Beyond Geometric Conversion: Enriching Industrial CAD Models for BIM-Based Design

Davide Avogaro[1], Maximilian Maria Wurm[2], Johan Wickström[2] and Carlo Zanchetta[3]

[1]Department of Management and Engineering, University of Padova, 36100 Vicenza, Italy
[2]Paul Scherrer Institut PSI, 5232 Villigen PSI, Switzerland
[3]Department of Civil, Environmental and Architectural Engineering, University of Padova, 35131 Padova, Italy

davide.avogaro.2@phd.unipd.it, maximilian.wurm@psi.ch, johan.wickstroem@psi.ch, carlo.zanchetta@unipd.it

Abstract –

Interoperability between industrial and construction digital models remains a persistent challenge, particularly in projects that require the integration of manufacturing machinery within building environments. Manufacturing workflows typically rely on CAD models based on the STEP standard, whereas the Architecture, Engineering, and Construction (AEC) sector adopts BIM methodologies founded on the IFC standard, which integrates geometry with semantic, relational, and contextual information. Although both standards are mature and widely adopted, interoperability remains limited, with the main bottleneck arising from the transformation of industrial CAD models into semantically meaningful BIM representations.

This paper addresses this gap by proposing a lightweight, open-source framework for STEP-to-IFC integration that goes beyond simple geometric conversion. Instead, the framework adapts industrial product logic to satisfy the information requirements of the AEC domain. The proposed procedure reduces assembly complexity by removing unnecessary components, simplifying overly detailed geometries, limiting the number of objects to those strictly required, and enriching them with appropriate semantic properties.

The framework is implemented as a Blender add-on using the Bonsai extension and features an intuitive user interface, making it accessible to practitioners without advanced programming skills. Preliminary results demonstrate its effectiveness in enriching industrial CAD models with BIM-compatible semantics, highlighting its potential for both research and professional applications in complex industrial–building integration scenarios.

Keywords –

STEP to IFC; CAD to BIM; industry and civil interoperability; BIM; Blender; Bonsai

1 Introduction

Interoperability between digital models remains one of the most critical challenges in both the Architecture, Engineering, and Construction (AEC) and industrial domains. The manufacturing industry typically relies on Computer-Aided Design (CAD) systems to design machinery, products, and furniture that will be produced by industries. In contrast, the civil sector generally adopts Building Information Modeling (BIM) methodologies, which integrate geometric representations with structured product and semantic information. In the design of industrial or production plants, as well as in certain specialized projects, machinery must be integrated within buildings that are required to accommodate, protect, and support all systems necessary for their operation. Consequently, it is essential for civil engineers to import models into their authoring software that provide comprehensive information necessary for designing buildings, systems, supports, and related infrastructure. By integrating these domains, the design process can be accelerated, achieve higher-quality outcomes, and reduce the likelihood of errors. Such projects are commonly classified as complex, as they are not only complicated, multifaceted, and composed of numerous interconnected components [1], but also characterized by an additional layer of complexity arising from CAD–BIM integration.

Within their respective domains, manufacturing and construction rely on well-established standards for representing three-dimensional and information-rich models: the Standard for the Exchange of Product Data (STEP) format [2] in manufacturing, and the Industry Foundation Classes (IFC) format [3] in construction. Although both standards are based on explicit and formally defined data structures, effective interoperability between the two domains remains limited due to substantial differences in conceptual modeling paradigms, levels of detail, and intended use

cases. Notably, the primary challenge does not lie in data transfer from the civil to the industrial domain (i.e., IFC to STEP), as several tools are available to convert IFC models into simplified STEP geometries, typically renouncing product and semantic data. Instead, the more significant issue concerns the reverse process—from STEP to IFC—because IFC models are required to represent not only geometry but also rich product, relational, and contextual data.

This study proposes a framework that should not be regarded merely as a geometric conversion tool, but rather as a system capable of adapting industrial product logic to the information requirements of the AEC domain based on open-source tools and formats. The proposed approach demonstrates strong potential for application in both academic research and professional practice.

2 Background

As demonstrated in previous work [4], the existing literature on CAD–BIM interoperability between the industrial and AEC domains is limited, with most relevant contributions dating to the 2006–2008 period and lacking alignment with recent standard developments such as STEP AP242 and IFC4.3_ADD2. This situation highlights a clear gap in the current body of knowledge. Moreover, the reviewed studies consistently emphasize the absence of practical, deployable solutions and underscore the urgent need for interoperable, open-source tools capable of mitigating the fragmentation within the AEC sector [5–8].

At present, the most common approaches adopted to compensate for the lack of robust conversion tools include the following practices:

- Importing geometries as meshes, resulting in excessively complex representations that require the manual addition of product properties and relationships. Such meshes are often derived from low-level CAD models [9] that provide only viewable representations, characterized by faceted geometries or "frozen" solids, such as the Stereolithography (STL) format, the Initial Graphics Exchange Specification (IGES), or CATIA Graphics Representation (CGR). Each manual step is prone to error, and the use of highly detailed geometries leads to unnecessarily models.
- Manual redrawing of simplified components across different BIM authoring software, with properties and relationships added manually. This process is time-consuming and highly error-prone.
- Using converters that map all elements to *IfcBuildingElementProxy* or *IfcElementAssembly* entities, which causes semantic degradation, does not address excessive geometric complexity, and generates instances for all parts of a product—including components irrelevant for civil purposes (e.g., screws, bolts, labels). This results in IFC files with an excessive number of objects and redundant information.
- Relying on proprietary ecosystems managing both CAD and BIM environments, which does not resolve data exchange issues with external stakeholders and may conflict with regulatory requirements that mandate open formats, as is often the case in public administration projects. Additionally, large models can cause performance issues, making it difficult to upload and obtain a complete project view.

In summary, current procedures focus primarily on conversion tools without considering the fundamental differences between industrial and civil design logics. Industrial models are typically highly detailed, representing every part of a product to the level required for industrial production. Such models contain a large number of objects (e.g., screws, bolts) and detailed geometries that exceed the information needs of the AEC domain and can significantly compromise BIM performance. In contrast, the AEC sector requires models with a lower level of geometric detail, omitting minor components while preserving overall dimensional information. Moreover, industrial models often lack product properties, requirements, and inter-object relationships, which are typically stored in separate documents or managed via other systems. Conversely, in the IFC format, such data are represented as object properties, enabling professionals to design more efficiently and implement validation procedures to improve quality and save time. Thus, a simple conversion tool is insufficient. What is required is a procedure capable of adapting the logic of industrial products to the information requirements of the AEC domain. An initial attempt to address this challenge was proposed by [4], introducing a framework that leverages open-source technologies to adapt industrial product logic for AEC use. The present work extends this framework, presenting a new version designed to overcome limitations identified in the previous implementation.

3 Materials

A market analysis was conducted to identify open-source tools capable of importing industrial geometries and exporting IFC models suitable for implementing the proposed procedure. This analysis identified two primary candidates: FreeCAD [10] and Blender [11] equipped with the Bonsai extension [12]. Blender and Bonsai were selected due to the availability of documentation and because these tools are specifically developed and widely adopted for civil and BIM-oriented workflows.

Blender does not natively support the direct import of STEP files. To address this limitation, an intermediate exchange format was required that would preserve geometric integrity and structural information. Among the formats supported by Blender, glTF 2.0 (Graphics Library Transmission Format) was selected, as it is an efficient, open, and neutral standard well aligned with Blender's internal architecture. Developed and maintained by the Khronos Group, glTF 2.0 is designed for the efficient transmission and representation of three-dimensional content. It consists of a structured set of files, including a JSON file (.gltf) describing the scene, node hierarchy, materials, cameras, and references to geometry and texture data; one or more binary files (.bin) storing high-density buffered data such as meshes and animations; and image files (.jpg, .png) defining textures associated with materials.

The glTF format was chosen not only for its performance and ability to preserve product hierarchies, but also for its support of custom metadata through the *extras* field. While STEP files typically convey only geometric information and do not include semantic or property data, it is possible to develop a procedure to encode any additional properties associated with industrial components within the *extras* field, thereby enabling their transfer into the BIM environment. The open-source software Mayo [13] was used to convert STEP files into glTF format, which can then be imported into Blender without loss of geometric fidelity.

Blender supports the development of custom add-ons developed in Python, including the creation of tailored user interfaces. This capability enables the implementation of structured and intuitive workflows that are accessible even to users without advanced programming skills. The proposed framework is entirely based on open-source tools, ensuring independence from proprietary software ecosystems and broad accessibility. The overall framework and its operational steps are described in detail in the following section.

4 Framework

4.1 Case study: Paul Scherrer Institut (PSI)

The Paul Scherrer Institut (PSI) is a research center that conducts internationally recognized, cutting-edge research in the natural and engineering sciences. PSI develops, builds, and operates highly complex particle accelerator–based large-scale research facilities, which are made available to both the scientific community and industry. Among its most prominent facilities are the Swiss X-ray Free Electron Laser (SwissFEL) and the Swiss Light Source (SLS) [14].

Due to its extensive building portfolio and the continuous development and upgrading of facilities to incorporate new technologies and enhance its scientific offerings, PSI routinely faces challenges related to the integration of BIM models used for civil design with CAD models employed for the industrial design of its research infrastructure. For clarity, the following sections present a girder from the SLS facility as an illustrative example (Case Study 1, CS1). The framework has also been tested on additional models from the same facility, including a linear accelerator (CS2), a transfer line (CS3), and an undulator (CS4).

4.2 Add-on

The framework is presented step by step in the following sections. The user interface of the Blender add-on developed for this framework is shown in Figure 1 and is available on GitHub [15]. The arrangement of commands in the interface follows the sequential workflow of the procedure, guiding the user through the proper use of the framework. The framework is not limited to the STEP format but is generally applicable to all file formats that can be imported into Blender and that provide a hierarchical decomposition of components.

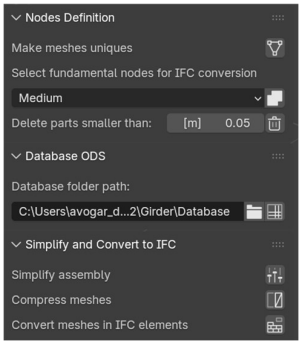

Figure 1. User interface of the proposed Blender add-on that reflects the methodological workflow

4.2.1 Make meshes unique

This operation is required to prevent unexpected behavior and to ensure the correct execution of the procedure. The command standardizes object names by removing unsupported or problematic characters that could cause ambiguities or errors during script execution. In addition, modeling software often employs multi-user meshes when multiple identical objects are present to reduce memory usage, meaning that a single mesh dataset is stored and referenced by multiple distinct objects. While this strategy is effective for optimizing storage, it can lead to errors when mesh data are accessed programmatically, potentially causing script failures. To avoid such issues, all meshes are made unique so that each object has its own mesh, ensuring that it references independent mesh data. This step is required solely to

guarantee the robustness and reliability of the code and does not affect the geometric outcome of the model.

4.2.2 Select fundamental nodes for IFC conversion

This step aims to adapt the complexity of the hierarchical structure of the imported STEP model to meet the requirements of the AEC domain. Well-structured industrial models often feature deeply nested hierarchies to manage and organize the numerous components that make up a product. This hierarchy is expressed through parent nodes containing multiple child nodes. As shown in Figure 2, when an industrial model is imported into the Scene Container, the hierarchy is displayed using two main types of icons: a triangle icon representing mesh objects and an axes icon representing empty objects. Mesh objects consist of vertices, edges, and polygonal faces, whereas empty objects are void transformation nodes used solely for grouping and manipulating child objects. Typically, empty objects act as parent nodes that contain other empty objects and mesh objects. When a parent node is collapsed in the tree view, the number of its child elements is displayed next to its name, as illustrated in Figure 2.

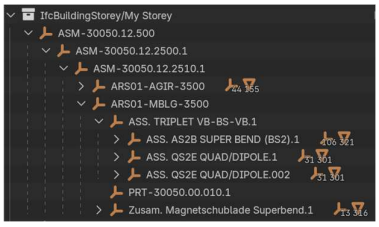

Figure 2. Hierarchical structure of the imported model in its initial assembly tree

For civil purposes, distinguishing every individual part of an industrial product is generally unnecessary. Instead, the focus is on representing the overall product at an appropriate level of decomposition. Figure 3 illustrates a simplified hierarchy in which only the elements relevant for BIM representation are retained, representing the outcome of the proposed procedure.

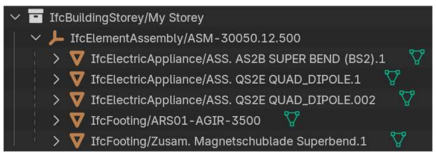

Figure 3. Final assembly tree

Accordingly, the user's first task is to select the fundamental nodes that will correspond to independent IFC objects. Within the assembly tree, a fundamental node represents an object that will be converted into an IFC element. If a fundamental node contains other fundamental nodes as children, it corresponds to an *IfcElementAssembly* aggregating multiple IFC elements. Conversely, if a fundamental node is a leaf in the assembly tree, all of its child mesh components are merged into a single object, as this represents the maximum level of decomposition required for civil applications. So, if a leaf fundamental node is an empty object, all mesh objects contained within its hierarchy are merged into a single mesh object.

To perform this operation, the user selects the desired object, chooses the appropriate Level of Detail (Low, Medium, or High) from the corresponding menu, and then presses the *Select nodes for IFC conversion* button. The concept of Level of Detail is discussed in Section 4.2.5. Once the button is pressed, a custom property named *Node for IFC conversion* is created and associated with the selected object. This property states that the node is a *fundamental node* and is essential for the correct functioning of the subsequent processing scripts.

4.2.3 Delete small parts

This step is optional but highly recommended, as it allows the removal of elements that are unnecessary for civil purposes and typically characterized by a high number of vertices, faces, and triangles, such as screws, bolts, and similar components. The command deletes all meshes that do not have the *Node for IFC conversion* property assigned and whose size is below a user-defined threshold. Meshes marked with the *Node for IFC conversion* property are preserved, as some small components—such as plugs, valves, or comparable elements—may still be relevant and required in the IFC model. By defining an appropriate threshold and activating the *Delete parts smaller than* command, the user can efficiently eliminate insignificant elements while retaining those essential for IFC conversion. Figure 4 shows the difference between the initial state and the model after deleting objects smaller than 0.05 × 0.05 × 0.05 m. As reported in Table 2, in the considered case studies, approximately the 61.2% of the total number of objects are removed through this operation.

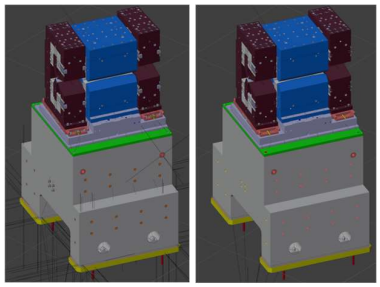

Figure 4. Comparison of the initial state (left) and the girder after removing elements below 0.05 × 0.05 × 0.05 m (right)

4.2.4 Database and ODS printing

For each fundamental node, an OpenDocument Spreadsheet (ODS) file is generated. The ODS format is an open-source spreadsheet standard. The user must specify a directory on their local system where these ODS files will be stored. Each ODS file is named after the corresponding fundamental node, excluding any automatically generated progressive suffixes. In modeling software such as Blender, progressive suffixes (e.g., ASS. AS2B QUAD_DIPOLE.1, ASS. AS2B QUAD_DIPOLE.002) are appended to distinguish objects with identical base names. Since these objects are geometrically and hierarchically identical, generating multiple ODS files with the same content is unnecessary; therefore, a single ODS file is created using only the base name. Each ODS file contains a list of the elements in the hierarchy of the fundamental node after which the file is named. If the fundamental node contains other fundamental nodes as children, as shown in Figure 5, the user is required to define the data for the corresponding IFC objects, including the *IfcClass*, *PredefinedType*, *ObjectType*, and any required property sets (Psets) and properties. If the fundamental node is a leaf node, as illustrated in Figure 6, the user must instead specify data related to the mesh components that will constitute that node. Specifically, for each mesh, the user indicates whether it should be deleted—for example, if it represents an internal or otherwise irrelevant component—or retained. For retained meshes, the user may also specify whether the geometry should be simplified. When geometries are excessively detailed for civil applications, simplification replaces the original mesh with its bounding box, thereby significantly reducing the overall model size.

As illustrated in Figures 5 and 6, the cells requiring user input within the ODS files are highlighted to guide the data-entry process: grey cells must not be filled, dark green cells are mandatory, and light green cells are optional.

Based on the directory specified by the user, the script automatically creates two subfolders—*Completed* and *To be completed*—if they are not already present. For each fundamental node, the script checks whether an ODS file with the corresponding name exists in the *Completed* folder. If no such file is found, a new ODS file is generated in the *To be completed* folder.

After completing the data entry in the ODS files, the user must manually move the files from the *To be completed* folder to the *Completed* folder. This workflow supports the reuse of ODS files across different conversion projects, as multiple models may share identical components—for example, different STEP files of girder models may reference the same *ASS. AS2B QUAD_DIPOLE* magnet, making it unnecessary to compile the corresponding ODS file more than once. By maintaining a single, centralized repository of completed ODS files, redundant data entry is avoided, resulting in improved efficiency and reduced preparation time. In addition, storing manually entered data in a centralized database prevents the need for repeated compilation, which is particularly beneficial in work-in-progress scenarios where the design of industrial products both affects and is affected by the design of the building. This approach enables the procedure to be repeated and updated versions to be shared with minimal or no additional manual effort.

4.2.5 Simplify assembly

Once all parameters have been defined, the geometry can be simplified according to the information specified in the ODS files. For each leaf fundamental node, the system determines the appropriate degree of simplification based on the Level of Detail (LOD) selected by the user. When the LOD is set to *Low*, the script reads the corresponding ODS file, deletes all child mesh components marked for removal by the user, and replaces the geometry of all remaining child components with their respective bounding boxes. When the LOD is set to *Medium*, the script deletes all child components marked for removal and simplifies only those child meshes explicitly flagged for simplification in the ODS file by replacing them with their bounding boxes. When the LOD is set to High, no child mesh components are deleted or simplified. In all cases, the remaining child

Element Name	Product in DB	LOD Preset	To be deleted	MID: To be simplified	IfcClass	Predefined Type	Object Type	Pset_Installation InstallationDate	Pset_Installa PutIntoOper	Pset_Energ PowerDem	Qto_Footing NetWeight
ARS01-AGIR-3500	Yes	MEDIUM			IfcFooting			05.11.2025	18.11.2025		500
ASS. AS2B SUPER BEND (BS2)	Yes	MEDIUM			IfcElectricAppliance	USERDEFINED	Super Bend	06.11.2025	18.11.2025	10000	
ASS. QS2E QUAD_DIPOLE	Yes	MEDIUM			IfcElectricAppliance	USERDEFINED	Magnet	06.11.2025	18.11.2025	5000	
Zusam. Magnetschublade Superbend	Yes	MEDIUM			IfcFooting	USERDEFINED	Support	06.11.2025	18.11.2025		

Figure 5. ODS file of the fundamental node ASM-30050.12

Element Name	Product in DB	LOD Preset	To be deleted	MID: To be simplified	IfcClass	Predefined Type	Object Type	Pset_Nam	Pset_Nam
PRT-30050.41.465			Yes						
AS2A BOTTOM YOKE			No	Yes					
BS2 SIDE YOKE			No	Yes					
NRM-ET-00.7036			Yes						

Figure 6. ODS file of the leaf fundamental node ASS. AS2B SUPER BEND (BS2)

mesh components associated with a fundamental node are merged into their parent node. This process significantly reduces the number of objects in the scene, limiting it to the number of fundamental nodes while preserving the level of geometric abstraction required for the selected LOD. Results of the simplification process for the different LOD levels are shown in Figure 7.

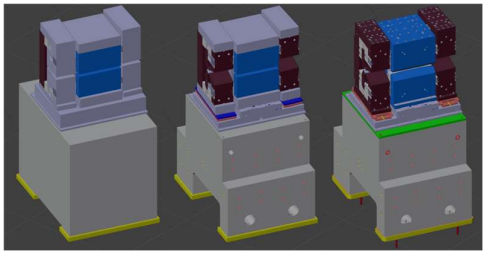

Figure 7. Simplified model outcomes for low (left), medium (center), and high (right) LOD

4.2.6 Compress geometries

To further reduce geometric complexity and file size, the geometries can be optimized using the *Compress geometries* command. This operation merges vertices that are within a defined proximity threshold, removes duplicate vertices, converts triangular faces into quadrilateral faces where possible, deletes vertices and edges not associated with any faces, and enforces geometric planarity. These optimizations reduce the overall complexity of the geometry during IFC export, resulting in significantly lighter IFC files.

4.2.7 IFC conversion

The final step converts the processed geometries into IFC elements using Bonsai capabilities. For each fundamental node, the script reads the IFC attributes and properties defined by the user in the corresponding ODS file and generates the appropriate IFC entities, as shown in Figure 3. Decomposition relationships between parent and child elements are established through *IfcRelAggregates* relationships, thereby explicitly representing the hierarchical structure of the product.

The conversion process is compatible with all IFC versions supported by Bonsai. However, the user must ensure that the selected IFC classes, attributes, and property sets (Psets) are valid for the chosen IFC schema version, as inconsistencies or unsupported definitions may cause the script to fail. For this reason, the use of IFC4 or later versions is recommended. The results of the simplification process at the different LOD levels are presented in Figure 8. Although the geometric level of detail varies, the product data associated with the objects remains unchanged across all representations. As shown, some unexpected behaviour can be observed in the colouring of elements.

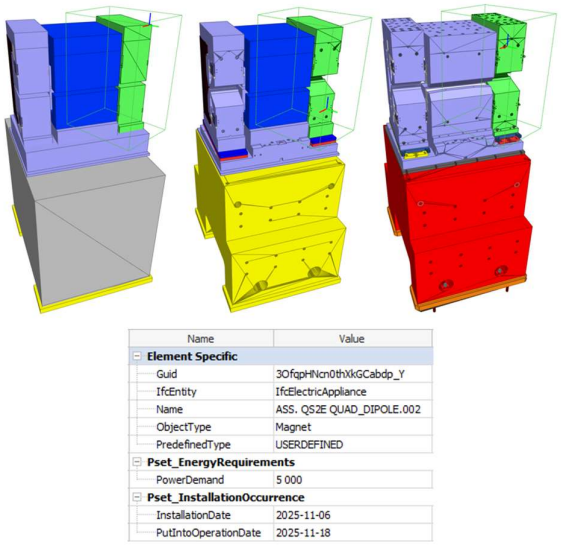

Figure 8. Resulting IFC models viewed in BIMvision [16] with different levels of detail: low (left), medium (center), and high (right). The same magnet selected has the same properties (bottom) in all models

5 Discussion

At the beginning of the present work, the primary objective was to overcome some of the limitations identified in the previous framework [4]. One of the main limitations of that version was the excessive number of rows required to be filled in the spreadsheet. Using the same case study files, the current version of the framework reduced the number of rows to complete by 86% for CS1, 74% for CS2, 81% for CS3, and 82% for CS4, resulting in an average reduction of 81% across all case studies.

However, the most relevant results emerge from the comparison with the original STEP file, which enables a meaningful evaluation of the current status of the tool against other solutions. The first metric of interest is the file size relative to the original STEP model. As shown at Table 1, IFC models generated with a high Level of Detail (LOD) occupy more storage space than the original STEP files. It is important to note that, even at this high LOD, IFC models were processed by removing small elements, as described in Section 4.2.3. This indicates that the same tessellated geometries encoded in IFC require more storage space than when represented in the STEP format. This observation further enhances the significance of the results obtained with lower LODs, which demonstrate encouraging compression ratios and confirm the effectiveness of the proposed approach in reducing model size while maintaining an acceptable level of detail.

Table 1 File size comparison for different case studies, including mean compression percentage

	CS1 [MB]	CS2 [MB]	CS3 [MB]	CS4 [MB]	Mean of %
STEP	12.0	39.8	7.9	314.4	100.0%
IFC high LOD	45.0	140.7	10.9	531.6	258.7 %
IFC medium LOD	4.0	16.5	3.8	147.1	42.4 %
IFC low LOD	0.1	0.8	0.1	0.3	0.9 %

This issue is not solely related to file size. Several systems can handle large IFC files without significant performance loss, making geometric simplification unnecessary. The key factor is instead selecting only the required objects. Some tools convert all STEP entities into *IfcBuildingElementProxy* or *IfcElementAssembly* objects. While this enables geometry import into BIM or civil environments, it is ineffective for product data management. Large numbers of objects without associated product information add complexity while providing only generic classifications. Without linked data, the benefits of the IFC standard are largely lost, reducing the model to a collection of heavy 3D geometries. Table 2 shows the number of objects removed or merged during conversion. The resulting compression rate is encouraging and confirms that the proposed framework effectively adapts an industrial model for civil engineering applications.

Table 2 Object count across process phases for different case studies, including mean compression percentage

	CS1	CS2	CS3	CS4	Mean of %
Initial stage	1633	3729	501	7632	100.0%
Delete small objects [<0.05m]	839	1241	213	2148	38.8%
Simplify geometries	10	325	6	52	2.8%
Convert IFC meshes	6	256	4	43	2.2%

Another key efficiency metric is the workload for both the user and the computational process, as shown in Figure 9. The main factor affecting the total duration is model complexity, measured by the number of objects. On average, processing rates are about 0.58 elements per second for low LOD, 0.63 for medium LOD, and 0.72 for high LOD. As the number of objects increases, both user effort and processing time grow, regardless of the selected LOD. Performance differences between LOD levels mainly reflect the computational cost of handling geometries of varying complexity, and this gap becomes more evident as the number of objects increases.

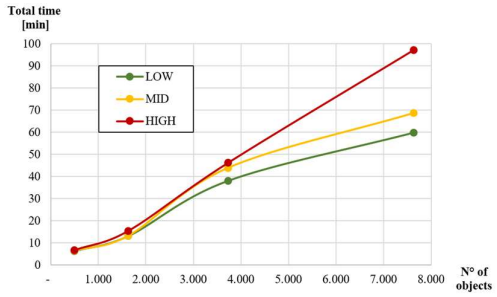

Figure 9. Procedure duration (minutes) relative to the number of objects contained in the STEP model.

5.1 Limitations

A key limitation of the framework is that, despite being designed for general professionals, it remains cumbersome and requires targeted training for effective use. Users must also possess knowledge of the IFC standard to correctly assign properties. Errors can occur during ODS file compilation (e.g., typographical mistakes), highlighting the need for more error-resistant mechanisms such as dropdown menus or predefined inputs. Additionally, manual file transfers introduce further risk of user error.

Another limitation lies in the reliance on STEP files with a specific decomposition logic, assuming a well-structured hierarchy of products and parts. In practice, such files may contain inconsistencies, merged components, or unexpected configurations. Therefore, broader testing on real-world cases is needed to assess robustness and identify improvements for professional use.

5.2 Future works

Future work will address the identified limitations through more extensive testing, providing clearer insights into application time and performance. Enhancements are planned to reduce manual effort, including features to automatically populate the *Node for IFC conversion* property using existing ODS database entries, enabling recognition of previously defined components and streamlining the workflow.

The framework will also be extended to support additional IFC classes, such as *IfcDistributionElement* and *IfcPort*, and to incorporate system management properties. Furthermore, it will be expanded to handle material data that require dedicated IFC classes rather than simple attributes or property sets.

The current need for expertise in IFC and Blender limits scalability and adoption. To mitigate this, a more

user-friendly solution should be developed to assist in selecting IFC classes and properties while reducing direct interaction with Blender. A potential approach is a web-based platform that runs Blender processes in the background, simplifying the workflow and improving accessibility.

6 Conclusion

This work examines the challenges of integrating industrial and civil domains through open standards such as STEP and IFC. Despite relying on mature data models, their differing paradigms and information requirements hinder interoperability, particularly when embedding industrial machinery within building models. The primary bottleneck does not lie in simplifying BIM data for industrial use, but rather in enriching industrial CAD models to satisfy the semantic, relational, and contextual requirements of the BIM environment. To address this gap, the study introduces a lightweight, open-source framework that goes beyond mere geometric translation. The framework demonstrates the feasibility of aligning industrial product logic with AEC information structures. The entire procedure is based on open-source technologies and is implemented as a Blender add-on using the Bonsai extension. Its intuitive user interface makes the workflow accessible even to users without specialized programming skills.

The procedure presented in this paper focuses on reducing the assembly complexity of industrial models by removing unnecessary components, simplifying excessively detailed geometries, reducing the number of objects to those strictly required, and attaching the appropriate semantic properties to them. In addition, mechanisms for storing and reusing manually inserted data are incorporated to minimize repetitive tasks. These strategies are intended to make the framework suitable for professional use by reducing manual effort and saving time during the conversion process. Preliminary results indicate the effectiveness of the proposed approach in adapting industrial product logic to AEC requirements.

7 Acknowledgements

The authors would like to thank Paul Scherrer Institut (PSI) for their technical support, for providing the materials used in this case study, and for granting permission to publish selected images of mechanical components from the Swiss Light Source (SLS) project.

References

1. Xia, B.; Chan, A.P. Measuring Complexity for Building Projects: A Delphi Study. *Eng. Constr. Archit. Manag.* **2012**, *19*, 7–24.
2. *ISO 10303-21*; **2016**.
3. *ISO 16739-1*; **2024**.
4. Avogaro, D.; Zanchetta, C. Bridging Information from Manufacturing to the AEC Domain: The Development of a Conversion Framework from STEP to IFC. *Systems* **2025**, *13*, 421.
5. Farinha, F.; Gonçalves, R.J.; Garção, A.S. Integration of Cooperative Production and Distributed Design in AEC. *Adv. Eng. Softw.* **2007**, *38*, 772–779, doi:https://doi.org/10.1016/j.advengsoft.2006.08.038.
6. Gielingh, W. An Assessment of the Current State of Product Data Technologies. *Comput.-Aided Des.* **2008**, *40*, 750–759, doi:https://doi.org/10.1016/j.cad.2008.06.003.
7. Yang, D.; Eastman, C.M. A Rule-Based Subset Generation Method for Product Data Models. *Comput.-Aided Civ. Infrastruct. Eng.* **2007**, *22*, 133–148, doi:https://doi.org/10.1111/j.1467-8667.2006.00476.x.
8. Yang, Q.Z.; Zhang, Y. Semantic Interoperability in Building Design: Methods and Tools. *Comput.-Aided Des.* **2006**, *38*, 1099–1112, doi:https://doi.org/10.1016/j.cad.2006.06.003.
9. Nzetchou, S.; Durupt, A.; Remy, S.; Eynard, B. Semantic Enrichment Approach for Low-Level CAD Models Managed in PLM Context: Literature Review and Research Prospect. *Comput. Ind.* **2022**, *135*, 103575, doi:https://doi.org/10.1016/j.compind.2021.103575.
10. © The FreeCAD Team FreeCAD Available online: https://www.freecad.org/.
11. Blender Foundation Blender Available online: https://www.blender.org/ (accessed on 18 September 2025).
12. IfcOpenShell Bonsai Available online: https://bonsaibim.org/index.html (accessed on 18 September 2025).
13. Delorme, H. Mayo Available online: https://github.com/fougue/mayo (accessed on 9 April 2025).
14. Paul Scherrer Institute PSI PSI - Large Research Facilities Available online: https://www.psi.ch/en/research/large-research-facilities (accessed on 14 January 2026).
15. Avogaro, D. STEP-to-IFC Available online: https://github.com/Daavogaro/STEP-to-IFC (accessed on 14 January 2026).
16. BIMvision Available online: https://bimvision.eu/download/.