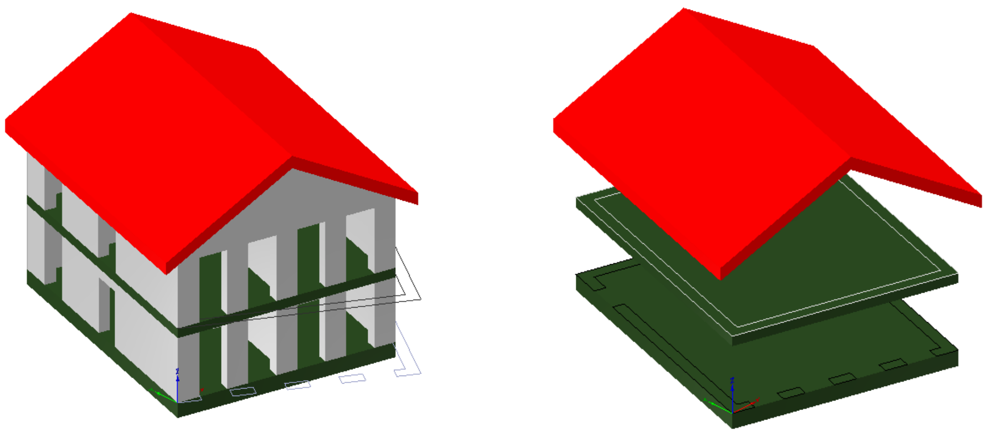


Figure 32: Problems in the axis by generation of footprint and transformation

Upon acquiring the resulting footprint, the analysis proceeds to the subsequent storey, searching for the appropriate entity that will accommodate the aid geometry to be fitted into the walls below (“IfcSlab [floor]” or “IfcSlab [Roof]”). Before analyzing the vertices from the footprint, the axes must be once again transformed. The footprint is set accordingly to the Storey’s entity, but it will be annexed to the slab’s entity. The latter might present different origin and directions from the Storey. For this to be fixed, once again the footprints placement is transformed by using the same Formula 4.1. In this case, “A” being the storey’s matrix and “B” the entities.

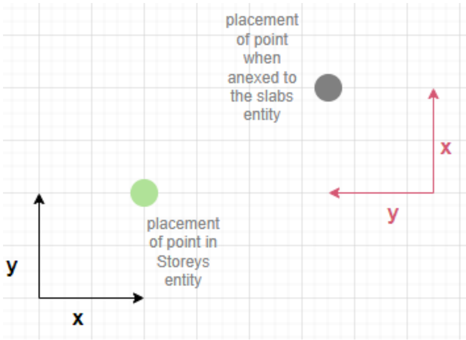


Figure 33: Misplacement of point due to difference in origin and direction

A loop is then established to analyze the points within the footprint. When a point exhibits an angle of 90 degrees between its preceding and succeeding points like seen in the Figure 34, indicating an inner corner, the geometry placement process ensues. Should the angle not meet this criterion, the system advances to the next point for further evaluation. The angles go through the outline of the polylines, which means that the out corners are here excluded, exhibiting a 270 degrees angle and not 90.

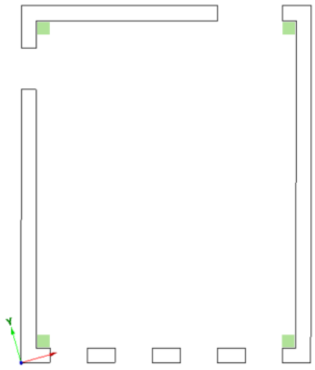


Figure 34: Green points for placement of geometry aid

As a foundational geometry for the aid, the extrusion of a IfcLShapeProfileDef (Profile IfcLShapeProfileDef from the IFC4 specifications) was selected to effectively restrict movements along the x- and y- axes. The profile is generated with a width and depth of 1 cm and thickness with the chosen minimum thickness from the printer parameters. However, it is essential to note that the origin of the placement for this profile, according to the IFC4 specifications utilized by the viewer as seen in Figure 35 (on the left), lies in the middle of its bounding box, not in the corner; resulting in the Figure 35 (on the right) placement, where can be seen that the points don’t match the corner from the profile. Consequently, despite having the desired point, adjustments are necessary to ensure that the corner of the geometry aligns precisely with this point and that it faces the appropriate direction.

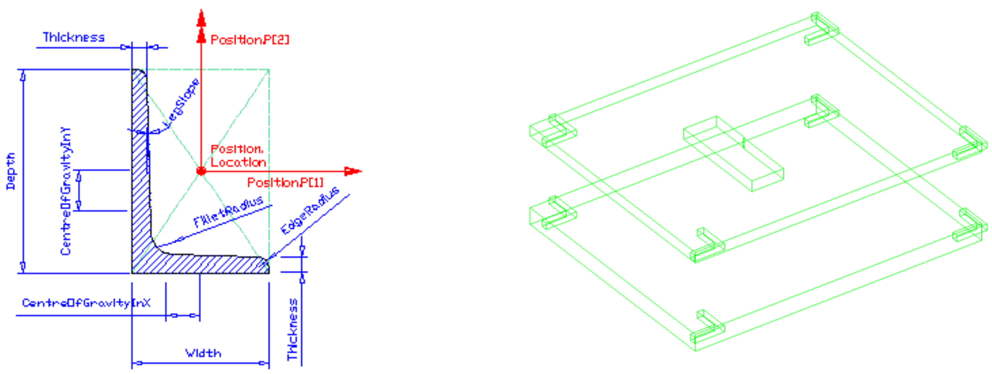


Figure 35: IfcLShapeProfileDef retrieved from the IFC specifications (left) and resulting positioning in model without rotation (right)

For these corrections, the direction from the normalized directions to the previous and next points are examined to determine the suitable direction for the placement correction of the IfcLShapeProfileDef. The points must be normalized before calculating the direction to ensure that the resulting will be in the middle of both directions, as depicted in Figure 36. The IfcLShapeProfileDef is then corrected in the x- and y- axes by the direction times 0.5 cm, which corresponds to half of the bounding box sides of the L profile.

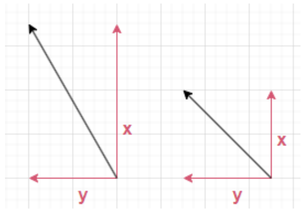


Figure 36: direction IfcLShapeProfileDef placement correction with not normalized and normalized directions

Once the x- and y- axes are appropriately adjusted, the rotation corrections are obtained by creating a new matrix containing the direction to the next point and the normal point of the lower face of the current slab entity, and project the resulting x- and y-points along this matrix so it will point to the right direction and get the right placement in the z axis. The Profile is extruded once it is placed with direction z of -1 and magnitude equal to the minimum thickness.

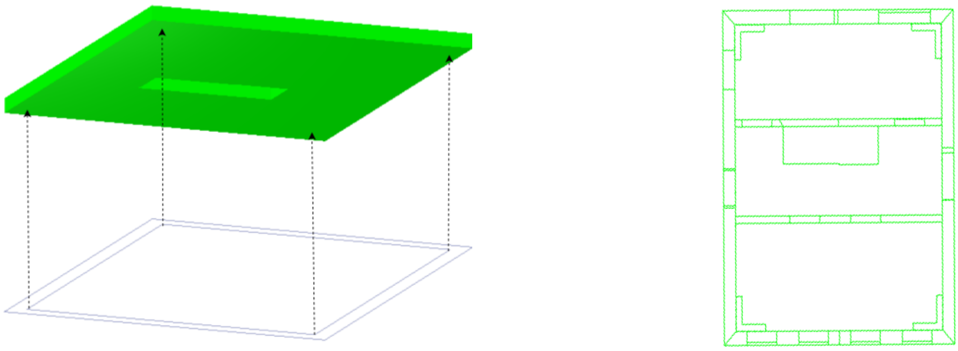


Figure 37: Point projected at face above (left) and corrected positioning of IfcLShapeProfileDef (right)

Since the footprints encompass the entire floor area, and some slab entities are separated into multiple sections - The roof, for example, is in this case composed from two enitites - some geometries, once placed into the entity might stand out as shown in Figure 38. For these cases, a test is made before attaching the geometry to check whether the resulting placement is out of the range of the entity's geometry bounding box. If yes, the geometry is not attached. When analyzing placement and bounding boxes, it is important to determine whether the reference belongs to local (within entity) or global (within project) placement. Geometries with placement that overlaps openings related to the slab entity are also not attached, exemplified by Figure 39, where another profile should be placed at the bottom right corner, but due to the stair opening, it was skipped.

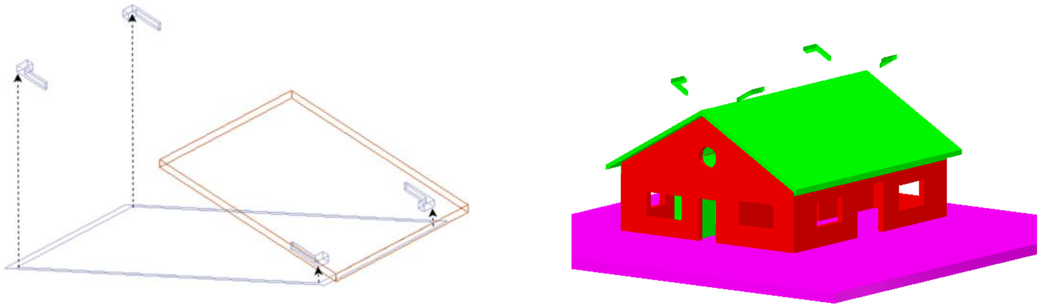


Figure 38: Geometry aids standing out of the model when projected in one side of the roof (left) and resulting view from both roof entities with outstanding geometries (right)

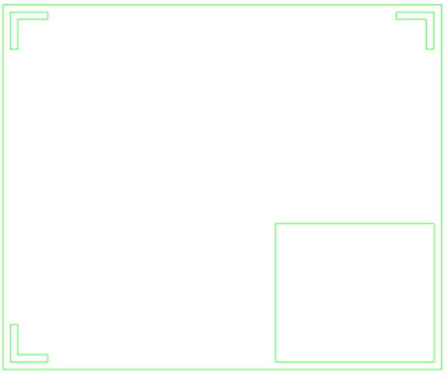


Figure 39: Geometry aid skipped in the bottom left corner due to opening

After attaching all the geometry aids to their respective entities, the storeys are subjected to another iteration. All the children entities for each storey are accessed, and a z-axis movement is applied with an incremental offset that begins at 0 and increases with each new storey. This process enables a preview of each file, allowing visualization of the printing outcome in the viewer exposed in Figure 40. Since the offset is made through the iteration of each storey, just the entities belonging to that storey will be moved. If entities are not documented as belonging to said storey, they will not be moved. The roofs will also be moved by the current offset plus one iteration, so it will be possible to see the interior of the storey that the roof belongs to.

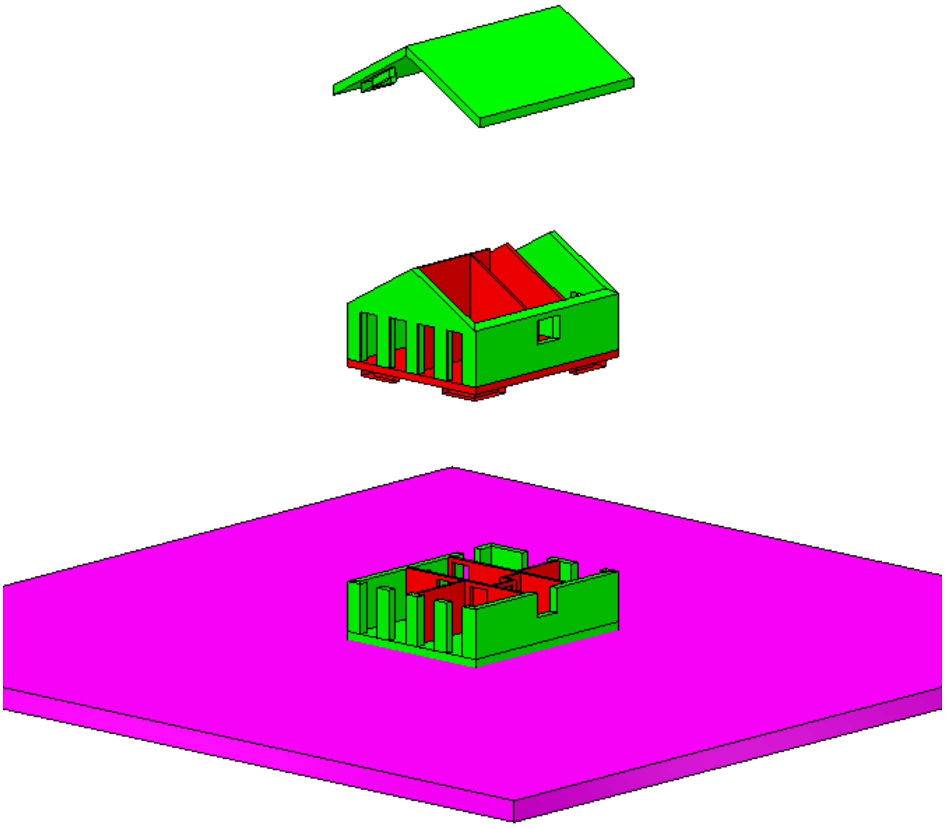


Figure 40: Separated storey view

4.3.6.3 Save

Finally, after being satisfied with the resulting preview of the model, it is ready to be exported. The available options are STL binary (default), ASCII STL, 3MF and OBJ. The selection is made before saving so the user can be informed of all the options. By saving the printable file, they choose a name and the target folder. Additionally, by previously selecting the “Save Building storey separately” option, each storey is saved in a different file. They are named with “name_StoreyN”, with “name” being the one chosen in the directory and “N” the number of the storey. The roof is also saved as a different file, being named “name_RoofN”, with “N” being the group of roof entities that belong to one storey.

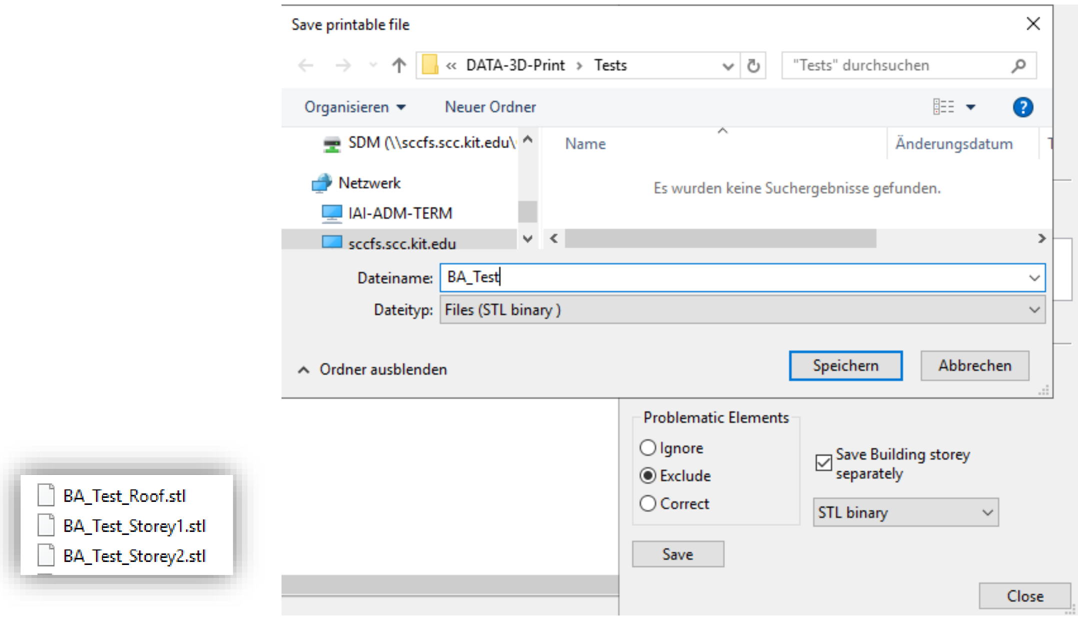


Figure 41: Save file

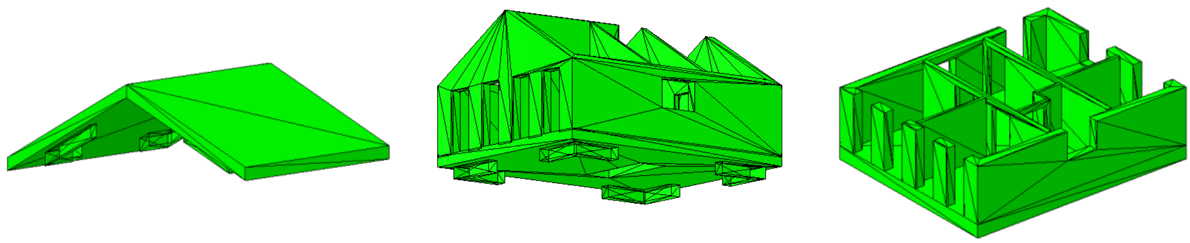


Figure 42: resulting STL files

5 Final Tests and Evaluation

Several 3D models were printed to assess the module's performance and to evaluate the outcomes. The tests are divided into two parts. The first part involves evaluating the use case that was illustrated in the implementation chapter. The second part is closely tied to the initial tests from chapter 3 for comparison. The objectives of these tests are as follows:

1) Compare the results obtained from the plugin checks with the actual outcome and verify if the checking feature effectively identified problematic areas in the final print.
2) Assess the effectiveness of the model's filters and corrections.
3) Evaluate the "separate by storey" option, which entails observing the geometry aids to determine they were well located and their functionality.

Part one: “Musterhaus”

- Test 1: The initial test involves printing only the building shell at a scale of 1:100. This test utilizes the filter elements from the plugin without significant alterations to the slicer settings. For the sake of time management, the model was again scaled by half in the slicer, which will also help to determine whether the epsilon was well defined. It's important to note that all the “Musterhaus” models will ultimately be presented at the same scale. The primary objective here is to observe the results when someone with limited knowledge of 3D printers and no time for extensive learning interacts with the technology.

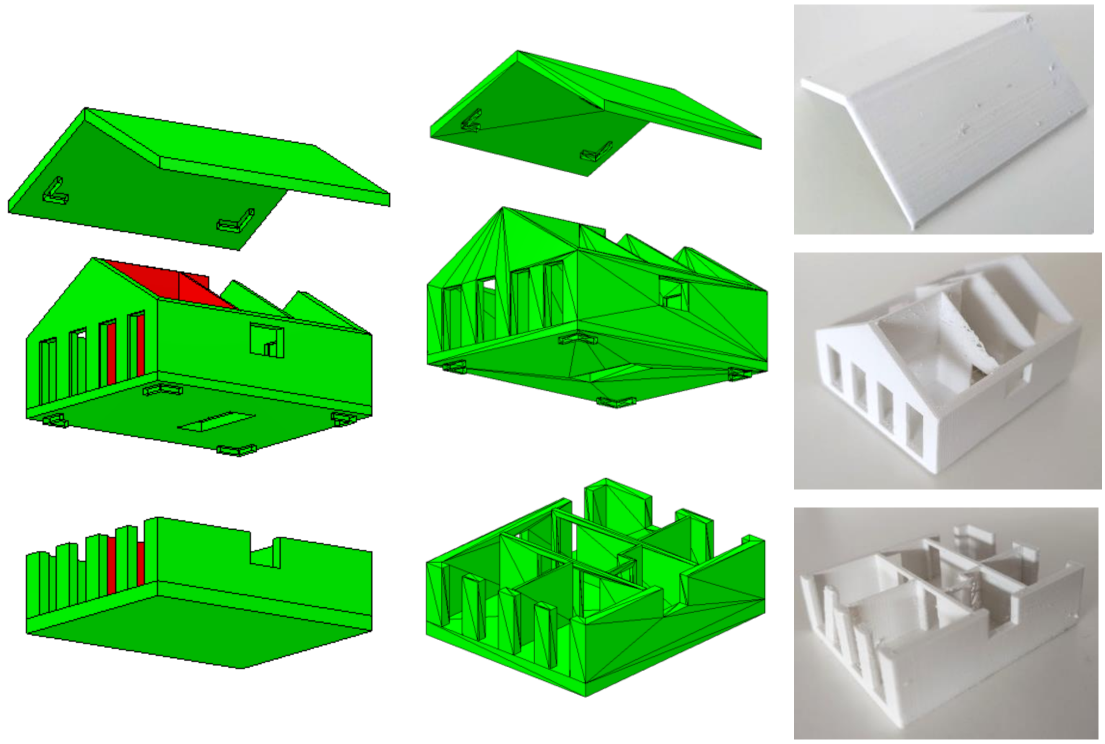


Figure 43: Final Test 1 – IFC (left) to STL (middle) to printed model (right)

Figure 44: Final Test 1 - Assembly aid geometries in the roof (left), slab (middle) and put together (right)

- Test 2: The same settings as Test 1 in the plugin was employed, but adjusting the slicer settings for the model. This scenario simulates the outcome when an individual possesses a modest understanding of 3D printing and can make informed decisions regarding factors like the appropriate printer nozzle, strategic support placement, and suitable printing material. Notably, the only area where support structures were generated was within the middle floor, specifically for the geometry aids.

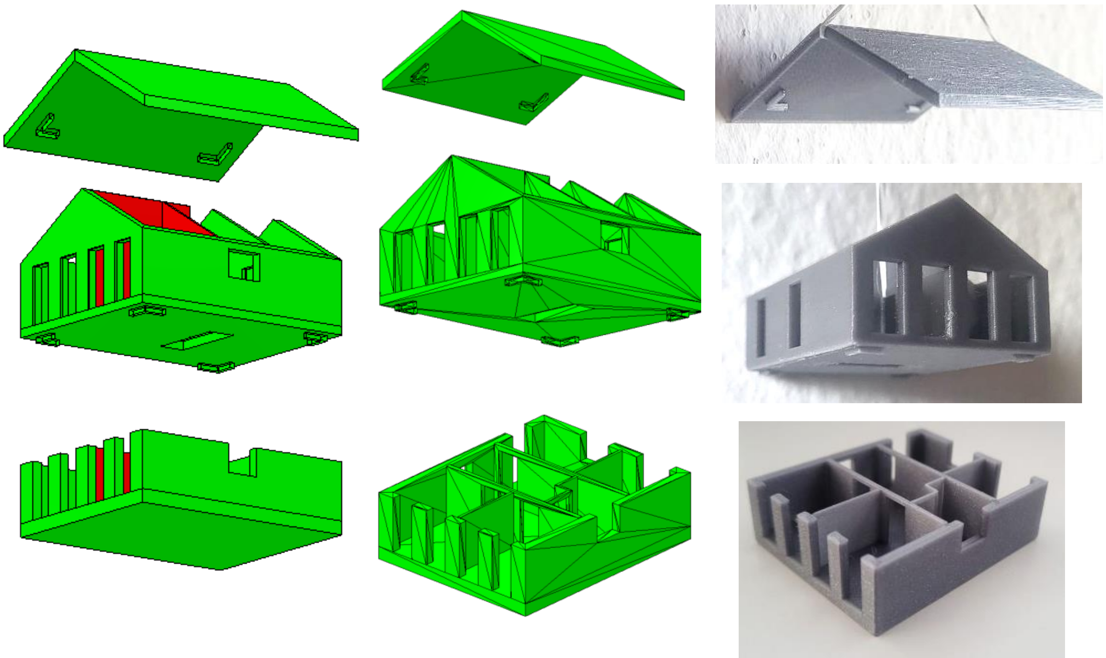


Figure 45: Final Test 2 – IFC (left) to STL (middle) to printed model (right)

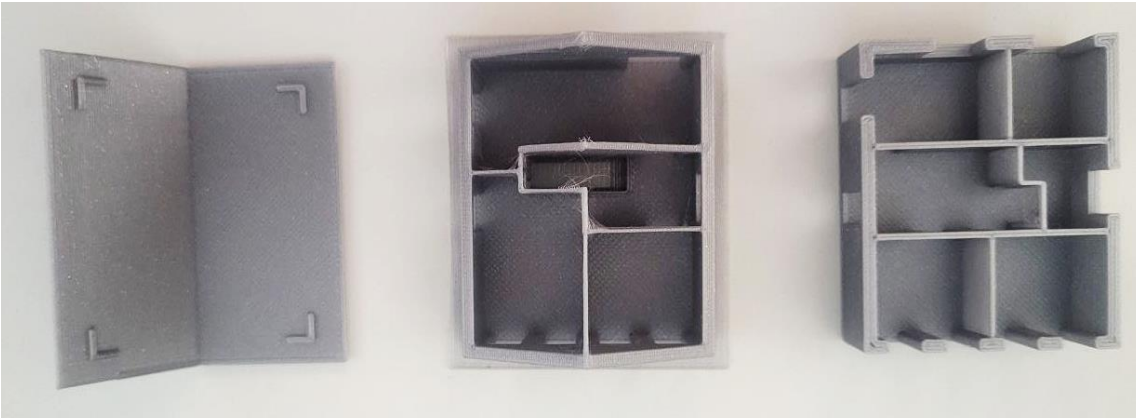


Figure 46: Interior From Final Test 1

Figure 47: Support from Final Test 2 (left) and storey fit together (right)

- Test 3: Implementing the actual parameters demonstrated in the example from the implementation chapter. This involves a scale of 1:200 with problematic elements correction and separated floor. No modifications were made to the slicer settings, and support structures were not utilized.

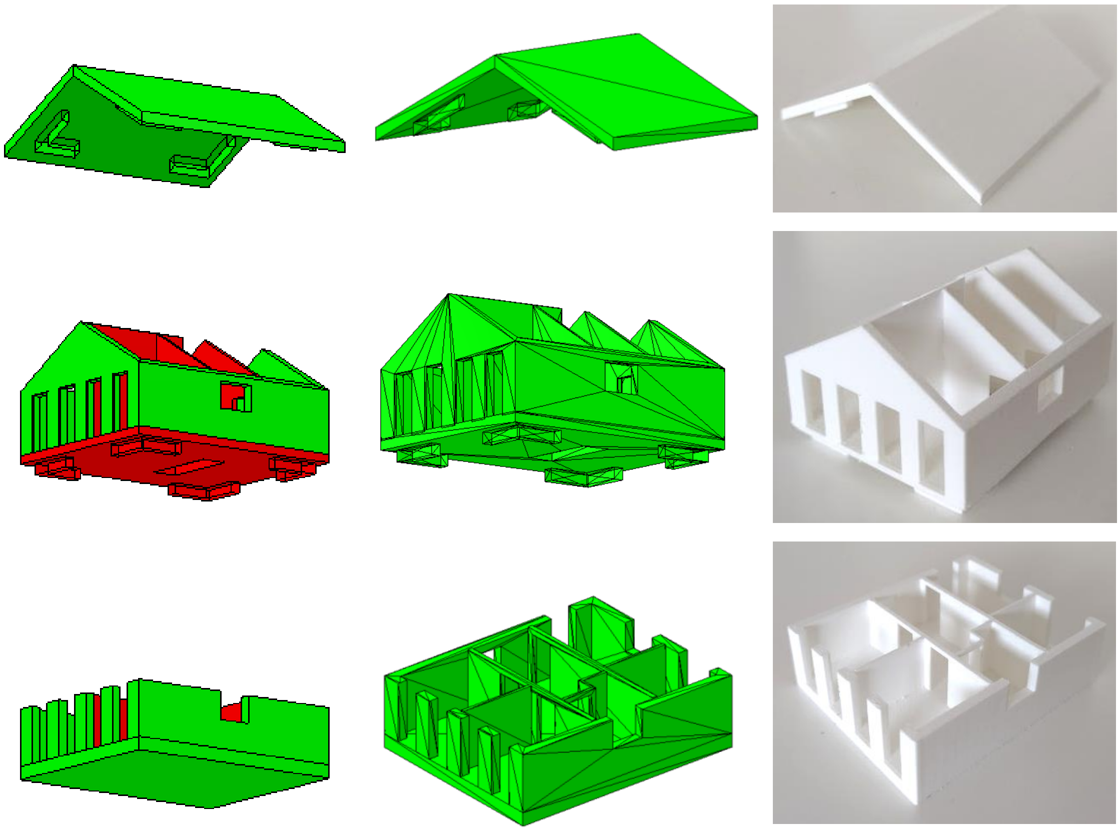


Figure 48: Final Test 3 – IFC (left) to STL (middle) to printed model (right)

Figure 49: Final Test 3 - Assembly aid geometries in the roof (left), floor (middle) and put together (right)

There were no significant change in the time printing time for the part 1 tests. Test 1 exhibited certain issues during the printing process, particularly in areas that had been identified as problematic by the plugin. These concerns are evident in the interior walls depicted in Figure 43 (on the right). While the assembly aids emerged slightly undersized, they remain functional.

Test 2 yielded good results. Although there is still some evidence of stringing in the thin walls, as depicted in Figure 46, every element is well-defined. Despite the continued thinness of the assembly aid, appropriate support parameters in the slicer contributed to a satisfactory fit.

In the case of Test 3, thicker walls and slabs were achieved due to the implemented corrections, positively impacting the geometries. All elements are clearly defined without any sign of stringing. However, an issue arose due to inadequate support for the middle storey. The excess weight resulted in a slight sagging of the slab, creating a clash with the inner walls below. Consequently, this prevents the floors from fitting seamlessly, as highlighted in Figure 49 (on the right).

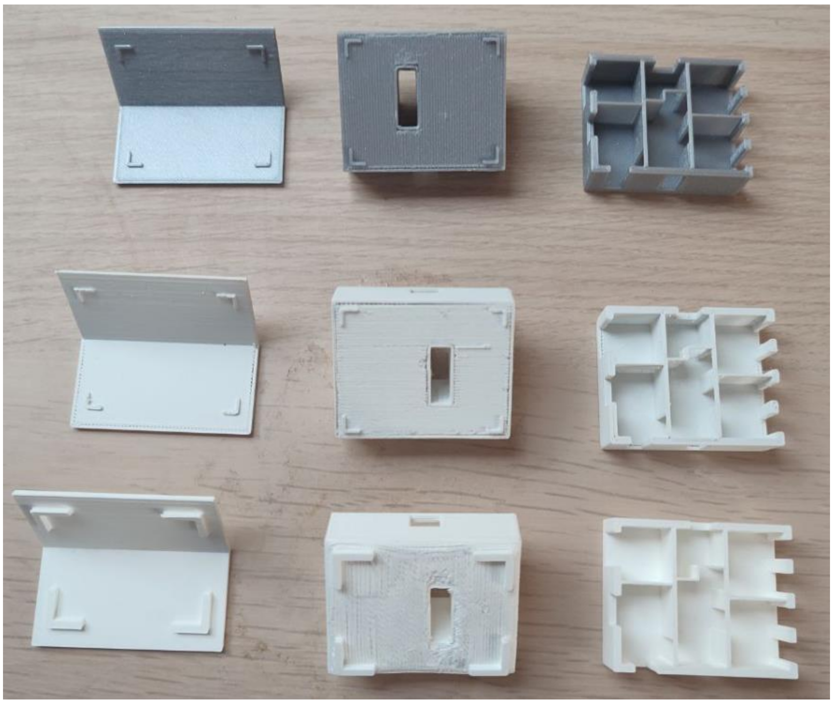


Figure 50: Final tests comparison part 1 - Assembly aid geometries in Test 1 (middle), Test 2 (top) and Test 3 (bottom)

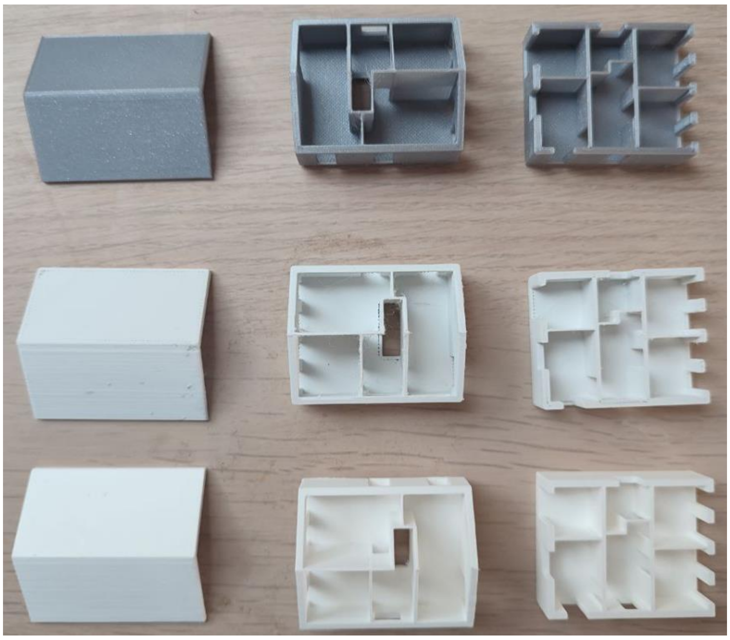


Figure 51: Final tests comparison part 1 – Inner view in Test 1 (middle), Test 2 (top) and Test 3 (bottom)

Figure 52: Final tests comparison part 1 – Whole model in Test 1 (left), Test 2 (middle) and Test 3 (right)

Part 2: Initial Tests Comparison

- Test 4: The FZK-Haus was once again printed. The elements were filtered and the floors separated. In Figure 54 (on the left), the exterior walls are highlighted in red. This occurred due to the solid error check triggered by the presence of windows and door openings not well defined. By employing the KITModelViewer's more detailed check, it can be seen in Figure 53 that certain edges were defined more than once. Typically, such errors are automatically rectified by the slicer during processing. The final outcome consisted of well-defined elements devoid of any stringing, with components fitting together seamlessly.

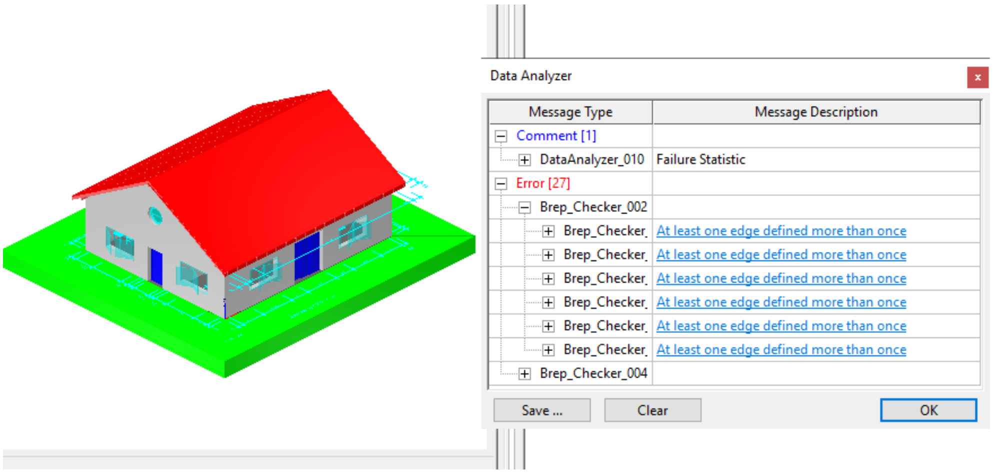


Figure 53: KITModelViewer Data Analyzer

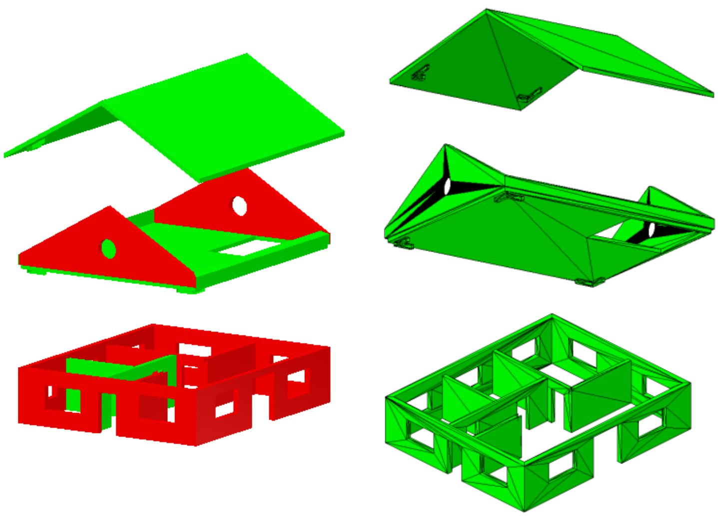


Figure 54: Final Tests part 2 - Test 4 IFC file (left) to STL (right)

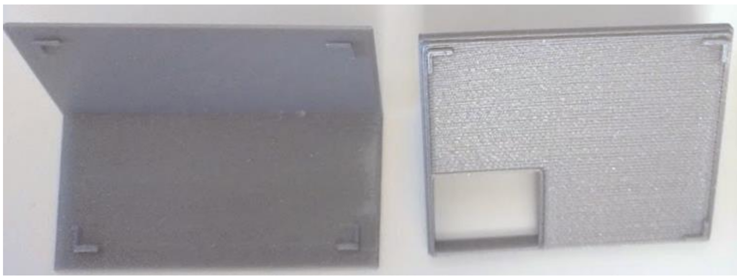


Figure 55: Final Tests part 2 - Test 4 printed model with Assembly aid

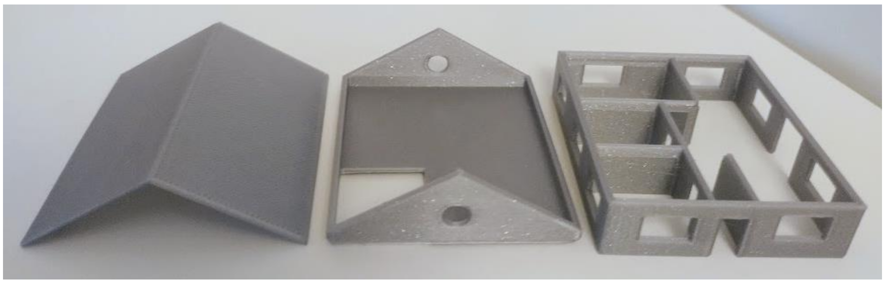


Figure 56: Final Tests part 2 - Test 4 printed model separated

Figure 57: Final Tests part 2 - Test 4 printed model

- Test 5: The IAI building was printed utilizing the plugin to filter elements leaving just the floors above the ground level. Elements like rails and stairs that would likely pose printing challenges were excluded, and certain corrections were attempted. Within the plugin, it was evident that the entire model was flagged as problematic as can be seen in Figure 58, accounting for the unfavorable outcomes witnessed in the preliminary tests. To address this, the correction function was invoked to improve wall and slab thickness. Subsequent processing involved no additional adjustments during slicing, and support structures were omitted.

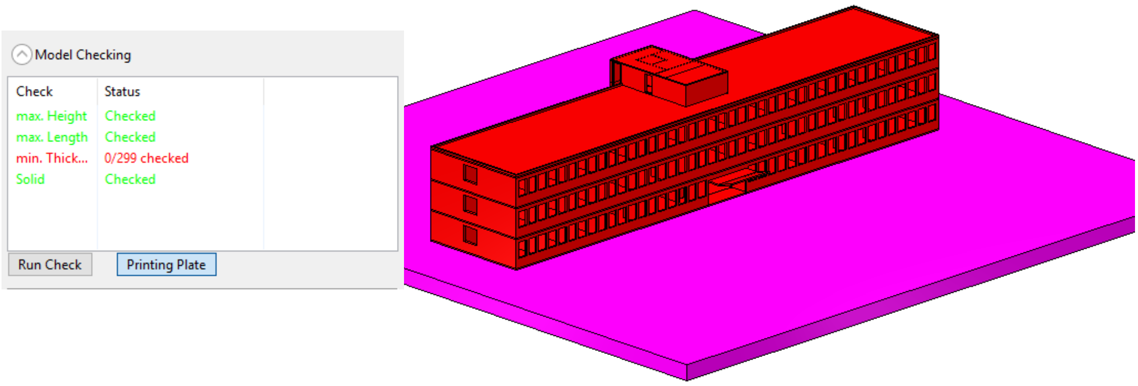


Figure 58: IFC file after checking tool

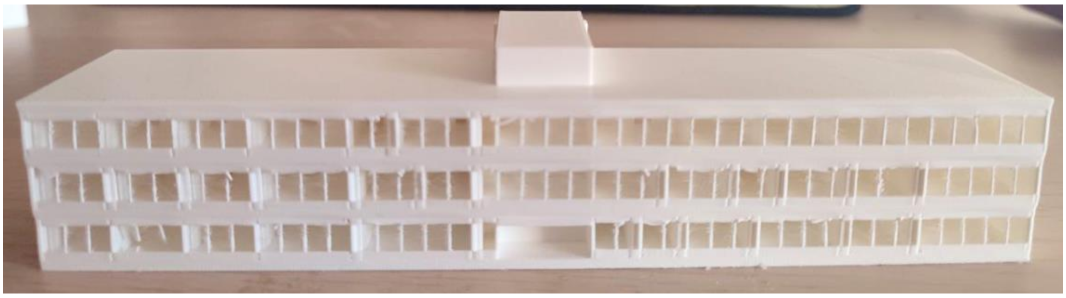


Figure 59: Final Tests part 2 - Test 5 printed model front view

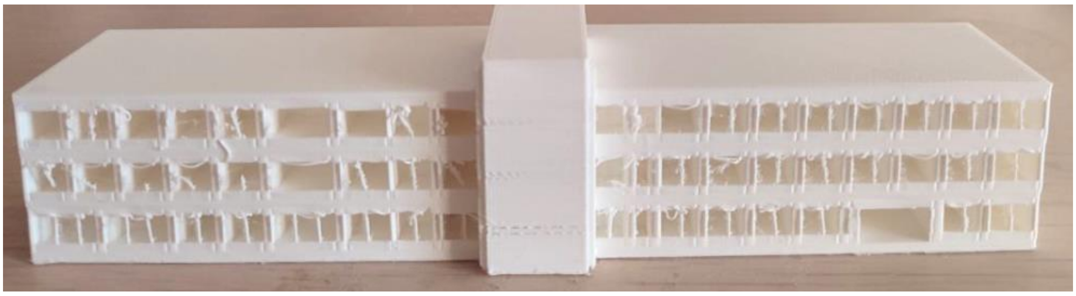
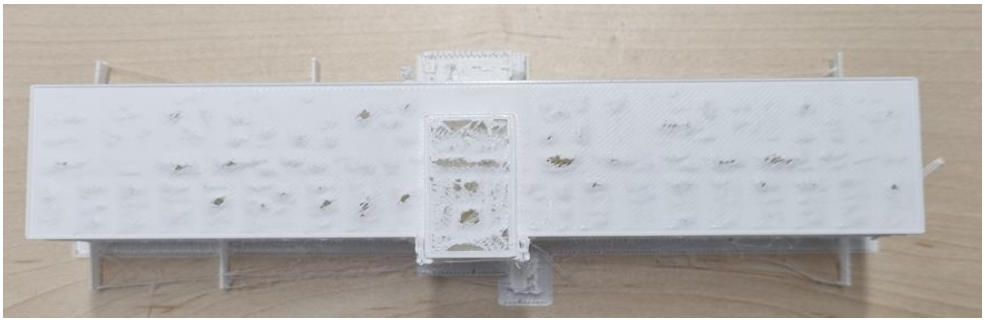


Figure 60: Final Tests part 2 - Test 5 printed model back view

The end outcome demonstrates a substantially improved building outline with notably enhanced stability, as depicted in Figure 61. Nevertheless, the windows, as shown in Figures 59 and 60, remain fragile and not well defined.



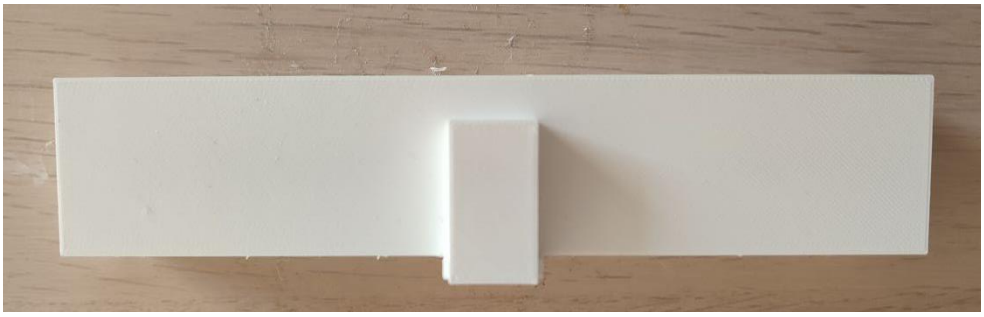


Figure 61: Final Tests part 2 - Building from IAI before (top) and after (bottom) treatment in the plugin

Overall, the plugin's tests yielded promising results, serving as a commendable starting point. Basic elements were better defined and fewer issues were presented in the outcome, such as stringing and dropping parts. The plugin effectively highlighted problematic areas encountered during the scale model printing, indicating components that require careful handling, even if a satisfactory solution is not immediately apparent. The flexibility of the testing modifications proved advantageous, requiring only parameter adjustments to align with printer capabilities and settings, which ultimately also facilitating the printer operator's tasks.

Furthermore, the ability to preemptively address modifications prior to exporting to the slicer offers a distinct advantage, particularly considering the time-consuming process of transitioning between software platforms to identify and address problematic sections. This approach avoids such inefficiencies.

While the majority of tests were conducted using basic models, some features will face some challenges depending on the complexity and documentation fond in project. This was exemplified by the standard properties filter, which only functions when appropriately documented or by the separate storey feature that may find difficulty in its placement or geometry depending on the walls’ footprints or documentation. The correction of problematic elements also encountered obstacles, and improvements are planned for implementation. Notably, the correction of wall elements exhibits a range of efficacy, and its success tends to diminish with increasing complexity. During module implementation, performance degradation was observed with larger, more intricate files, resulting in software sluggishness that constrained testing within the timeframe of this thesis. In the future, expanded evaluations encompassing diverse scenarios are envisioned to optimize results.

Throughout the implementation process, the identification of areas for workflow enhancement emerged. As the file size increases, the task of identifying target entities becomes more challenging, becoming crucial to search and select/ deselect specific entities in a faster way. More intricate filtering through properties and attributes, the ability to selectively address problematic elements—whether by correction, exclusion, or ignoring—instead of applying a uniform solution to all elements, stands out as a potential advantage. Additionally, the capability to save all modifications made during the process would facilitate further work at a later time. Importantly, these advancements are intended to be integrated while maintaining a user-friendly interface to prevent undue complexity.

6 Conclusion

The introduction of Additive Manufacturing (AM) technology has brought revolutionary changes to various different industries. For the construction sector, it holds promises of facilitating processes, while enhancing creativity, reducing costs and applying sustainable practices. Its implementation unlocks new dimensions for traditional use-cases and workflows, as evidenced by the integration of AM in both real-scale constructions or small-scale models. While small-scale models serve as useful visualization tools and platforms for simulations, the transition from digital Building Information Modeling (BIM) models to physical prints is not always smooth. The digital model not meeting the printer settings usually requires remodeling, and the lack of understanding in the 3D printing process results usually in unwanted results. Collaborative BIM projects involve stakeholders with diverse visualization approaches, where printing might not be a priority. Some missteps that might occur during printing include printing of unnecessary elements resulting in waste of material, the transformation of complex or big elements to the small scale resulting in lack of details, deformation or the entire disappearance of it.

Addressing these challenges, this thesis presents the introduction of a preparatory module that can be integrated to the BIM viewer KITModelViewer. This module aims to make quick model adjustments and corrections for efficient printing, reducing material and time waste.

The implementation of the module aims to follow a user-friendly dialog workflow while keeping the main aspects from the original conception. Given the extensive scope of Industry Foundation Classes (IFC) specifications and the diversity in BIM models, the implementation attempts to find solutions that can be applied in all scenarios, where the proposed task will be successful independent of how it was documented. While some tasks have achieved this objective, others can be applied in some cases, but requires further implementation.

The conducted tests on the plugin demonstrated notable improvements in the printing process. By refining fundamental elements, issues like stringing and detached parts were effectively addressed. The plugin's ability to identify problematic areas during scale model printing proved invaluable, even in cases where immediate solutions were still not available.

As outlined in the initial goals, the plugin's achievements closely align with its intended purpose. The optimization of the conversion process, seamless integration with the KITModelViewer, and targeted entity selection based on properties underscore its commitment to enhancing efficiency and accessibility. The plugin's comprehensive checks, including solidity verification, thickness assessment, and scaled dimensions' confirmation, ensure compatibility and accuracy for 3D printing. Discussions around modifying problematic elements and adjusting their thickness further demonstrate its dedication to customization for specific printing needs.

In summary, while challenges were encountered with intricate and big models, the plugin's future holds promise for enhanced results through further implementation. In the forthcoming stages, detailed evaluations and tests are anticipated to further optimize the plugin's capabilities. These enhancements aim to strike a balance between efficiency and user-friendly interface, thereby advancing the evolution of effective 3D printing workflows for BIM model.

Bibliography

Autodesk. (2023, August 10). Retrieved from Revit Software: https://www.autodesk.de/products/revit/overview?term=1-YEAR&tab=subscription

blender.org. (2023, August 10). Retrieved from https://www.blender.org/

Borrmann, A. (2018). Building Information Modeling: Technol-ogy Foundations and Industry Practice. Springer International.

buildingSMART. (2023, July 6). Retrieved from Industry Foundation Classes (IFC): https://technical.buildingsmart.org/standards/ifc

buildingSMART International. (16, July 2023). Retrieved from https://www.buildingsmart.org/

Chen, L., He, Y., Yang, Y., Niu, S., & Ren, H. (2017). The research status and development trend of additive manufacturing technology.

Di, S. (2023). *Konzeptionierung eines automatisierten Arbeitsablaufs im skalierten FDM-3D-Druck mit dem Ziel der Generierung von druckfähigen Modellen aus BIM-Daten.* Karlsruhe.

Dickson, C. (2023, August 5). *3D Printing Wiz*. Retrieved from Binary vs. ASCII STL: Choose Best File Format For 3D Printer: https://3dprintingwiz.com/binary-vs-ascii-stl/

Domínguez, I. A., Romero, L., Espinosa, M. M., & Domínguez, M. (2013, August). 3D Printing of Models and Prototypes in Architecture and Construction. *Revista de la Construcción*, p. 16.

Douglas, K. (2023, April 9). *All3DP*. Retrieved from The Main 3D Printing File Formats of 2022: https://all3dp.com/2/3d-file-format-3d-model-types/

Eastman, C. M. (1999). *uilding Product Models: Computer Environments Supporting Design and Construction.* CRC-Press.

Graphisoft. (2023, August 10). Retrieved from Archicad – Graphisoft Deutschland: https://graphisoft.com/de/archicad

Haupt, T. C. (2022). *Construction in 5D: Deconstruction, Digitalization, Disruption, Disaster, Development: Proceedings of the 15th Built Environment Conference.* Springer International Publishing.

IAI- KIT. (2023, May 08). Retrieved from https://www.iai.kit.edu/english/1302.php

Krämer, M. (2023). BIM in Real Estate Operations: Application, Implementation, Digitalization Trends and Case Studies. Wiesbaden: Springer Fachmedien.

Materialise. (2023, August 10). Retrieved from Materialise Magics: https://www.materialise.com/en/industrial/software/magics-data-build-preparation?gad=1&gclid=CjwKCAjwxOymBhAFEiwAnodBLJRU568yCXe3B0rDVH5KpnSbZQK1hkH3iHzP23StlvRP2nrilKCe3BoC7B4QAvD_BwE

Noardo, F., Arroyo Ohori, K., Krijnen, T., & Stoter, J. (2021). An Inspection of IFC Models from Practice. *Applied Sciences*.

Petrie, R. (2023, July 15). *buildingSMART International*. Retrieved from openBIM Definition: https://www.buildingsmart.org/about/openbim/openbim-definition/

Prusa Research by Josef Prusa. (2023, August 10). Retrieved from Prusa Slicer: https://www.prusa3d.com/page/prusaslicer_424/

Rangel, B. (2023). *3D Printing for Construction with Alternative Materials.* Springer International Publishing.

Römer, F. (2023, July 29). *Heildelberg24*. Retrieved from https://www.heidelberg24.de/baden-wuerttemberg/bauen-kraus-peri-europas-groesstes-gebaeude-3d-drucker-heidelberg-materials-spezial-beton-zr-92237651.html

Sayegh, S. (2020). *A critical review of 3D printing in construction: benefits, challenges, and risks.* Wroclaw University of Science and Technology.

Siemens. (2023, August 10). Retrieved from NX CAM software: https://plm.sw.siemens.com/en-US/nx/manufacturing/cam-software/?stc=dedi100034&gclid=CjwKCAjwxOymBhAFEiwAnodBLIzVG

UguioFCndYpjo6f0GHfOnUJE5Q6-DEugsIYnTKPvppPU1dDFRoCtDcQAvD_BwE

Singh, R. a. (2018). *Additive Manufacturing: Applications and Innovations.* CRC Press/Taylor & Francis Group.

Srivastava, M. (2019). *Additive Manufacturing: Fundamentals and Advancements.* CRC Press.

UltiMaker. (2023, August 10). Retrieved from UltiMaker Cura: https://ultimaker.com/software/ultimaker-cura/

(2020). *VDI 2552-11.* Verein Deutscher Ingenieure.

wxWidgets Cross-Platform GUI Library. (2023, August 10). Retrieved from https://www.wxwidgets.org/

Xu, W., Huang, S., Han, D., Zhang, Z., Gao, Y., Feng, P., & Zhang, D. (2022). Toward automated construction: The design-to-printing workflow for a robotic in-situ 3D printed. *Case Studies in Construction Materials*.

Appendix

List of Figures

List of Tables

List of Abbreviations

3DP	3 Dimensional Printing
3MF	3D Manufacturing Format
ABS	Acrylonitrile butadiene styrene
AM	Additive Manufacturing
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
BIM	Building Information Modeling
Brep	Boundary Representation
CAAD	Computer Aided Architectural Design
CAD	Computer Aided Designs
CAM	Computer Aided Manufacturing
CAE	Computer Aided Engineering
CSG	Constructive Solid Geometry
FDM	Fused Deposition Modeling
G-Code	Geometry Code
GUI	Graphical User Interface
IAI	Institute for Automation and applied Informatics
IDM	Information Delivery Manual
IFC	Industry Foundation Class
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
KIT	Karlsruhe Institute of Technology
MEP	Mechanical, electrical and plumbing
MVD	Model View Definition
OBJ	Wavefront Object
PETG	Polyethylene Terephthalate Glycol
PLA	Polylactic Acid
STEP	Standard for the Exchange of Product Data
STL	Standard Triangle Language
UI	User Interface
XML	Extensible Markup Language