A graph-based approach for module library development in industrialized construction

Journal Article

Author(s):
Cao, Jianpeng; Bucher, David; Hall, Daniel; Eggers, Mike

Publication date:
2022-08

Permanent link:
https://doi.org/10.3929/ethz-b-000541809

Originally published in:
Computers in industry 139, https://doi.org/10.1016/j.compind.2022.103659

Funding acknowledgement:
197294 - 7DayHouse: Fabrication-aware Design (SNF)

Computers in Industry 139 (2022) 103659

Contents lists available at ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A graph-based approach for module library development in industrialized construction

Jianpeng Cao[a,*], David F. Bucher[a], Daniel M. Hall[a], Mike Eggers[b]

[a] Institute of Construction and Infrastructure Management, ETH Zurich, Stefano-Franscini-Platz 5, Zurich 8093, Switzerland
[b] Project Frog, 99 Green St, San Francisco, CA 94111, United States

A R T I C L E I N F O

Article history:
Received 26 October 2021
Received in revised form 3 March 2022
Accepted 15 March 2022
Available online 29 March 2022

Keywords:
Module library
Graph database
Similarity analysis
BIM
Industrialized construction

A B S T R A C T

A library of prefabricated parts and assemblies, i.e. module library, can help a firm in the construction industry transition to a more industrialized and product-oriented approach. However, existing approaches to manage such libraries are oriented around single-use projects. There is need for a more flexible data structure to support storage, analysis and reuse of design information. This paper proposes a graph-based approach to develop a module library. The approach includes a graph representation of modules, a graph database development, and a graph-based similarity analysis. The proposed approach is validated using a prefabricated timber panel system via a web-based application. Implementation demonstrates a more efficient process for bill of material generation and identifying the impacts of design changes.

1. Introduction

Industrialized construction is experiencing a new wave of attention and investment. In a 2019 McKinsey report, the construction industry could deliver a $20 billion annual savings and 50% time saving if industrialized construction is adopted (Bertram et al., 2019). A crucial factor for achieving this goal is DfMA (Design for Manufacturing and Assembly). DfMA is originated from advanced manufacturing, and is used as the basis in concurrent engineering for cost-effective design. The main idea of DfMA is to consider the manufacturing and assembly issues in the design phase. DfMA saves design time and cost through the use of standard common parts, the reuse of previously designed details and the application of modular design (Anderson, 2003). Industrialized construction adapts this strategy of DfMA by developing standard types of prefabricated building elements (e.g., panel, component and/or volumetric modules) and then providing these modules to architects who can then fulfill design intentions. The strategy is particularly beneficial for a product structure that has a high degree of repeatability and standardization (Bertram et al., 2019), such as hotels, affordable housing, schools, hospitals and so forth.

However, it is not always easy for the construction industry to shift to this new paradigm. The industrialized construction projects sometimes suffer from the negative aesthetic perception of "modular", "box-like", or "cookie-cutter" architecture. As a result, designers might not be willing to accept this process due to limited design flexibility. Other challenges such as increased capital cost at the early adoption stage and lack of manufacturing knowledge for designers also restrict the DfMA application in actual projects (Lu et al., 2020; Gao et al., 2020). To mitigate these challenges, it is critical to provide designers with manufacturing input during their design.

Today, the most common approach to product libraries is to provide design flexibility in various configurations by selecting and recombining well-designed and engineered building elements from product libraries (Cui et al., 2020). Existing construction product libraries e.g. NBS library, Open Source Wood, etc, enable designers to find similar designs and templates on which to copy or modify. However, those libraries contain few prefabricated components, or modules, suitable for industrialized construction and do not provide easy access for designers to retrieve desired content (Li et al., 2020). Some new industrialized construction firms, such as DMD Modular, Project Frog, CIMC MBS, etc., start to develop their own module libraries as their core products, or product catalog. The module library of industrialized construction companies almost falls into three categories: volumetric modular, flat-pack and kit-of-parts, according to their business strategies (Pullen et al., 2019). These firms claim to offer more competitive pricing, expedited project schedules, and

* Corresponding author.
E-mail addresses: cao@ibi.baug.ethz.ch (J. Cao), bucher@ibi.baug.ethz.ch (D.F. Bucher), hall@ibi.baug.ethz.ch (D.M. Hall), meggers@projectfrog.com (M. Eggers).

https://doi.org/10.1016/j.compind.2022.103659
0166-3615/

increased product quality. Even so, the overall development and application of such module libraries are still limited in the broader construction sector.

Even when module libraries are developed, they are done so by relying on specialist knowledge and accumulated experience (Salama et al., 2017). Although many researchers propose various techniques and guidelines to support the module identification (Salama et al., 2017; Isaac et al., 2016; Samarasinghe et al., 2019), the configuration of modules is often still determined on a project-by-project basis (Gosling et al., 2016). However, this can lead to many unique typologies of modules that can be difficult to manage. A helpful analogy can be found from LEGO Group in the early 2000s, when the number of unique piece types reached 12,000 and nearly bankrupted the company (Feloni, 2014). Therefore, it is crucial to consider the similarity among projects in order to develop a module library. The similarity indicates that a standard and adaptable design can be applied in multiple projects. The reuse of the modules in future projects will lead to the continuous improvement of project quality and return of investment made initially inside a single project (Tetik et al., 2019).

To facilitate the reuse of the modules, the design is digitalized as building information models (BIM). BIM can provide 3D visualization, monitoring, analysis and prediction during the project life cycle. Each module can be developed as a BIM file via BIM authoring tools, such as Autodesk Revit. A BIM module library consists of diverse models for architectural, civil, structural and mechanical applications. However, a file-based management approach provides limited facility for users to extract the data from the respective BIM authoring tool beyond simple queries on objects and their properties (Solihin et al., 2017). Besides, the query languages introduced so far require a high level of knowledge about the IFC (Industry Foundation Classes) object model and about data mapping mechanisms (Tauscher et al., 2016). Last but not the least, module reuse not only requires the search and comparison in terms of parameters but also the structural composition, which focuses on relationships between different parts of a product.

In this research, we propose to solve these challenges using a graph-based approach for module library development. There are two key components to our solution. First, we propose the use of a graph database – a database that uses graph structures for data storage and data manipulation – to store the BIM models of each module. Second, we propose to identify similar modules in a graph-based module library using a graph similarity analysis. To our knowledge, this is the first study on the development of a building product library via graph database.

The paper is organized as follows. In the next section, we conduct a literature review on the study of modules, focusing on the reusability of module libraries in project development. Next, we generalize the application areas of graph modeling in the construction industry. Specifically, we analyze the use of the graph-based approach for module identification. Then, we introduce the graph database for modules' storage and management. From that, we propose a novel graph-based approach for module library development in the BIM environment. After that, we gave an illustrative example of how the approach is applied to a dataset of timber-framed panels, including a description of system architecture and implementation results. Finally, we discuss the limitations and future research, and conclude with our intended contributions to the literature.

2. Literature review

2.1. Modules in construction projects

Construction literature does not have a consistent definition of a module, modular system or modularization (Gosling et al., 2016). In many instances, modular is used to refer to volumetric assembly units (e.g. factory-prefinished "boxes" that are then transported to the construction site). For example, Murtaza et al. (1993) described a module as "a volume fitted with all structural elements, finishes, and process components that, regardless of system, function, or installing craft, are designed to occupy that space". De La Torre et al. (1994) defined a module as "a product resulting from a series of offsite assembly operations; it is usually the largest transportable unit or component of a facility."

To solve this ambiguity, we look for similar concepts adopted in the manufacturing industry. Salvador categorized five perspectives of product modularity, including component commonality, component combinability, function binding, interface standardization, and loose coupling (Salvador, 2007). Besides the general definition, two characteristics of modular products are highlighted: 1) the similarity between the physical and functional architecture of the design, and 2) the minimization of the degree of the coupling between the physical and functional components (Ericsson and Erixon, 1999).

Accordingly, two modularization methods are similarity- and coupling-based approaches (Borjesson and Hölttä-Otto, 2014). The similarity-based approach to modularization defines modules based on similar functions or properties among products to achieve product variations, or similar manufacturing methods or suppliers, for production efficiency. The coupling-based approach to modularization detects modules by maximizing the coupling within the modules while minimizing the coupling in between the modules. The objective is to reduce the amount of interface management, allowing changes to one module without impacting others, and reducing reciprocal interdependence between modules in order to minimize design feedback (Levitt, 2015; Thompson et al., 2017).

In this paper, we follow the similarity-based approach and define the module in terms of component commonality. To this regard, the modular product design problem refers to the designation of any main component, such as volumetric elements and wall panels, that can benefit from standardized and reusable design in multiple projects.

2.2. Module library development

Due to the loosely-coupled nature of the project-based organization in conventional construction, design reuse remains an underutilized source to improve project performance (Dubois and Gadde, 2002). Industrialized construction has the opportunity to reuse design through the use of a module library of standardized products. The library can support the application of product platforms which contain common processes and technical solutions shared by multiple projects (Bonev et al., 2015; Jensen et al., 2014; Peltokorpi et al., 2018; Jansson et al., 2014), so as to save the design cycle and lower production costs. However, current module libraries in the construction industry do not apply any form of big data analytics for the development and assessment of design options in industrialized construction (Gbadamosi et al., 2020). These design options include the configuration of prefab modules, geometry and material of design parts and components, as well as alternative suppliers and manufacturers. Previous scholars study modules mostly by case studies (Peltokorpi et al., 2018; Viana et al., 2017) and identify the configuration of modules based on their experience and guidelines (Salama et al., 2017). To improve efficiency, there is an opportunity to apply a graph-based product modeling and graph database technology for module library management.

2.3. Graph-based product modeling

In computer science, a graph is an abstract data structure, consisting of a finite set of nodes and a set of ordered or unordered pairs of edges. The structure may also be assigned with certain values to each node or edge, such as a categorical label or a numeric value.

More advanced graph structures, such as hypergraphs, contain edges that can connect any number of nodes. Although a graph can be used to represent complex engineering systems (Boccaletti et al., 2006; Zawiślak and Rysiński, 2017), there is little research studying graph modeling in industrialized construction.

Existing research using graph-based modeling in building design is mostly limited to floorplans and spatial layouts (Wang et al., 2018; Wong and Chan, 2009; Strug et al., 2014; Nauata et al., 2020; Strug and Ślusarczyk, 2009). Few studies tested the graph representation at the granularity of the element level and how it could support module library development (Isaac and Navon, 2013). Khalili and Chua developed a graph-based modeling approach to group single precast elements into higher-level prefabrication assemblies (Khalili and Chua, 2013). They searched for all subgraphs exhaustively and filtered out the feasible configurations by constructability rules. Isaac et al. applied a clustering algorithm to detect optimal configurations of modules in a housing unit (Isaac et al., 2016). A hierarchical clustering algorithm was also taken by Samarasinghe et al. to detect modules in mechanical, electrical and plumbing systems (Samarasinghe et al., 2019). However, the above approaches are based on a single project and detected modules which might not be representative for standardization needed in industrialized construction. An increasing number of unique modules might deteriorate the production efficiency. In addition, the previous studies perform limited similarity analysis for model reuse. Gbadamosi et al. (2020) and Li et al. (2020) propose a semantic-based similarity analysis to retrieve BIM models. The structural similarity of module compositions is not taken into account. Last but not the least, most practices store those modules in a file-based system, without easy access for designers and manufacturers to edit, update, and reuse. To access the data locked inside the BIM tools, previous studies apply a DB link from software developers to export BIM data into a relational database and present domain-related operations as standardized SQL queries (Solihin et al., 2017). Other studies using RDBMS (relational database management system) develop a database structure to integrate data among different software (Solihin et al., 2017; Liu and Issa, 2012; Park and Cai, 2017; Wu et al., 2019). However, a relational database is inefficient to store relationships between different elements.

2.4. Graph database

Compared with other types of database, such as relational database, graph database explicitly builds the dependencies between nodes of data. This characteristic enables the representation of products exactly as they are in the 3D modeling environment or real life. For example, when designers search for similar products, the graph database can not only take into account the existence of components, but also the structures of the product. Other features of using graph database in terms of data structures, query languages and integrity constraints have been summarized by Angles (2012).

Previous studies have introduced the methodologies to extract information from a BIM file (Hor et al., 2018; Ismail et al., 2017; Ismail and Strug, 2018; van Treeck and Rank, 2007; Isaac et al., 2016). However, BIM files are built upon a complicated hierarchical order and reference relationships. They also can contain redundant information. Processing multiple large-scale BIM files is still a time-consuming and error-prone task. Previous studies on BIM-to-graph transformation do not take into account geometric information (Ismail et al., 2017; Ismail and Strug, 2018), which is important for module design and manufacturing. Finally, the adjacency relationships between elements are not explicitly defined in the BIM files. In this study, we build up an easy and user-friendly interface supporting transformation from BIM into a graph database. The elements and their attributes can be extracted, and the adjacency relationships can be detected automatically. The graph database for a module library lay the foundation for design-related tasks.

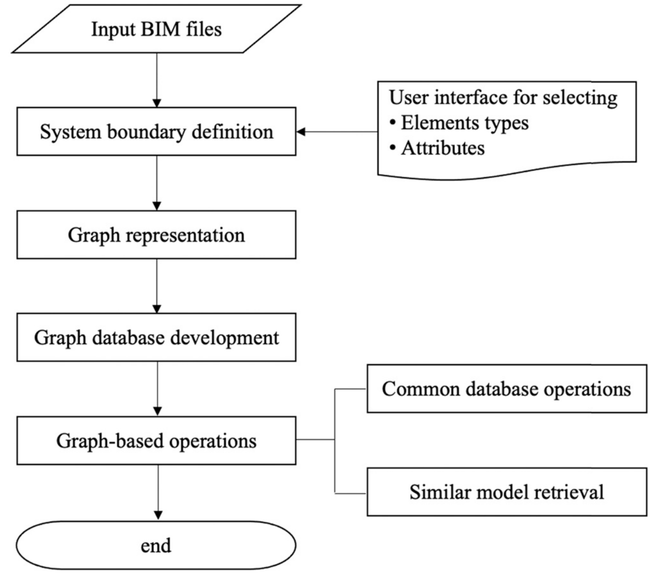

Fig. 1. A approach supporting the development of the module library.

3. Graph-based module library approach

In this section, the proposed graph-based approach for the module library development is described. The approach is composed of four key steps (Fig. 1):

1. Determining the system boundary by selecting building elements and their attributes as inputs.
2. Representing the selected module information as a graph and conducting the same process for all other modules concurrently.
3. Importing graph data into the graph database.
4. Performing graph-based applications, including common graph operations, and similar module retrieval.

3.1. System boundary definition

The categories of a module library in industrialized construction include volumetric modular, flat-pack and kit-of-parts, depending on companys' business strategy (Pullen et al., 2019). To match different strategies, the system boundary is restricted in terms of three aspects: building types, elements types, and element attributes for the library development. Take timber panelized buildings as an example. The input files are a collection of timber panels in different shapes. Each timber panel contains various types of elements, including framing studs, wallboards, insulation layers, and metal fasteners. To simplify the representation of the product, users are able to select the elements "framing studs" as the important components. Finally, users can select dimensions, material properties, or cost of the studs as the element attributes. The selected information is stored in a graph database for similar model retrieval and project-related queries, such as cost estimation.

3.2. Graph representation

The library is built upon the graph representation of selected elements. The graph model has the capacity to represent building elements at different levels of detail and at different stages of the process (Ślusarczyk et al., 2017). In this study, we apply a labeled attributed graph to represent each module, as well as its components. For example, a timber panel contains studs, headers, sills and fasteners. Let us first define a labeled attributed graph.

Definition 1. A labeled attributed graph over the node set (N) and the edge set (E) is a system G = (N, E, L_N, A_N), where:

- N is the node set, representing a set of single building elements.
- E is the edge set, representing a set of relationships between elements, such as adjacency relationships.
- L_N is the label of the node, representing the distinct category of building elements, such as exterior walls.
- A_N is the attributes of nodes, representing the properties of the building elements, such as cost.

In order to build such a graph structure, a web-based tool is programmed in the present research. The tool is written in Javascript using Autodesk Forge, a cloud-based development platform for BIM data manipulation, and perform three main operations for the defined graph, including node identity extraction, node similarity analysis and edge identity detection.

3.2.1. Node identity extraction

This step is aimed to define node identity for graph representation. A node in a graph refers to a single building element, and node attributes refer to its properties. An element in BIM involves a large sum of information, including unique element IDs, geometries, locations, material properties, etc. Through a developed UI (Fig. 2), users are able to visualize all element categories within the project. Once a category is selected, the related properties can be further chosen for extraction. Besides, the interface also enables users to add new attributes for other applications. For example, the assembly sequence of a module that is not usually modeled in original design files can be stored as node attributes to facilitate module assembly.

3.2.2. Node similarity analysis

Building elements in different projects might share similar properties. For example, a timber stud with a length 1 meter is similar to a stud with a length 1.1 ms. The component similarity has a large effect on construction cost and schedule performance (Staub-French and Nepal, 2007). To quantify the variations of element types, we first compute commonality among elements by establishing a common node identity. This step is to compare nodes by incorporating properties into a similarity analysis. We apply K-Means clustering to determine the distinct element types. K-means is an unsupervised learning technique to partition n observations into K clusters in which observations belonging to a cluster are similar to each other. Then, the elbow method is used to decide the optimal number of groups (K). Finally, the elements within each group are considered as the same element and renamed as a sub-category. The results can be checked by users and updated as well. In this way, we promote product standardization at the element level. Fig. 3 shows an example of clustering analysis for panel components, which are categorized based on element length and type. For instance, there are three "2 × 8 SILL" instances, which are further classified into two groups, namely 2.1 and 2.2 by length.

element_length	element_type	normalized_type
11.500000	4x8 STUD	0.1
7.541667	4x8 PLATE TOP	1.1
7.291667	2x8 SILL	2.1
3.750000	2x8 SILL	2.2
3.250000	2x8 SILL	2.2
3.791667	4x8 HEADER	3.1
3.750000	4x8 HEADER	3.1
11.500000	2x8 STUD	4.1

Fig. 3. Node categorization by K-means.

3.2.3. Edge identity detection

The goal of this step is to define the edges of a graph. An edge between a pair of nodes refers to a connection between two elements. The previous study defines three types of connections between building components, including direct connections, indirect connections, and functional connections (Isaac et al., 2016). Connection types can be specified as edge attributes for more precise graph representation. In this study, we apply a 3D collision detection algorithm to determine the spatial relationship (connected or disconnected) between components. The algorithm consists of the following procedures.

1. Generating a bounding box for each element;
2. Retrieving the minimal point (lower-left-rear corner of the box) and the maximal point (upper-right-front corner of the box)
3. Iteratively implementing the logic equation between every two different bounding boxes as

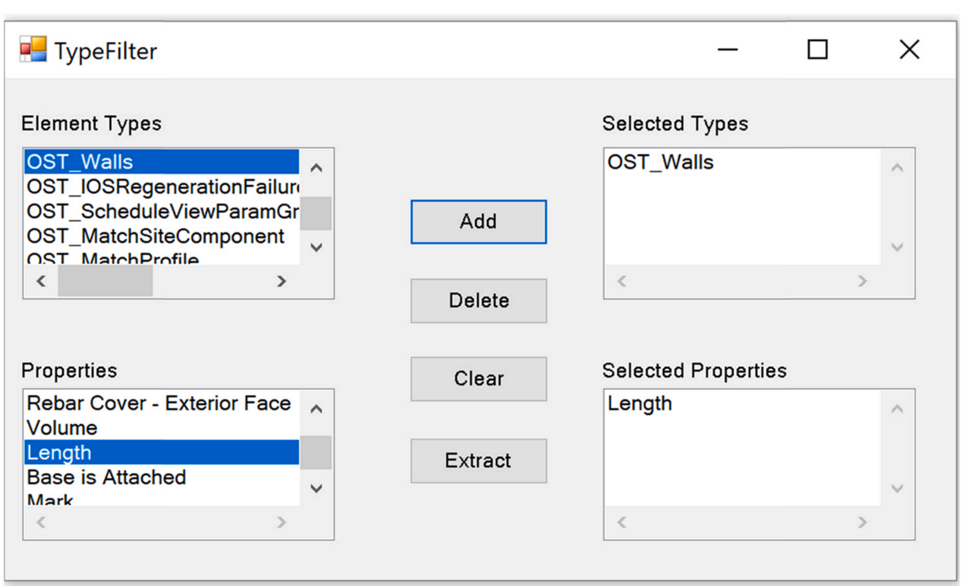

Fig. 2. The user interface of node identity extraction.

$Min_{ui} \leq Max_{vi}$, $Max_{ui} \geq Min_{vi}$

Where i refers to x, y, z coordinates of the minimal point and the maximal point. For example, Min_{ux} stands for the x coordinate of the minimal point of the bounding box u. If the equations are satisfied with regard to x, y, z, the two bounding boxes (elements) are connected. Otherwise, they are not connected.

4. Returning an adjacency matrix of a graph.

3.3. Graph database development

Finally, the adjacency matrix is transformed into a graph structure, and the extracted element categories and properties are attached to node labels and node attributes respectively. After the designs of BIM files are transformed into graphs, this next step is to store them in a graph database. In this study, Neo4j, a widely-used graph database, is set up for data storage. As Neo4j is a schema-less database, data stored on a disk is all linked lists of fixed-size records. More details about the database development are illustrated in Section 4.1 by an implementation.

3.4. Graph database applications

Once the database is developed, common operations, such as insertion, deletion, update and selection, can be implemented in Cypher query language to visit data efficiently. Cypher is like SQL a declarative, textual query language, but for graphs. Besides, customized functions can be developed for domain applications. To support module library reuse, we build a customized function to enable similar module retrieval based on graph similarity analysis.

3.4.1. Common database operations

Storing BIM data in the graph database has advantages over other database systems. First, the graph database can help create a better representation of the bill of materials from design and engineering data sources. Since a product is represented as a graph, the components for manufacturing the product can be easily obtained by traversing all nodes of a graph, while RDBMS and other NoSQL databases typically see significant performance degradation when traversing data beyond three levels of hierarchical depth (Rathle, 2020). Apart from the product data, the supply chain data, such as suppliers, customer data, such as claims, can also be stored as nodes and connected to the component nodes in the graph. With such a demand-product-supply view of BOM, the communication among sales teams, engineering teams, manufacturers, and suppliers can be sped up. Second, the graph database can help manage design changes. Changes in a construction project can occur during the design of building components. The changes can have direct impacts on the components which belong to the same system or indirect impacts on components which belong to different systems. The interdependency between components can be represented as an attribute of the edges. Once a component is revised, a real-time alert will be triggered to notify the users of the affected components, which are connected by the changed component. The functionality benefits the BIM users as BIM only offer visualization of the new model design without highlighting the components that are affected by the changed components (Moayeri et al., 2017).

3.4.2. Graph similarity analysis

In graph theory, two techniques can be used to conduct graph similarity analysis: 1. graph edit distance (GED) and 2. maximum common subgraphs (MCS). In this research, we use a GED-based similarity analysis. The graph edit distance is first proposed by Sanfeliu and Fu (1983). It is defined as the transformation from the input graph to the target graph by six operations, including insertion, deletion and substitution of nodes and edges. Mathematically, the GED is formalized as follows:

Definition 2. The graph edit distance between graph G1 and graph G2 is defined as:

$$GED(G1, G2) = \min_{(e_1, \ldots, e_k) \in P(G1,G2)} \sum_{i=1}^{k} c(e_i) \quad (1)$$

where P(G1, G2) refers to the set of operations transforming G1 to G2. Practically, it can embody the assembly operations needed to manufacture a product. The fewer operations needed, the more similar the two products are. To indicate the similarity clearly, the GED is further converted into the similarity score between (0,1]. The conversion is performed as:

$$\text{normalized } GED(G1, G2) = e^{-\frac{GED(G1,G2)}{\frac{|G1|+|G2|}{2}}} \quad (2)$$

A widely used method to compute GED is A* algorithm (Riesen et al., 2007). However, the A* algorithm suffers from high memory consumption, as it explores all possible mappings between two graphs. The problem is even evident for large graphs. Considering the building assemblies usually contain many components, the GED computation might take a long time. Many improved solutions have been proposed in the literature. In this study, we apply an upgraded A* algorithm which uses a depth-first search strategy with less memory and computation time (Abu-Aisheh et al., 2015). The algorithm is packed in the Python NetworkX.

3.4.3. Similar module retrieval

This function is to retrieve similar modules according to users' input design. Modules with similar structures might serve for the same design intention. Compared with the traditional keyword search approach, the graph-based search can involve more structural information by taking advantage of the graph data structure. For prefabricated assemblies, not only the geometry data but components data, such as the bill of materials (BOM), are crucial to determine the similarity in terms of manufacturing cost and time. The retrieval is achieved by calculating the normalized GED score between the users' input design with the modules in the library. The similar design is ranked by the GED score from high to low as output. Additionally, by calculating the GED scores, the pairs of modules with GED scores that exceed a threshold can be categorized as the same group. For mass customization, it is important for companies to manage the trade-off between the variability and economy by controlling the number of modules in the same category.

For the effective reuse of modules in the database, the library also incorporates a multi-criteria similarity analysis according to users' demands. In the case of complex modules, designers not only consider the product structural similarity, but also building-related performances (e.g., cost, loading capacity, footprint, etc.) to satisfy local building regulations and project-specific requirements. Those performances can be derived from the element properties in the empirical formula. For example, the load capacity of a timber panel is one of the important design factors. The capacity is dependent on panel length, height, stud spacing, sheathing nail spacing, sheathing thickness, presence of drywall and opening, and different support conditions (Quayyum, 2020). With the properties stored in the database, we can perform a parameter-based model to calculate load capacity for each panel. Then, the reuse query can be encoded as a vector of metrics that signify the preferred performance of the product. Final outputs of module candidates can thus be retrieved by mapping user queries to product properties.

4. Test case

In order to verify the validity of our proposed approach, we have developed a prototype system for a timber panel library (shown in

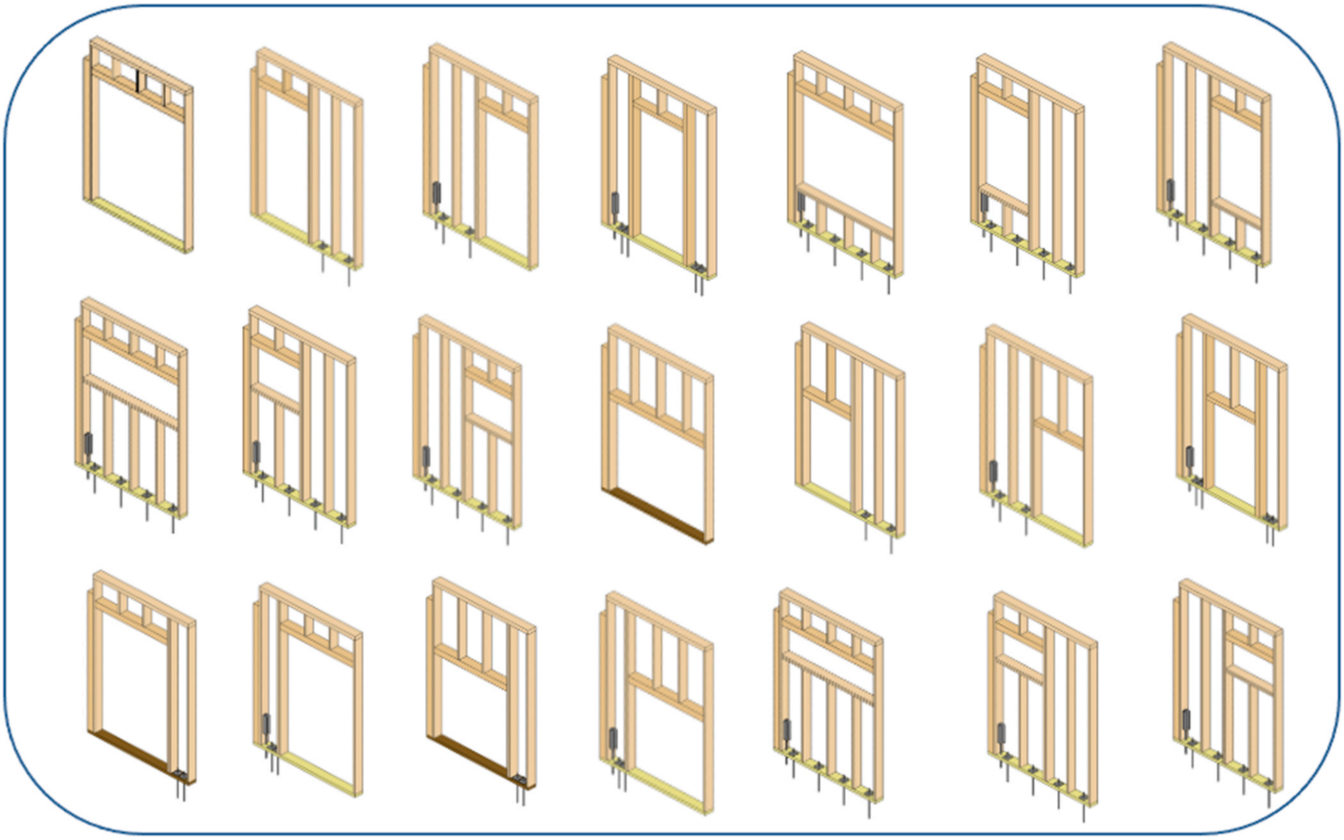

Fig. 4. Example of an existing timber panel library used by industrialized construction firm.

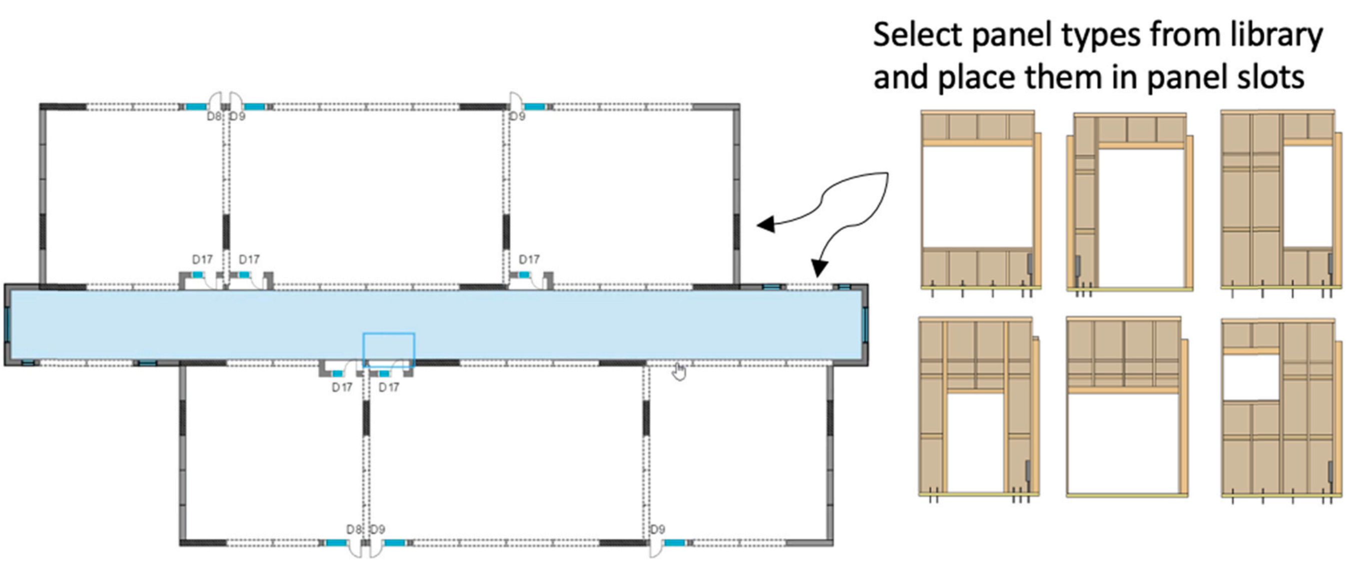

Fig. 5. Example of a technical configurator for timber-panelized buildings.

Fig. 4). The library is originated from an industrialized construction firm in the United States, mainly focusing on timber construction. Their projects consist of prefabricated timber panels, that are combined in various configurations to create mass-customized buildings. The technical configurator (Fig. 5) is supported by a library of 3D models in Autodesk Revit, 2D engineering drawings in AutoCAD, and technical documents that define all relevant design regulations. Using the configurator, the architects can first define the space boundary and room partitions on a grid system. Then, they select panel types from the library and place them in panel slots. Finally, a structural analysis engine in the backend will check the design based on local building regulations. The prototype aims at supporting the panel library storage, sharing, and reuse.

4.1. Complexity and relevance

As a simple example, we select a timber-framed panel product for a test case. Timber prefabricated panels are an increasingly popular and utilized industrialized building system (Staib et al., 2008). The panel consists of multiple parts and sub-assemblies,

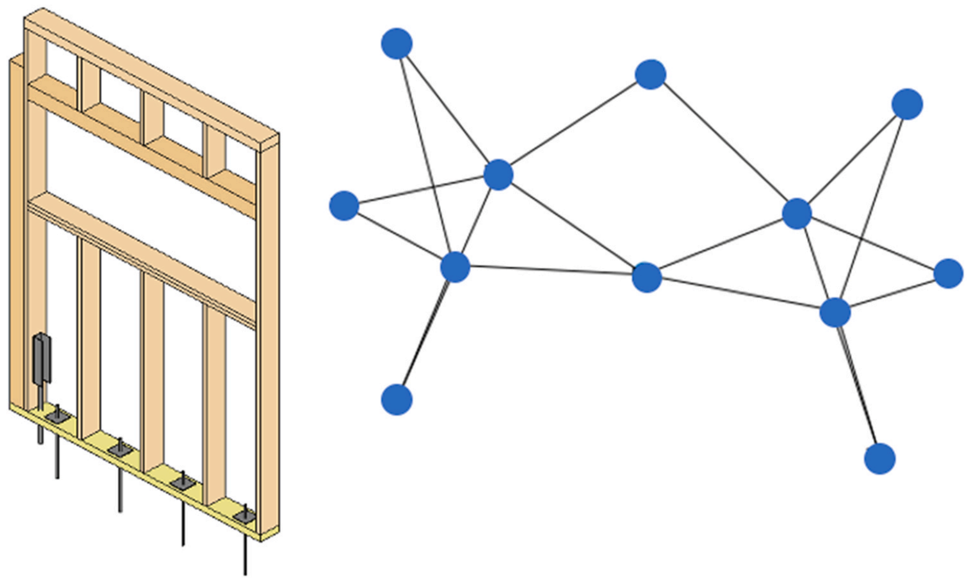

Fig. 6. Graph representation of a panel where nodes represent structural studs and edges represent connection relationships.

making it a good case to demonstrate structural complexity. A high-level fabricated panel integrates structural frames, insulations, services, vapor control layers, internal linings, doors and windows, joineries and claddings. In the test case, we limit the components of a panel to the structural frame. By doing so, we do not lose the generality because the structural diversity of panels mostly lies in the frame layout and geometry. To be specific, the stud type, material, dimensions, and spatial relationship are studied in illustrated applications. For more complex building products, such as prefabricated pods, more components and properties need to be taken into consideration. The design process requires collaboration among different trades. This might require data integration among different trades' databases.

4.2. System architecture and data model

The global system architecture, as well as the data model were developed with a focus on modularity and exemplary characteristics. Specifically, we aim for a prototype that is reproducible by others and presents the main concepts of development in a simple way. Globally, the prototype was implemented with a three-tier architecture, using the relatively new mono repository (monorepo) approach. This provides for all applications and libraries to be contained in the same repository and benefits the development with easy dependency management. Changes can be easily verified across all affected parts of the system. Our prototype includes a frontend based on the Angular framework and the backend implemented in NestJS. We use a graph-based database solution, in our case Neo4J, connecting to the backend. It consists not of tables, but of the data nodes with the properties and the edges accounting for relations.

Upon launching the platform, the backend ensures that the input model is retrieved from Autodesk Forge. The geometry and properties are processed and decoupled from each other. Geometry is extracted by Forge Viewer API and sent directly to the frontend, while properties are extracted by Forge Model Derivative API and implemented by casting into the Parts class. Next, the generated objects are stored as nodes in the database. At this point, the edges between the nodes are not obtained yet. A 3D collision detection algorithm is performed to calculate the neighbors of the objects via the UUID (Universally Unique Identifier). Derived from this calculation, objects of the Relation class are created as two connected node pairs and added to the database. Finally, a complete graph representation has been created. Fig. 6 shows an example of timber panel model and corresponding graph representation. The backend can now access the data and perform design-related tasks. The frontend can connect the geometry to parts with relations for visualization and show the results of the processed tasks. Fig. 7 shows the system architecture and Fig. 8 shows the data processing workflow.

4.3. Application and results

The application tier of the prototype processes the main functionalities. This research mainly paid attention to the design-related tasks, including the bill-of-materials (BOM) generation, identification of impacts of design changes, and similar model retrieval. More application services, such as cost analysis, could be added to the prototype to support the module library reuse.

4.3.1. Bill-of-materials generation

The bill-of-materials (BOM) is the key to information systems within industrialized construction. A BOM can define a product as they are designed, as they are ordered, as they are built and as they are maintained. It acts as a bridge connecting islands of information, such as MRP (Material Resource Planning), ERP (Enterprise Resource Planning), and PDM (Product Data Management). Those systems share the same information as BOM with different views.

In existing BIM applications, a BOM is generated by selecting element categories and filtering all individuals within those categories. In a large construction project, such an approach cannot automatically display the assembly-part relationship. For example, it is difficult for production managers to know which specific parts under the "stud" category are used in "Panel A205", and further conduct quality inspection. By comparison, an assembly is stored as a graph and all associated components are stored as nodes. Fig. 9 shows the manufacturing BOM of "Panel A205", which comprises a collection of components, together with the required quantities. To support the production process, the BOM also contains product specifications, such as raw material, cutting length, etc.

4.3.2. Identification of impacts of design changes

Existing BIM applications are not capable of automatically manifesting the effect of a design change, unless the change violates a rigid constraint, such as a clash. As a result, designers need to keep the changes in mind and check the effects manually. Instead, with

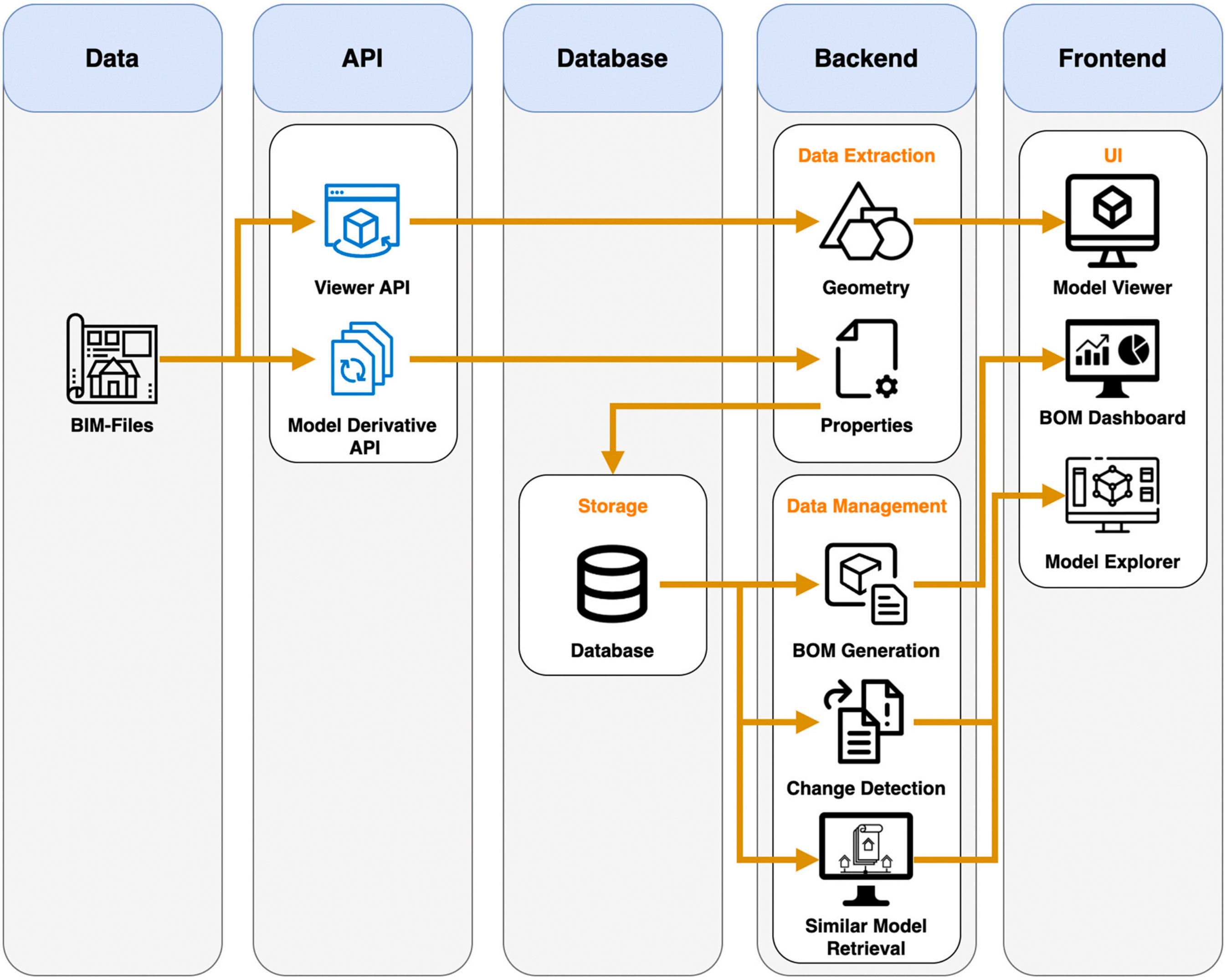

Fig. 7. System architecture.

the graph model, the element dependencies, such as physical adjacency, are explicitly represented and stored as edges. The dependency modeling assists designers to identify the components affected by a proposed change. In this case, the panels comprise studs, headers, sills, plates, and fasteners. Changing any of the components will directly affect its nearby components. The effect can be indicated by the neighborhood of the changed node. A transaction will be implemented in the graph database to perform the changes made. The affected building elements can be visualized through the interface. As an example in Fig. 10, the size of the header (light blue) is changed, the studs on the top of the header and on the two ends of the header are affected and colored in red. If the change is made by mistake, the designers can easily recover the results committed by the transaction and maintain consistency with the data stored in the backend.

4.3.3. Similar model retrieval

The similar model retrieval in the existing product library is built upon the attribute-based approach. For example, a query like “a timber panel with an opening” takes material and architectural features into account. By matching the attributes given by users' requests and product information, similar products are obtained. However, this approach does not consider the structural similarity and therefore is not efficient for complex assemblies analysis, such as framed panels. As a result, the same query will return all panels in Fig. 9.

To demonstrate the proposed approach is effective to retrieve similar models, we attempt to set a target model and return a set of similar models. Fig. 11 provides a sample of the results. The panel models in the first column are used as targets, and a set of similar models from the library are retrieved in the following columns. In the first two rows, the targets are panels with window openings. The difference between the two targets is the window width. In the last two rows, the targets are panels with door openings. The difference is the door width. The type of the opening (window or door) and the width of the opening affects the stud framing of the panels. Especially around the opening area, a double stud is needed. To help users find the most similar models, the outputs are ranked based on the GED score (below) from high to low. In this study, we filtered out the outputs with GED scores lower than 0.6. From the results, it is clear that the retrieved models belong to the same types of target panels and share a similar structure and geometry. One thing needs to be noted. The threshold of GED score can be determined by users and is dependent on the model types. If we lower the threshold of GED score, the panels that belong to different types will be regarded as similar ones. However, this will not affect the search results.

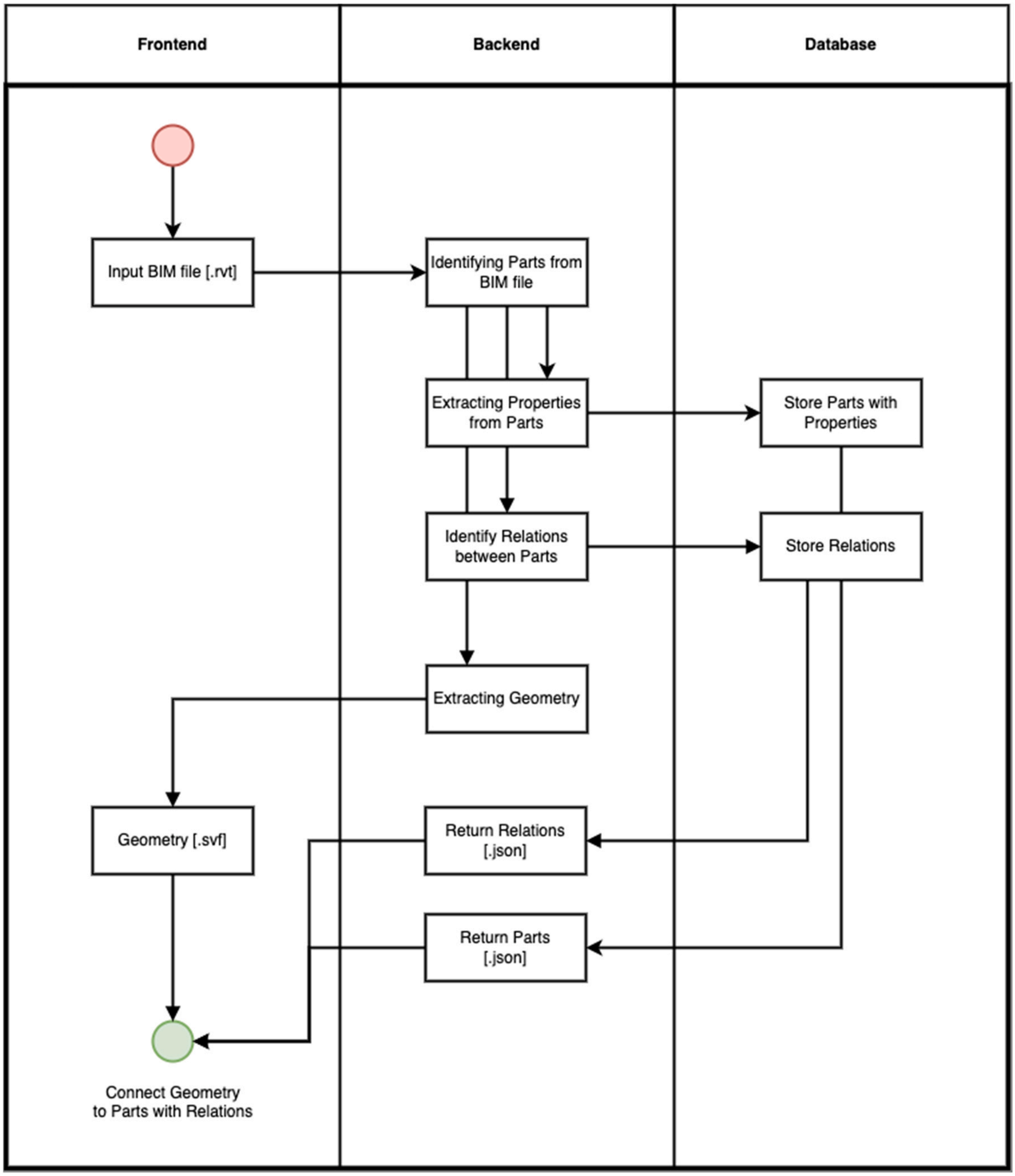

Fig. 8. Data processing workflow.

5. Limitation and future research

The study offers a novel graph-based approach for module library development. Previous studies neglect the importance of the reuse of the prefabricated modules due to the loosely coupled nature of construction projects (Dubois and Gadde, 2002). Industrialized construction shares a degree of repeatability and standardization (Bertram et al., 2019). Therefore, by developing such module libraries, industrialized construction firms can speed up their project development cycle and boost their productivity. To continue the research on product library development, we provide the following discussion on the limitations and future research.

5.1. An industry product platform

The industrialized construction industry is seeking a platform solution that can be deployed across multiple projects and sectors (Cao et al., 2021). This platform consists of a scalable module library. Previous studies demonstrated that a BIM product repository can simplify the design process and improve product efficiency (Gbadamosi et al., 2020; Nath et al., 2015). To support models' storing, sharing, and reuse, this research builds upon a collection of BIM models and establish a graph-based approach. However, in our developed prototype, the library is limited to one specific company doing timber panelized construction. To achieve a supply-chain

	TypeName	Material	CuttingLength	Quantities
1	"6x8 STUD"	"DFL #1"	3137.0	1
2	"2x8 STUD"	"DFL #2"	3505.0	2
3	"2x8 STUD CRIPPLE"	"DFL #2"	711.0	3
4	"3x8 PLATE BOTTOM"	"DF PT"	2438.0	1
5	"HDU8"	null	null	1
6	"FASTENER_ANCHOR_BOLT_BASE_PLATE"	null	null	4
7	"4x8 PLATE TOP"	"DFL #2"	2299.0	1
8	"2x8 SILL"	"DFL #2"	2223.0	2
9	"6x8 HEADER"	"DFL #1"	2223.0	1
10	"2x8 STUD CRIPPLE"	"DFL #2"	502.0	3

Fig. 9. The manufacturing BOM of the panel "A205".

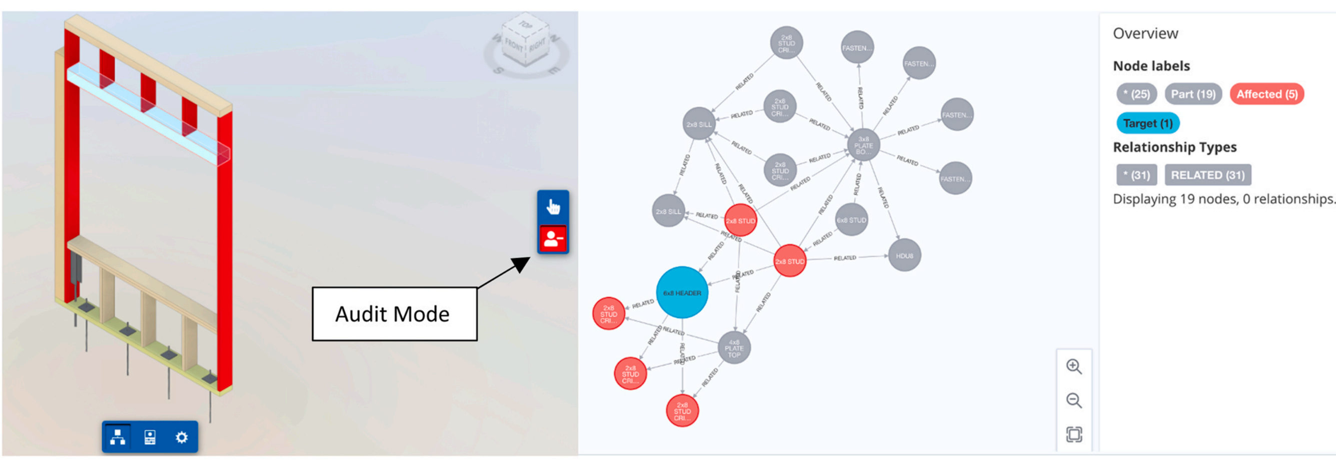

Fig. 10. The red studs are affected when the blue header is changed [the model view (left) and database view (right)].

ecosystem to enable mass customization, a platform would need to be developed that enabled products from multiple companies to be added to this platform and shared with the industry. When compared to existing product libraries such as NBS Library, the use of a graph-based approach can help recommend specific complementary products to help develop a module library. This idea is aligned with the "Platform Design Programme" initiated by Construction Innovation Hub. The industry participants can form an alliance and engage in the supply chain network. A decentralized framework can be utilized to build the industrial level platform (Jiang et al., 2021), to facilitate cross-enterprise information sharing among multiple stakeholders with transparency and security. In that case, when clients attempt to launch industrialized buildings on the platform, they can easily retrieve the manufacturing quote, as well as an estimated production schedule, from different companies.

5.2. Product variety derivation and auditing

An increasing number of unique modules in the library might deteriorate product efficiency. In this research, we conduct a similar model retrieval to compare the new design with the existing ones. If a high degree of similarity exists, the new design will not be added to the library as a unique module to maintain economies of scale. Furthermore, if a firm has completed multiple projects and wishes to reduce the number of unique modules, they can use the model retrieval to audit their existing library. When several modules demonstrate similarity, this can facilitate a structured and data-driven conversation about which modules to keep and which to remove from the library.

For a company that wishes to begin the development of a module library (or to begin a new product line), there is an opportunity to

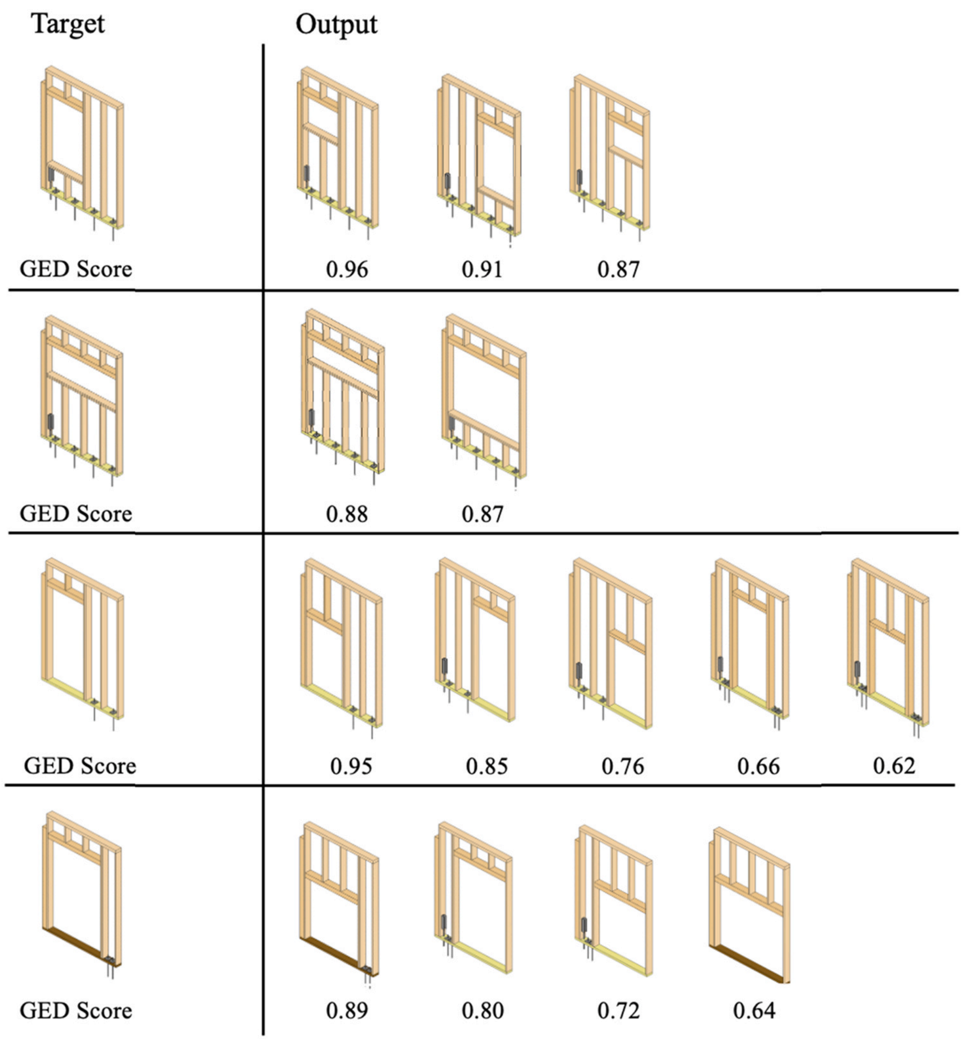

Fig. 11. Similar panels retrieval based on GED score calculation.

automate the development of new modules with increasing product variety. Previous studies have demonstrated the graph-based modeling can be used for automatic product variety derivation, but most are limited in the floor plan generation (Wang et al., 2018; Wong and Chan, 2009; Strug et al., 2014; Nauata et al., 2020). To extend the application to the modular building design, graph grammars can be implemented on the graph represented modules as future research. The graph grammars denote a series of graph transformations from base products to a set of distinct end products. The typical transformations include attaching, removing, swapping and scaling (Du et al., 2002) existing modules. These grammars will be triggered by the customer-selected product features or project requirements. As a result, a quick response to the clients' needs can be generated by configuring modules from the library.

5.3. Multi-criteria reuse analysis

For the module library reuse in new projects, multiple decision-making factors need to be considered, such as cost, lead time of production, etc. for different project settings. Other performance simulations can be added with a similar approach, such as LCA analysis (Hollberg et al., 2020). This lightweight solution replaces the process of importing CAD models into authorized software for running simulation. As a result, it mitigates error-prone data exchange and unnecessary remodeling. However, we do not explore an optimal module selection under multiple criteria, especially among different types of modular products, such as timber panels versus volumetric modules. For volumetric modules, there exist different decision-making factors affecting the module selection (Hwang et al., 2018). Therefore, to facilitate module library reuse, the next step is to build a recommendation system incorporating key decision-making factors for industrialized building design. It is beneficial for clients to be notified of the optimal product when there exist hundreds of product choices.

6. Conclusion

Industrialized construction offers a faster completion time, better quality and is more environmental-friendly than traditional construction (Ferdous et al., 2019), especially for projects with a degree of standardization. An increasing number of stakeholders adopt this approach by delivering their projects from design to manufacturing and assembly. Modular products owned by industrialized construction firms is a set of prefabricated components, including volumetric modules, flat-packs and kit-of-parts (Pullen et al., 2019). Although the categorization of modules (Gosling et al., 2016), as well as a variety of modules' designs (Liew et al., 2019; Lawson and Ogden, 2008), have been illustrated in previous studies, the application of a module library is not being deployed at scale. Not all clients are aware of using them crossing multiple projects. One of the significant reasons is a lack of a sharable industry product platform where stakeholders can easily obtain modules and adapt them to new project settings. As a result, modules are mostly developed project by project, losing the benefits of economies of scale.

To enable the adoption of industrialized construction by a greater segment of the industry, this research proposes a graph-based approach based on the BIM environment to support the storage, sharing and reuse of the module library. This paper attempts to make several contributions to the literature. Most importantly, we apply a graph database for module library development. A collection of modules can be stored as graphs in the database. Graph modeling has been used in many applications related to project design and control, such as floor planning, space navigation and modular design. Considering those applications are usually built independently and supported by different systems, the development of a graph database for module library will facilitate those applications in terms of common information management. Besides, to support module reuse in future projects, we create three application scenarios where a module library can be used for design-related tasks, including BOM generation, identification of impacts of design change, and similar module retrieval. Finally, a prototype system built upon web technology is developed and implemented in a timber panel system of an industrialized construction firm. This makes the first step towards the development of a shared, industry-wide product platform for mass customization.

CRediT authorship contribution statement

Jianpeng Cao: Conceptualization, Methodology, Data curation, Writing – original draft, Software. **David F. Bucher:** Software, Visualization, Writing – review & editing. **Daniel Hall:** Supervision, Writing – review & editing, Funding acquisition. **Mike Eggers:** Resources.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by Swiss National Science Foundation [Grant number: 197294].

References

Abu-Aisheh, Z., Raveaux R., Ramel J.Y., Martineau, P., 2015. An exact graph edit distance algorithm for solving pattern recognition problems. In: ICPRAM 2015 – 4th International Conference on Pattern Recognition Applications and Methods, Proceedings, pp. 271–8. ⟨https://doi.org/10.5220/0005209202710278⟩.

Anderson, David M., 2003. Design for Manufacturability & Concurrent Engineering; How to Design for Low Cost, Design in High Quality, Design for Lean Manufacture, and Design Quickly for Fast Production. CIM press. (ISBN: 978-1878072238).

Angles, A., 2012. A comparison of current graph database models. In: Proceedings – 2012 IEEE 28th International Conference on Data Engineering Workshops, ICDEW 2012, IEEE, pp. 171–7. ⟨https://doi.org/10.1109/ICDEW.2012.31⟩.

Bertram, N., Fuchs, S., Mischke, Palter R., Strube, G., Woetzel, J., 2019. Modular construction: from projects to products. In: McKinsey & Company: Capital Projects & Infrastructure, pp. 1–34. ⟨https://www.ivvd.nl/wp-content/uploads/2019/12/Modular-construction-from-projects-to-products-full-report-NEW.pdf⟩. (Accessed 8 March 2020).

Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U., 2006. Complex networks: structure and dynamics. Phys. Rep. 424, 175–308. https://doi.org/10.1016/j.physrep.2005.10.009

Bonev, M., Wörösch, M., Hvam, L., 2015. Utilizing platforms in industrialized construction: a case study of a precast manufacturer. Constr. Innov. 15, 84–106. https://doi.org/10.1108/CI-04-2014-0023

Borjesson, F., Hölttä-Otto, K., 2014. A module generation algorithm for product architecture based on component interactions and strategic drivers. Res. Eng. Des. 25, 31–51. https://doi.org/10.1007/s00163-013-0164-2

Cao, J., Bucher, D.F., Hall, D.M., Lessing, J., 2021. Cross-phase product configurator for modular buildings using kit-of-parts. Autom. Constr. 123, 103437. https://doi.org/10.1016/j.autcon.2020.103437

Construction Innovation Hub. Platform Design Programme Defining the Need. ⟨https://constructioninnovationhub.org.uk/wp-content/uploads/2020/12/FINAL_VERSION_DEFINING-THE-NEED-REPORT-DEC-2020.pdf⟩. (Accessed 8 March 2020).

Cui, Y., Li, S., Liu, C., Sun, N., 2020. Creation and diversified applications of plane module libraries for prefabricated houses based on BIM. Sustainability 12, 453. https://doi.org/10.3390/su12020453

Dubois, A., Gadde, L.E., 2002. The construction industry as a loosely coupled system: implications for productivity and innovation. Constr. Manag. Econ. 20, 621–631. https://doi.org/10.1080/01446190210163543

Du, X., Jiao, J., Tseng, M.M., 2002. Graph grammar based product family modeling. Concurr. Eng. Res. Appl. 113–128. https://doi.org/10.1177/106329303039729

Ericsson, A., Erixon, G., 1999. Controlling Design Variants: Modular Product Platforms. Society of Manufacturing Engineers, Dearborn, MI (ISBN: 978-0872635142).

Feloni, R., 2014. How Lego Came Back From the Brink of Bankruptcy. ⟨https://www.businessinsider.com/how-lego-made-a-huge-turnaround-2014–2?r=US&IR=T⟩. (Accessed 8 March 2020).

Ferdous, W., Bai, Y., Ngo, T.D., Manalo, A., Mendis, P., 2019. New advancements, challenges and opportunities of multi-storey modular buildings – a state-of-the-art review. Eng. Struct. 183, 883–893. https://doi.org/10.1016/j.engstruct.2019.01.061

Gao, S., Jin, R., Lu, W., 2020. Design for manufacture and assembly in construction: a review. Build. Res. Inf. 48, 538–550. https://doi.org/10.1080/09613218.2019.1660608

Gbadamosi, A.Q., Oyedele, L., Mahamadu, A.M., Kusimo, H., Bilal, M., Davila Delgado, J.M., Muhammed-Yakubu, N., 2020. Big data for design options repository: towards a DFMA approach for offsite construction. Autom. Constr. 120, 103388. https://doi.org/10.1016/j.autcon.2020.103388

Gosling, J., Pero, M., Schoenwitz, M., Towill, D., Cigolini, R., 2016. Defining and categorizing modules in building projects: an international perspective. J. Constr. Eng. Manag. 142, 1–11. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001181

Hollberg, A., Genova, G., Habert, G., 2020. Evaluation of BIM-based LCA results for building design. Autom. Constr. 109, 102972. https://doi.org/10.1016/j.autcon.2019.102972

Hor, A.E.H., Sohn, G., Claudio, P., Jadidi, M., Afnan, A., 2018. A semantic graph database for BIM-GIS integrated information model for an intelligent urban mobility web application. ISPRS Ann. Photogramm., Remote Sens. Spat. Inf. Sci. 4, 89–96. https://doi.org/10.5194/isprs-annals-IV-4-89-2018

Hwang, B.G., Shan, M., Looi, K.Y., 2018. Knowledge-based decision support system for prefabricated prefinished volumetric construction. Autom. Constr. 94, 168–178. https://doi.org/10.1016/j.autcon.2018.06.016

Isaac, S., Bock, T., Stoliar, Y., 2016. A methodology for the optimal modularization of building design. Autom. Constr. 65, 116–124. https://doi.org/10.1016/j.autcon.2015.12.017

Isaac, S., Navon, R., 2013. A graph-based model for the identification of the impact of design changes. Autom. Constr. 31, 31–40. https://doi.org/10.1016/j.autcon.2012.11.043

Ismail, A., Nahar, A., Scherer, R., 2017. Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard. In: Digital Proceedings of the 24th EG-ICE International Workshop on Intelligent Computing in Engineering 2017, pp. 146–57. ⟨https://www.researchgate.net/publication/318600860_Application_of_graph_databases_and_graph_theory_concepts_for_advanced_analysing_of_BIM_models_based_on_IFC_standard⟩. (Accessed 8 March 2020).

Ismail, A., Strug, B., 2018. Building knowledge extraction from BIM/IFC data for analysis in graph databases. In: Proceedings of the International Conference on Artificial Intelligence and Soft Computing, p. 652–64. ⟨https://doi.org/10.1007/978-3-319-91262-2⟩.

Jansson, G., Johnsson, H., Engström, D., 2014. Platform use in systems building. Constr. Manag. Econ. 32, 70–82. https://doi.org/10.1080/01446193.2013.793376

Jensen, P., Olofsson, T., Smiding, E., Gerth, R., 2014. Developing products in product platforms in the AEC industry. In: Computing in Civil and Building Engineering 2014. American Society of Civil Engineers, Reston, VA, pp. 1062–1069. https://doi.org/10.1061/9780784413616.132

Jiang, Y., Liu, X., Kang, K., Wang, Z., Zhong, R.Y., Huang, G.Q., 2021. Blockchain-enabled cyber-physical smart modular integrated construction. Comput. Ind. 133, 103553. https://doi.org/10.1016/j.compind.2021.103553

Khalili, A., Chua, D.K.H., 2013. IFC-based framework to move beyond individual building elements toward configuring a higher level of prefabrication. J. Comput. Civ. Eng. 27, 243–253. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000203

Lawson, R.M., Ogden, R.G., 2008. "Hybrid" light steel panel and modular systems. Thin-Walled Struct. 46, 720–730. https://doi.org/10.1016/j.tws.2008.01.042

Levitt, R.E., 2015. An extended coordination theory for governance of tasks, projects, firms, and business ecosystems., Working Paper, Global Projects Center, Stanford University. ⟨http://gpc.stanford.edu/publications/extended-coordination-theory-governance-tasks-projects-firms-and-business-ecosystems⟩. (Accessed 7 June 2019).

Liew, J.Y.R., Chua, Y.S., Dai, Z., 2019. Steel concrete composite systems for modular construction of high-rise buildings. Structures 21, 135–149. https://doi.org/10.1016/j.istruc.2019.02.010

Liu, R., Issa, R.R.A., 2012. Automatically updating maintenance information from a BIM database. In: Computing in Civil Engineering 2012. American Society of Civil Engineers, pp. 373–380. https://doi.org/10.1061/9780784412343.0047

Li, N., Li, Q., Liu, Y.S., Lu, W., Wang, W., 2020. BIMSeek++: retrieving BIM components using similarity measurement of attributes. Comput. Ind. 116, 103186. https://doi.org/10.1016/j.compind.2020.103186

Lu, W., Tan, T., Xu, J., Wang, J., Chen, K., Gao, S., Xue, F., 2020. Design for manufacture and assembly (DfMA) in construction: the old and the new. Archit. Eng. Des. Manag. 0, 1–15. https://doi.org/10.1080/17452007.2020.1768505
Moayeri, V., Moselhi, O., Zhu, Z., 2017. Design change management using BIM-based visualization model. Int. J. Archit., Eng. Constr. 6, 1–9. https://doi.org/10.7492/ijaec.2017.001
Murtaza, M.B., Fisher, D.J., Skibniewski, M.J., 1993. Knowledge-based approach to modular construction decision support. J. Constr. Eng. Manag. 119, 115–130. https://doi.org/10.1061/(ASCE)0733-9364(1993)119:1(115)
Nath, T., Attarzadeh, M., Tiong, R.L.K., Chidambaram, C., Yu, Z., 2015. Automation in construction productivity improvement of precast shop drawings generation through BIM-based process re-engineering. Autom. Constr. 54, 54–68. https://doi.org/10.1016/j.autcon.2015.03.014
Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G., Furukawa, Y., 2020. House-GAN: relational generative adversarial networks for graph-constrained house layout generation. In: European Conference on Computer Vision, pp. 162–77. ⟨https://arxiv.org/abs/2003.06988⟩. (Accessed 8 March 2020).
Park, J., Cai, H., 2017. WBS-based dynamic multi-dimensional BIM database for total construction as-built documentation. Autom. Constr. 77, 15–23. https://doi.org/10.1016/j.autcon.2017.01.021
Peltokorpi, A., Olivieri, H., Granja, A.D., Seppänen, O., 2018. Categorizing modularization strategies to achieve various objectives of building investments. Constr. Manag. Econ. 36, 32–48. https://doi.org/10.1080/01446193.2017.1353119
Pullen, T., Hall, D.M., Lessing, J., 2019. White Paper: A Preliminary Overview of Emerging Trends for Industrialized Construction in the United States, pp. 1–24. ⟨https://doi.org/10.3929/ethz-b-000331901⟩.
Quayyum, S., 2020. Parametric models for wind load resistances of wood-frame walls. J. Struct. Eng. 146, 04019188. https://doi.org/10.1061/(asce)st.1943-541x.0002471
Rathle, R., 2020. Driving Innovation in Retail with Graph Technology How Top Retailers Use Neo4j. ⟨https://neo4j.com/whitepapers/retailers-graph-technology-neo4j/⟩. (Accessed 6 May 2020).
Riesen, K., Fankhauser, S., Bunke, H., 2007. Speeding up graph edit distance computation with a bipartite heuristic. In: Mining and Learning with Graphs, pp. 21–4. ⟨http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.6893&rep=rep1&type=pdf⟩. (Accessed 8 March 2020).
Salama, T., Salah, A., Moselhi, O., Al-Hussein, M., 2017. Near optimum selection of module configuration for efficient modular construction. Autom. Constr. 83, 316–329. https://doi.org/10.1016/j.autcon.2017.03.008
Salvador, F., 2007. Toward a product system modularity construct: literature review and reconceptualization. IEEE Trans. Eng. Manag. 54, 219–240. https://doi.org/10.1109/TEM.2007.893996
Samarasinghe, T., Gunawardena, T., Mendis, P., Sofi, M., Aye, L., 2019. Dependency structure matrix and hierarchical clustering based algorithm for optimum module identification in MEP systems. Autom. Constr. 104, 153–178. https://doi.org/10.1016/j.autcon.2019.03.021
Sanfeliu, A., Fu, K.-S., 1983. A distance measure between attributed relational graphs for pattern recognition. Trans. Syst. Man Cybern. 13, 353–362. https://doi.org/10.1109/TSMC.1983.6313167
Ślusarczyk, G., Łachwa, A., Palacz, W., Strug, B., Paszyńska, A., Grabska, E., 2017. An extended hierarchical graph-based building model for design and engineering problems. Autom. Constr. 74, 95–102. https://doi.org/10.1016/j.autcon.2016.11.008
Solihin, W., Eastman, C., Lee, Y.C., Yang, D.H., 2017. A simplified relational database schema for transformation of BIM data into a query-efficient and spatially enabled database. Autom. Constr. 84, 367–383. https://doi.org/10.1016/j.autcon.2017.10.002
Staib, G., Dörrhöfer, A., Rosenthal, M., 2008. Components and Systems Modular Construction – Design, Structure, New Technologies. Walter de Gruyterhttps://doi.org/10.11129/detail.9783034615662
Staub-French, S., Nepal, M.P., 2007. Reasoning about component similarity in building product models from the construction perspective. Autom. Constr. 17, 11–21. https://doi.org/10.1016/j.autcon.2007.02.013
Strug, B., Grabska, E., Ślusarczyk, G., 2014. Supporting the design process with hypergraph genetic operators. Adv. Eng. Inform. 28, 11–27. https://doi.org/10.1016/j.aei.2013.10.002
Strug, B., Ślusarczyk, G., 2009. Reasoning about designs through frequent patterns mining. Adv. Eng. Inform. 23, 361–369. https://doi.org/10.1016/j.aei.2009.06.007
Tauscher, E., Bargstädt, H.-J., Smarsly, K., 2016. Generic BIM queries based on the IFC object model using graph theory. In: Proceedings of the 16th International Conference on Computing in Civil and Building Engineering. Article 04014046.
Tetik, M., Peltokorpi, A., Seppänen, O., Holmström, J., 2019. Direct digital construction: Technology-based operations management practice for continuous improvement of construction industry performance. Autom. Constr. 107, 102910. https://doi.org/10.1016/j.autcon.2019.102910
Thompson, J.D., Zald, M.N., Scott, W.R., 2017. Organizations in Action. Routledge (ISBN: 9781315125930).
De La Torre, M.L., Sause, R., Slaughter, S., Hendricks, R.H., 1994. Review and Analysis of Modular Construction Practices. Lehigh University, pp. 1–152. ⟨https://preserve.lib.lehigh.edu/islandora/object/preserve%3Abp-4308350⟩. (Accessed 8 March 2020).
van Treeck, C., Rank, E., 2007. Dimensional reduction of 3D building models using graph theory and its application in building energy simulation. Eng. Comput. 23, 109–122. https://doi.org/10.1007/s00366-006-0053-7
Viana, D., Tommelein, I., Formoso, C., 2017. Using modularity to reduce complexity of industrialized building systems for mass customization. Energies 10, 1622. https://doi.org/10.3390/en10101622
Wang, X., Yang, Y., Zhang, K., 2018. Customization and generation of floor plans based on graph transformations. Autom. Constr. 94, 405–416. https://doi.org/10.1016/j.autcon.2018.07.017
Wong, S.S.Y., Chan, K.C.C., 2009. EvoArch: an evolutionary algorithm for architectural layout design. CAD Comput. Aided Des. 41, 649–667. https://doi.org/10.1016/j.cad.2009.04.005
Wu, S., Shen, Q., Deng, Y., Cheng, J., 2019. Natural-language-based intelligent retrieval engine for BIM object database. Comput. Ind. 108, 73–88. https://doi.org/10.1016/j.compind.2019.02.016
Zawiślak, S., Rysiński, J., 2017. Graph-Based Modelling in Engineering. Springer, Cham. https://doi.org/10.1007/978-3-319-39020-8