2026 European Conference on Computing in Construction
Corfu, Greece
July 12–15, 2026

GRAPH-BASED INSTANCE TRACEABILITY FOR BUILDING PRODUCT REUSE

Silvia Vangelova[1], Clara P. Blum[1], Kasimir Forth[1], Jens J. Hunhevicz[1,2], Pritha Jacob[1] and Catherine De Wolf[1]
[1]ETH Zurich, Zurich, Switzerland
[2]Empa, Dübendorf, Switzerland

Abstract

Tracing information related to the life cycle of building products supports decision-making for circular construction. However, current digital tools and standards in construction primarily address product identification at the type rather than instance level, resulting in information loss that hinders reuse. To trace products across projects at the instance level, we propose a graph-based workflow that maps globally unique identifiers (GUIDs) of element instances. While information continuity can be established manually, preliminary workflow evaluation in a case study reveals that manual GUID verification is impractical. We conclude that realising a digital reuse supply chain requires software automation for instance-level identification.

Introduction

Reuse in construction, one of the most resource-intensive industries, plays a key role in achieving a sustainable built environment (Lima et al., 2021; Eurostat, 2024). Reuse, in general, can be understood as a process that enables multiple use cycles of products (Alcayaga et al., 2019). Capturing and updating product life-cycle information at each use cycle and storing it throughout the product's lifetime is an important capability for enabling product reuse (Wan and Jiang, 2025). However, for construction products, this capability remains a challenge due to the long, multi-phased life cycle of buildings. Novel information management solutions are required to ensure long-term data availability and maintainability (Hunhevicz and De Wolf, 2023).

Current element-based information management solutions for building product reuse build on existing digital tools but remain only prototypical. To achieve instance-level traceability for reuse, potential solutions extend these tools by integrating globally recognised supply chain standards (Hradil et al., 2023). Specifically, the GS1 framework enables product instance identification using Serialised Global Trade Item Numbers (SGTIN) (GS1, 2024). By assigning a unique SGTIN to a physical product, this system distinguishes individual instances from otherwise identical products of the same type. Yet, maintaining instance-level traceability over a product's lifecycle remains challenging; critical data disconnects occur between phases even within a single project (Borrmann et al., 2018), let alone across multiple projects separated by years or decades.

In reuse scenarios, preserving instance-level traceability can be crucial, as secondary products often require individual structural validation or condition assessment. Type-level identification can lead to the loss of instance-specific information, such as material condition or load history, that often differs across reused elements of the same type (Blum et al., 2026). Although identification at the instance level is technically more challenging and often abandoned in favour of simpler type-level tracking, it remains valuable for reuse decisions (Byers and De Wolf, 2023).

While linking type- and instance-based identifiers via semantic models can help mitigate this information loss (Kebede et al., 2022), traditional building information modelling (BIM) software lacks the built-in functionality to implement such semantic models (Pauwels et al., 2017). This disconnect is compounded because BIM tools are primarily designed for construction using new products (Pocobelli et al., 2018), abstracting elements into replicable object-oriented types rather than tracing unique physical instances. By referencing a single parent definition to instantiate multiple elements of the same type (Costin et al., 2022), these applications prioritise computational efficiency at the expense of historical data continuity. Consequently, incorporating reused products into a design results in the instantiation of new elements with new application-generated identifiers, which can effectively overwrite their previous instance-level identity. Consequently, traditional software offers limited support for instance-level identification for product reuse.

Despite limited cross-project instance traceability, BIM authoring tools handle type-to-instance mapping within single projects very well by using relational data schemas internally, which can be translated into Industry Foundation Classes (IFC) for interoperable data exchange. The problem is that these data structures are enclosed within single project files, and there is no established mechanism to query highly granular, instance-level information across separate models. To overcome these native software limitations and explicitly define semantic relationships, BIM models can be linked to a graph in which elements are represented as nodes and their spatial relationships as edges. This approach decouples information from BIM models by enabling storage in graph databases (Cao et al., 2022) and can optimise life-cycle data management by linking project specifications to external product databases (Forth and De Wolf, 2025), facilitating disassembly sequencing (Sanchez et al., 2021) and tracing incremental changes

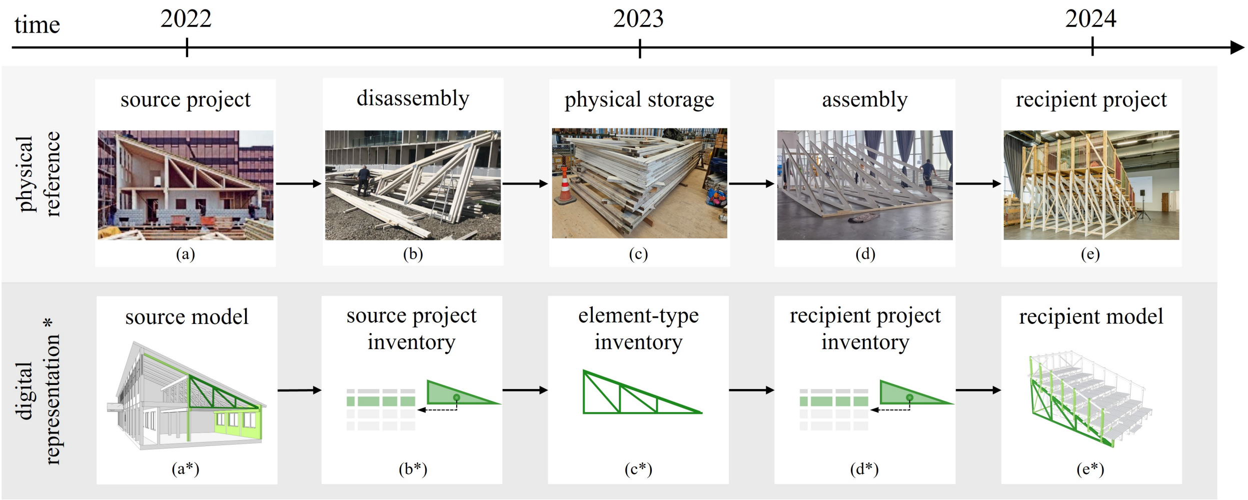

Figure 1: Physical and digital mapping of element instances across projects exemplified by the reuse of the Huber Pavilions. (*) Alignment in time is only approximate. Photo and model credit is provided in the Acknowledgements section

across project phases (Esser et al., 2023).

To address the gap in instance-level element identification, this study proposes a graph-based workflow for tracing product instances across different use cycles. The proposed workflow uses element-based graphs to map globally unique identifiers (GUIDs) from a source project to a recipient project, thereby preserving instance-level identification that is often lost during software exchanges. Following preliminary validation through a case study and a custom visualisation tool, we argue that establishing a digital reuse supply chain requires automated instance mapping that extends the capabilities of current design tools.

Point of Departure

Bridging physical and digital traceability

Reusing elements across multiple projects is a key strategy for enabling a circular economy in construction. However, the information needed to reuse an element is often lost at project handover or becomes outdated during operation and must be reacquired at the end of a building's life cycle (Honic et al., 2021). This acquisition process typically involves a building audit, during which a digital inventory of its constituent elements is created (Honic et al., 2019). To manage this inventory, identifiers such as quick response (QR) codes or near-field communication (NFC) chips can serve as a ''phygital'' link connecting a physical building element to its digital representation (Byers et al., 2025). Unlike the consumer goods industry, where traceability is optimised through linear distribution networks and statistically predictable behaviour, typically without the need for a digital model, construction operates on a project-by-project basis at a higher level of complexity. It involves longer time spans and various stakeholder groups that transfer multidisciplinary data between project phases. Consequently, construction requires coordinating products across two parallel timelines: the physical supply chain and the digital project-related information supply chain (Byers et al., 2025). This interdependency between the physical and digital sides of the construction supply chain makes it challenging to maintain consistent identifiers across both dimensions during reuse.

Type vs. instance traceability

Creating a building model for a project requires distinguishing between element types, which are defined by the shared properties of a product, and element instances, which identify each unique physical occurrence of that product and its specific positioning within the building. This distinction is well established in BIM, where buildings and products are represented in an object-oriented structure with classes and properties organised in a nested hierarchy. Specifically, the IFC schema models type-to-instance relationships in an object-oriented manner by separating universal specifications of a given product (IfcTypeObject) from the unique physical occurrences of this product (IfcObject) within a project (buildingSMART International, 2016). The IfcRelDefinesByType relationship maps each instance to its type, allowing element instances to inherit type-level properties from a single element type definition.

The IFC schema provides standardised property set definitions, such as the Tag property of IfcElement (buildingSMART, 2020), to distinguish specific element instances using serial numbers or application identifiers, such as GUIDs. The IFC Tag was originally intended for standardised global identifiers, such as those defined by GS1. This tag is often used to store element GUIDs, unique codes generated automatically by the authoring software to identify element instances. While these GUIDs are useful for tracking product changes, they are often unreliable for tracing at the instance level across different project phases because BIM authoring workflows often lead to generating inconsistent element identifiers (Esser et al., 2022).

The challenge of instance-level traceability

While there have been developments towards integrating GS1 identifiers into the buildingSMART Data Dictionary (bSDD) for IFC (GS1, 2025), current implementations for tracing across projects are more feasible for product types than for individual instances. One solution for tracing element information at the instance level is provided by Speckle. Speckle uses application IDs, the equivalent of element GUIDs, to maintain a link between an element and its authoring software, and hierarchical graphs to store project information. In these tree-like structures, elements contain other elements and references to elements stored outside the authoring application ecosystem. A suitable way to record these relationships is to store GUID key value pairs in a graph database. In the case of reusing elements from a source project in a recipient project, this requires mapping the original instance GUID from the source project to a product type ID (instance-to-type), and subsequently linking that type ID to the new instance-level GUIDs generated in the recipient project (type-to-instance). However, this process of instance and type mapping is where instance-level traceability breaks down due to information loss when models are transferred across different authoring software (Blum et al., 2026).

Scope and methodology of this study

Given the need to consistently trace product instances of the same type, along with their digital representations, across multiple use cycles in different projects, this study aims to develop a workflow for tracing the history of life-cycle-related information of reused building products across projects.
To answer the research question of how reused building elements of the same type can be digitally represented and identified at the instance level across projects, we analysed a reuse project at ETH Zurich to trace the flows of type- and instance-level information. Based on this analysis, we identified a cycle of instance-to-type mapping that repeats throughout the reuse process. This study therefore suggests that the digital side of the reuse supply chain should follow a cyclic pattern of recurring steps traversed using a graph-based workflow. Following this workflow, we propose that element GUIDs be mapped from source to recipient projects using a graph-based structure that enables the tracing of instance-level changes across product use cycles in line with the recurring steps.

Analysing Instance-to-Type Information Flow

Case study: the Huber Pavilions and Mehrwerk

As a case study, we trace the digital representation of several timber frames of the same type sourced from the deconstructed Huber Pavilions at the ETH Zurich Hönggerberg campus and their subsequent reintegration in Mehrwerk, an installation at the Kunsthalle Zürich (De Wolf, 2025). Specifically, we trace this single element type from its original function in supporting the pavilions' roof structure to its new function in supporting the seating platform of Mehrwerk, demonstrating how instance-level identification can be maintained throughout this transition.

We analyse the case study in five steps, tracing the physical reuse supply chain alongside its parallel digital representation (see Figure 1). Tracing begins with the source project (a), one of the Huber Pavilions (Knecht, 2022), for which a BIM model (a*) was created based on an audit conducted prior to disassembly. During the disassembly (b) of the building, a digital inventory of its constituent elements (b*) was created. The disassembled elements were then relocated to a physical storage location (c), where elements of the same type were physically grouped and represented digitally as a consolidated element-type inventory (c*). During the assembly (d) of the recipient project, Mehrwerk, a new inventory (d*) was created that specified the required elements. After completion of the recipient project (e), its final as-built model (e*) was created.

Identified instance-to-type mapping cycle for reuse

We observed a cycle of mapping instance and type relationships of building elements. We use the term ''cycle'' to encapsulate the pattern of recurring events in product use cycles, distinguishing these events from the physical ''circular'' flow of resources. This terminology emphasises that in reuse scenarios, a building project has one life cycle, whereas building products have multiple use cycles across different projects.

On the sourcing side, buildings at their end-of-life phase typically lack digital documentation. To facilitate reuse and environmental assessments, a digital model is created through a building audit. In this model, the geometry of each physical element is represented as a unique instance with a GUID automatically assigned by the software used to create the model. As the building is deconstructed, these elements lose their original context. When the elements are relocated to a storage facility, individual instance data is condensed into a digital inventory of elements grouped by type. In the recipient project, the steps for element GUID association are repeated by reusing the same elements in a new context. When a designer selects an element type from the inventory, they instantiate it within a new project model. This action generates new instance identifiers, effectively mirroring the initial flow of information by transitioning from the consolidated type-based inventory to element instances in the recipient project.

While current digital tools are built for project life cycles, e.g., managing a single building across specific project phases, they are not built for product use cycles. One example is managing a beam that is sourced from the end-of-life phase of one project to the initial design phase of another. Currently, there is no functionality that allows product information to be retained after the project's end-of-life phase for future use cycles. This is especially important during the early design phases, where implementing circular solutions has the greatest impact (Heisel and McGranahan, 2024).

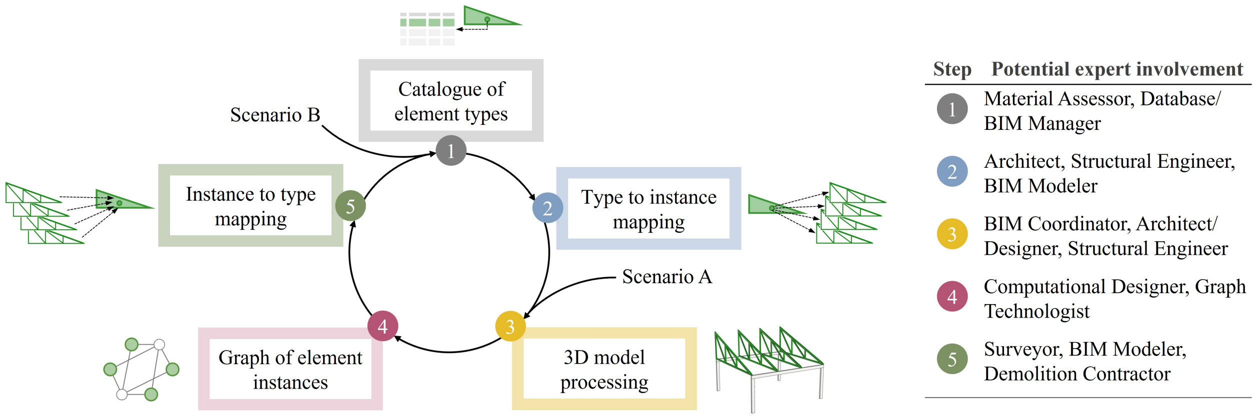

Figure 2: Graph-based workflow for product life-cycle information tracing. Potential expert involvement for each step is proposed. Scenario A starts with an inventory of reused products. Scenario B starts with the reuse of a building constructed with new products.

Workflow for Life-Cycle Information Tracing

In this section, we describe our proposed workflow for tracing information about multiple instances of a single element type in five steps, as illustrated in Figure 2.

In this workflow, we use four types of element instance and type representations (see Figure 3). Type a is a three-dimensional geometric representation, which can be a BIM or CAD model within the authoring software or an IFC export of it. Type b is the graph representation of the model, capturing the connectivities between different elements, where each node represents an element instance, and each edge is a spatial connection between two elements. Type c is an instance-to-type mapping, a key value-pair dictionary, where the keys are the element type IDs and the values are the instance GUIDs. Type d is a tabular dataset, which we refer to as a ''catalogue'', and can be either a digital inventory of physically stored elements or a specification table of the products to be used in a new design.

To better illustrate the graph-based workflow, we present two scenarios for initiating the reuse cycle. The first scenario (A) is when a building has no catalogue of the products used, it is built with new products, and is inventoried through disassembly for reuse. The second scenario (B) is designing using a given set of element types from an inventory of reused building products.

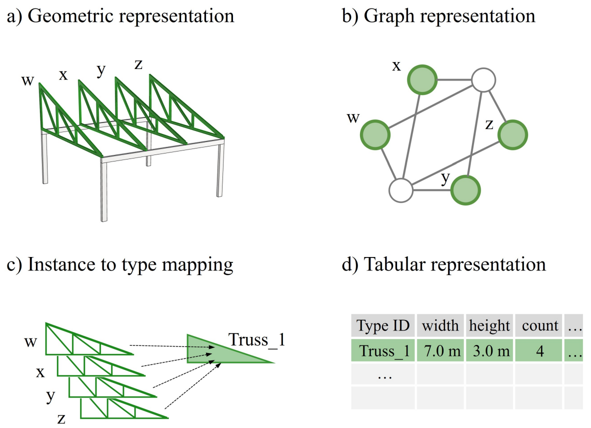

Figure 3: Element instance and type representations used in the graph-based workflow

In the following subsections, we describe our workflow based on scenario B, which consecutively loops through all steps in the circular loop (see Figure 2). The workflow begins with a catalogue of element types (1), which may include both reused and new products. During the design phase, these types are instantiated, establishing a type-to-instance mapping (2) where unique identifiers (GUIDs) are generated for the geometry of each element instance.

Once the design is finalised, 3D model processing (3) standardises the data via IFC export. This geometric data is then used to compute a graph of element instances (4), storing their geometric connectivity, with elements represented as nodes in the graph and linked via their GUIDs. Finally, the workflow concludes during the disassembly or end-of-use phase with instance-to-type mapping (5). In this step, specific element instances are remapped to new element types to repopulate an inventory for future projects, effectively starting a new use cycle.

Alternatively, for scenario A, the cycle can start with the model-processing step (3), as shown in Figure 2. This is how we applied the workflow to the case study since we do not have the original project specifications that catalogue all products used in the building. Additionally, the source building was constructed with new products rather than an inventory of reused ones.

Step 1: Catalogue of element types

Whether creating a model for new construction or reuse projects, the workflow starts with a catalogue of element types. In projects that reuse building elements, this catalogue is usually derived from an inventory, a finite set of reusable elements stored in a physical location, most of which come from a previous building. In new construction projects, this catalogue functions similarly to a bill of materials, specifying the product types that will be assigned to the element instances in the model. In this study, we define the catalogue of element types as a unified tabular

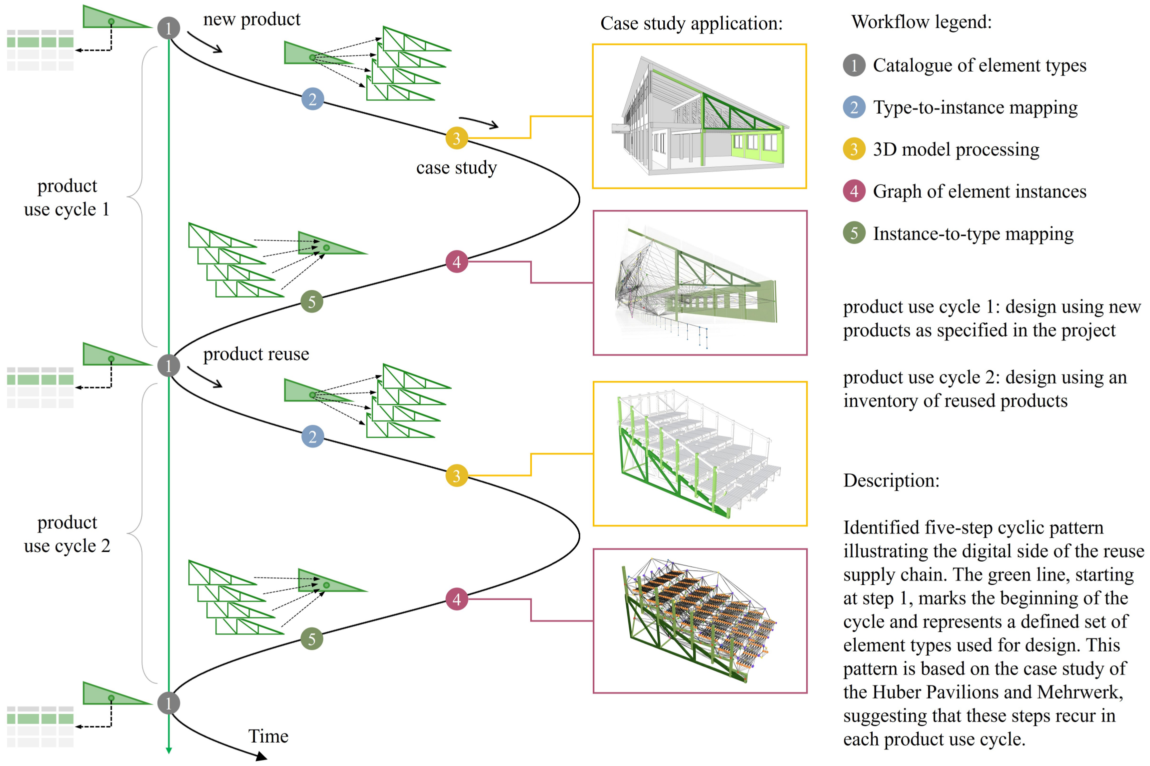

Figure 4: Identified instance-to-type mapping cycle for reuse and its application in the case study

dataset, regardless of whether they are reusable elements or new products. Defined by attributes such as type ID, dimensions, and available quantity, this catalogue serves as a predefined set of element types and is used in the next step of the workflow, where elements are instantiated in the authoring software.

Step 2: Type-to-instance mapping

Type-to-instance mapping occurs implicitly as geometry is created within the authoring software during the design process. The designer assigns an element type ID, which is stored as both a column in the inventory data table and a specific property of the element instance in the model. This linkage is critical for the subsequent model processing in Step 3 because it maintains traceability between the element type and the instance GUID generated by the authoring software in later steps.

Step 3: 3D model processing

To extract and retain relevant life-cycle-related information, we use two distinct processes, depending on the authoring software type of the source file, which are applicable for both scenario A and B. Both processes are based on the IFC schema because it is an open data exchange format and supports product identification through geometric representation and unique identifiers, namely, the tag property of IFC building elements. We differentiate between the two processing workflows:

1. BIM-based processing (e.g., Revit, Archicad): We export these models from native authoring software directly into IFC4, ensuring base quantities and geometry are included in the export configuration.
2. CAD-based processing (e.g., Rhino, AutoCAD): We first ensure the geometry of element instances is assigned to corresponding element-type layers. The model is exported in STEP format, attaching layer information to the geometric representation of each element. We then use IfcOpenShell to convert the STEP file to IFC4, and pythonOCC, a Python wrapper of OpenCASCADE, to generate the geometric representation and compute base quantities.

Step 4: Graph of element instances

Using the IFC file generated in the previous step, we compute a geometric connectivity graph to store how the elements are connected. For this step, we further developed an existing IFC viewer (Forth and Emenegger, 2025) to compute the graph in the back end and edit IFC properties in the front end (see Figure 5). The graph computation is performed using an additionally developed Python back end for the IFC-graph editor, where geometry is extracted using IfcOpenShell and OpenCascade. To optimise performance, we estimate the number of connected pairs based on the distance between oriented bounding boxes. For each candidate pair, we then analyse the actual geometry; if the distance between elements is less than 1 cm, then they are considered to be connected. We justify this assumption because a tolerance of 1 cm to 2 cm falls within standard allowable deviations for building construction (DIN, 2019). Using the final list of pairs, we con-

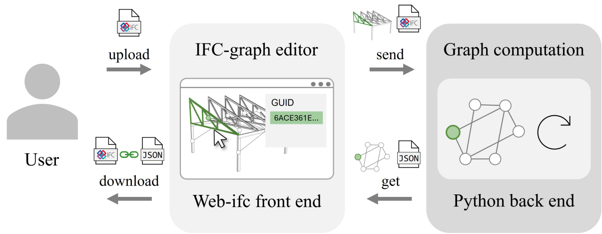

Figure 5: User interaction for assigning a GUID to an element and linking the geometric connectivity graph to the IFC file

struct a graph with NetworkX, attaching the corresponding GUID to each node and convert it to JSON.
Through this graph, for each node of an element, a list of connected elements can be retrieved. The resulting graph is stored as a NetworkX-based JSON. This approach was chosen because JSON is supported by the application programming interfaces (APIs) of commonly used BIM software, allowing the graph and GUIDs to be linked to their respective elements in the authoring software. For example, by using plugins to serialise and deserialise the JSON of the graph. To preserve the link between the graph and the IFC, we assign a unique identifier to the graph itself and store it within a custom IfcPropertySet as an ifcText.

Step 5: Instance-to-type mapping

Unlike in Step 2, where mapping occurs in the authoring application, in Step 5, we use the IFC-graph editor to update the mapping between specific instance GUIDs and type IDs. This step is necessary because the as-built model often diverges from the initial design model yet requires the original type IDs to be relinked with the final element GUIDs.
The front end of the IFC-graph editor is used to visualise the IFC files and the graph computed in Step 4. The viewer comprises a TypeScript front end based on web-ifc from That Open Company and the previously mentioned Python back end (see Figure 5). Through this interface, we manually verify and map GUIDs to type IDs for the selected elements to be reused. The application then exports an updated IFC with the correct GUIDs and type IDs and a linked JSON graph, where key value pairs of element GUIDs preserve the semantic link between the elements and their connected neighbours for the next use cycle.

Preliminary Evaluation and Future Work

For initial validation, we applied the graph-based workflow to the case study of the Huber Pavilions and Mehrwerk (see Figure 4) by tracing the reuse of timber frames from the source project (Huber Pavilions) to the recipient project (Mehrwerk). Originally, while the frames were geometrically identical, the transfer between software applications during the design of Mehrwerk resulted in a loss of instance-level traceability. Furthermore, the columns connected to the frames in the source project were modified in the recipient project, creating a distinct set of connections that required a new inventory version. Using our workflow,

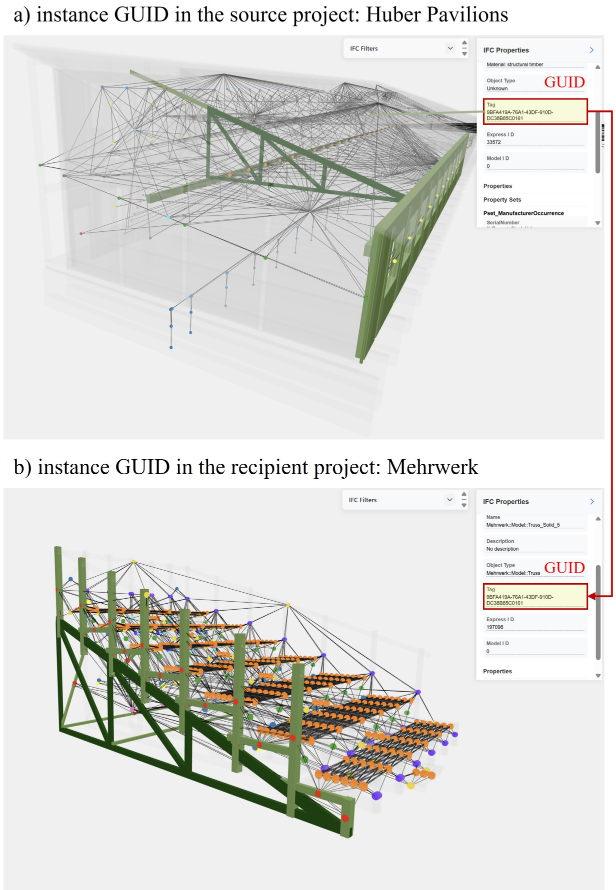

Figure 6: Manual GUID assignment of selected elements using the IFC-graph editor. This action modifies the original IFC model and links it to a graph through the assigned GUIDs.

we re-link the timber frames in both models.
The workflow begins with the source project. We exported the Huber Pavilion model from Archicad as IFC4 (Step 3). Then, using the IFC-graph editor, we computed the element connectivity graph (Step 4). We then mapped the type-to-instance relationships by assigning a specific Type ID property to the elements (Step 5). To link the frames from the source to the recipient project, we initiated the workflow by storing their element type IDs in a table (Step 1). We then assigned these types to the corresponding geometry in the Mehrwerk model (Step 2) before exporting it from Rhino and converting it to IFC4 (Step 3). Using the IFC-graph editor, we computed the graph for this new context (Step 4). Finally, we assigned the original GUIDs from the Huber model to the new frame instances (Step 5). This way, we were able to re-link the element instances across both models (see Figure 6).

Challenges and limitations

A limitation in the case study is the inability to verify the instance-level GUID assignment. While we successfully mapped element GUIDs across the source and recipient models, we retrospectively lacked ground-truth data to confirm which timber frame from the source project corresponded to which instance in the recipient project. Furthermore, in the case study, no instance-level differentiation has been implemented. This disconnect highlights a broader issue: collecting and maintaining instance-level data across the reuse supply chain is currently not realised. Even though the instance-to-type mapping solution in the IFC schema allows for the preservation of instance-level variables, manually verifying that every element has a correct GUID corresponding to a physical product requires significant effort, particularly during early design iterations, when geometry is rapidly changing. The labour of manual verification is also due to long project life cycles, during which information is lost across different software. Although GS1 standards specify the general requirements for instance-level identification through SGTINs, current construction software lacks the capability to integrate these standards efficiently. In particular, tracing building products composed of multiple sub-components or layers is challenging to integrate, as these can be disassembled and reassembled into different elements. To enable instance traceability without the overburden of manual tagging, algorithms need to be developed to automatically detect when a digital element instance in a new design contextually represents a physical element in the inventory, and maintain this link throughout further project phases. On the physical side, a potential solution to these efforts is to use extended reality for the identification of as-built elements. On the digital side, algorithms for measuring geometric similarity between elements can help identify matching pairs (Collins et al., 2024).

Regarding the IFC-graph editor developed for this study, user feasibility testing is needed to validate better graph visualisation approaches and determine whether the front-end graph visualisation actually supports instance-level identification or should be kept in the back end. Future research will focus on formalising the graph-based representation and connecting the viewer to a graph database, thereby enabling the integration of dynamic digital passports that evolve throughout product use cycles.

Conclusion

In this study, we developed a graph-based workflow for tracing product use cycles across projects. We demonstrated the workflow using an IFC-graph editor to map element GUIDs across two projects, using the Huber Pavilions and Mehrwerk as a case study. The workflow shows that preserving element identification outside the original authoring software is a viable strategy to reduce vendor lock-in and facilitate product data transfer across use cycles. However, data provenance remains to be addressed, as individual product data can be isolated from project databases with varying levels of access.

Making instance mapping more efficient by linking digital passports to graph databases of building projects will facilitate reuse. If realised, this functionality would serve two purposes: it would allow building owners and reuse partners to trace the life-cycle history of building products effectively while simultaneously supporting designers to integrate reused products into new designs. Applying the learnings from the case study of the Huber Pavilions and Mehrwerk to active projects will be the next step in advancing the graph-based reuse workflow.

Acknowledgments

This research has been funded by the Flagship projects SWIRCULAR (107.235 FS-EE) and ThinkEarth (107.163 FS-ENG), both supported by Innosuisse. The authors would like to thank Vanessa Schwarzkopf, Elias Knecht and Heidi Silvennoinen for providing the models and on-site photos of Mehrwerk and the Huber Pavilions. In addition, the authors gratefully acknowledge Dr. Roxanne Goldberg for valuable editorial support.
Credits in Figure 1:
(a) Photo by gta Archiv / ETH Zurich (Estate of Benedikt Huber);
(b) Photo by Elias Knecht, Master student, ETH Zurich;
(d) Photo by Vanessa Schwarzkopf, CEA, ETH Zurich;
(e) Photo by Anna Buser, CEA, ETH Zurich;
(a*) 3D model by Heidi Silvennoinen, CEA, ETH Zurich;
(e*) 3D model by Vanessa Schwarzkoph, Vincent Jackow, Tim Cousin, CEA, ETH Zurich;

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this paper, the authors used Google Gemini 3 Pro to enhance the clarity of the text. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

Alcayaga, A., Wiener, M., and Hansen, E. G. (2019). Towards a framework of smart-circular systems: An integrative literature review. Journal of Cleaner Production, 221:622–634.

Blum, C., Vangelova, S., Hunhevicz, J., Forth, K., and De Wolf, C. (2026). Breakpoints in data continuity across construction reuse projects.

Borrmann, A., König, M., Koch, C., and Beetz, J. (2018). Building Information Modeling: Why? What? How? In Borrmann, A., König, M., Koch, C., and Beetz, J., editors, Building Information Modeling: Technology Foundations and Industry Practice, pages 1–24. Springer International Publishing, Cham.

buildingSMART (2020). IfcElement attribute definitions. https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2/HTML/link/ifcelement.htm.

buildingSMART International (2016). IfcRelDefinesByType. https://standards.buildingsmart.org/IFC/RE-

LEASE/IFC4/ADD2/HTML/link/ifcreldefinesby-type.htm.

Byers, B. S. and De Wolf, C. (2023). QR Code-Based Material Passports for Component Reuse Across Life Cycle Stages in Small-Scale Construction. Circular Economy, 1(2):1–16.

Byers, B. S., Stengele, K., and De Wolf, C. (2025). Data carriers for circular construction supply chains: An exploratory quantitative analysis. Journal of Cleaner Production, 494:145053.

Cao, J., Bucher, D. F., Hall, D. M., and Eggers, M. (2022). A graph-based approach for module library development in industrialized construction. Computers in Industry, 139:103659.

Collins, F. C., Braun, A., and Borrmann, A. (2024). Finding Geometric and Topological Similarities in Building Elements for Large-Scale Pose Updates in Scan-vs-BIM. In Skatulla, S. and Beushausen, H., editors, Advances in Information Technology in Civil and Building Engineering, pages 517–530, Cham. Springer International Publishing.

Costin, A., Ouellette, J. W., and Beetz, J. (2022). Building product models, terminologies, and object type libraries. In Pauwels, P. and McGlinn, K., editors, Buildings and Semantics: Data Models and Web Technologies for the Built Environment, pages 3–24. CRC Press, London.

De Wolf, C. (2025). The Art of Connecting: Reuse of the Huber Pavilions. Edition Detail, Munich.

DIN (2019). Tolerances in building construction - Buildings. Standard DIN 18202, Deutsches Institut für Normung.

Esser, S., Vilgertshofer, S., and Borrmann, A. (2022). Graph-based version control for asynchronous BIM collaboration. Advanced Engineering Informatics, 53:101664.

Esser, S., Vilgertshofer, S., and Borrmann, A. (2023). Version control for asynchronous BIM collaboration: Model merging through graph analysis and transformation. Automation in Construction, 155:105063.

Eurostat (2024). Material footprints - details by final use of products. https://doi.org/10.2908/ENV_AC_RMEFD.

Forth, K. and De Wolf, C. (2025). Formalizing Information for Disassembly Potenttial of Buildings Using BIM and Labeled Property Graphs. volume 6. European Council on Computing in Construction.

Forth, K. and Emenegger, P. (2025). IFC and Neo4j Disassembly Graph Viewer. https://github.com/cea-ethz/disassembly-graph-viewer.

GS1 (2024). GS1 Standards enabling the Digital Product Passport. Whitepaper, GS1. https://gs1.eu/wp-content/uploads/2024/12/GS1-Standards-Enabling-DPP.pdf.

GS1 (2025). bSDD GS1 Identifiers. https://identifier.buildingsmart.org/uri/gs1/gs1/0.1.0.

Heisel, F. and McGranahan, J. (2024). Enabling Design for Circularity with Computational Tools. In De Wolf, C., Çetin, S., and Bocken, N. M. P., editors, A Circular Built Environment in the Digital Age, pages 97–110. Springer International Publishing, Cham.

Honic, M., Kovacic, I., Aschenbrenner, P., and Ragossnig, A. (2021). Material Passports for the end-of-life stage of buildings: Challenges and potentials. Journal of Cleaner Production, 319:128702.

Honic, M., Kovacic, I., and Rechberger, H. (2019). Concept for a BIM-based Material Passport for buildings. IOP Conference Series: Earth and Environmental Science, 225(1):012073.

Hradil, P., Jaakkola, K., and Tuominen, K. (2023). RFID-based traceability system for constructional steel reuse. In Life-Cycle of Structures and Infrastructure Systems. CRC Press. Num Pages: 8.

Hunhevicz, J. and De Wolf, C. (2023). Blockchain for a Circular Digital Built Environment. In IoT, Blockchain, and Smart Environments. Design Computation.

Kebede, R., Moscati, A., Tan, H., and Johansson, P. (2022). Integration of manufacturers' product data in BIM platforms using semantic web technologies. Automation in Construction, 144:104630.

Knecht, E. (2022). Future Learning Spaces - Re-Used.

Lima, L., Trindade, E., Alencar, L., Alencar, M., and Silva, L. (2021). Sustainability in the construction industry: A systematic review of the literature. Journal of Cleaner Production, 289:125730.

Pauwels, P., Zhang, S., and Lee, Y.-C. (2017). Semantic web technologies in AEC industry: A literature overview. Automation in Construction, 73:145–165.

Pocobelli, D. P., Boehm, J., Bryan, P., Still, J., and Grau-Bové, J. (2018). BIM for heritage science: a review. Heritage Science, 6(1):30.

Sanchez, B., Rausch, C., Haas, C., and Hartmann, T. (2021). A framework for BIM-based disassembly models to support reuse of building components. Resources, Conservation and Recycling, 175:105825.

Wan, P. K. F. and Jiang, S. (2025). Enabling a dynamic information flow in digital product passports during product use phase: A literature review and proposed framework. Sustainable Production and Consumption, 54:362–374.