MODULE LIBRARY DEVELOPMENT VIA GRAPH MINING

Jianpeng Cao, Daniel Hall
Institute of Construction and Infrastructure Management
ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

ABSTRACT

The use of prefabricated modules can benefit the construction industry with the economy of scales and production efficiency. However, the existing approach to develop module libraries is project-based, lacking the potential to reuse and manage in future projects. By taking the repeatability and manufacturability into account, this paper proposes a graph-based framework to identify possible modules automatically from multiple projects by frequent pattern mining. The results show that the repeated patterns share a degree of standardization and can be considered as module candidates. Finally, the framework is implemented as add-ons in the BIM environment to support module lifecycle management.

INTRODUCTION

Modular construction is experiencing a new wave of attention and investment. In 2019 McKinsey report, the construction industry could deliver a $20 billion annual savings and 50% time saving if modular construction is adopted (McKinsey Capital Projects, 2019). The modular construction takes the strategy of DfMA (Design for Manufacturing and Assembly), developing standard types of modules, like LEGOs, and then giving those kit-of-parts to architects to fulfill design intentions (WikiHouse, 2019). The strategy benefits modular construction in particular for the structure sharing a degree of repeatability and standardization (McKinsey Capital Projects, 2019), such as hotels, affordable housing and schools. Moreover, the standardization does not mean the design has to be "one-size-fits-all". Instead, modular construction could provide design flexibility in various configurations by selecting and recombining modules from module libraries (Cui et al., 2020). New industrialized construction firms, such as DMD Modular, Project Frog, CIMC MBS and etc., start to develop their own module libraries as their core products, or product catalog, and claim to offer more competitive pricing, expedited project schedules, and increased product quality.

The development and application of module libraries are still limited in construction projects. The development is mostly built upon specialist knowledge and accumulated experience. Although many researchers propose various techniques and guidelines to support the module identification (Isaac, Bock and Stoliar, 2016; Salama *et al.*, 2017; Samarasinghe *et al.*, 2019), the configuration of modules is still determined project by project (Gosling et al., 2016). In that case, the unique typology of modules will grow rapidly, end up with costs to balloon. A helpful analogy can be found from LEGO Group in the early 2000s, when the unique pieces reaching 12,000, nearly bankrupted the company (Feloni, 2014). Therefore, it is crucial to consider repeatability among projects in determining a module library or product catalog. The repeatability 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).

This research proposes a graph-based framework for identifying a reusable module library from historical projects. To take repeatability into account, we apply the frequent pattern mining to a dataset of building projects. Besides, to facilitate library lifecycle management, BIM (Building Information Model) is used as product digital representations. Unlike early studies of knowledge reuse, such as drawings (Strug and Ślusarczyk, 2009) and simulations (Hiyama et al., 2014), BIM modules can be semantically enriched for their use in the design and production phases (Hamid, Tolba and El Antably, 2018). Meanwhile, the information stored in BIM supports the module analysis in terms of their performance, such as manufacturability. In this way, previous studies on metric-based module identification (Salama et al., 2017) can be integrated within our framework.

The paper is organized as follows. In the next section, we conduct a literature review on the study of modules and the application of the graph modeling in the construction industry. Then, we introduce the proposed graph-based framework for module identification in the BIM environment. After that, we give an illustrative example of how the framework is applied to a dataset of residential projects. Finally, a discussion and conclusion of our contribution to the literature, as well as future work, is given.

LITERATURE REVIEW

Modules in construction projects

In the construction industry, the definition of a module, or a modular system and modularization, is not explained consistently in the construction literature (Gosling et al., 2016). Murtaza, Fisher and Skibniewski (Murtaza, Fisher and Skibniewski, 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”. More recent definitions not only refer to a module as a structural unit composed of walls, floors, ceilings, as well as their finishes, but also highlight the off-site fabrication (Hwang, Shan and Looi, 2018) and supply chain integration (Peltokorpi et al., 2018). In this study, we inherit the definition of the module above, and enrich it with a library of standardized and reusable design which can be applied to multiple projects. Previous studies have shown that a library of parametric prefabricated components can simplify the design process and improve production efficiency (Nath et al., 2015).

Graph modeling in construction projects

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 which can connect any number of nodes. Although graph has been proved to be a useful way to represent complex engineering systems (Boccaletti et al., 2006; Zawiślak and Rysiński, 2017), there is not much research studying the graph modeling in construction projects. Existing research using graph-based modeling in building design is mostly limited in the floorplans and spatial layouts. In that case, each room or space is represented as a node, and the adjacency relations are represented as edges. Node attributes represent the properties of rooms. The graph structure, as well as graph-based algorithms, facilitate the application of floorplan design and indoor navigation.

There are three main approaches related to floorplan design via graph-based modeling, including graph transformations (Wang, Yang and Zhang, 2018), evolutionary approach (Wong and Chan, 2009; Strug, Grabska and Ślusarczyk, 2014), and deep learning approach (Nauata et al., 2020). The graph transformation approach is built upon input graphs representing the original floorplans, and then graph manipulations, such as node/edge addition and subtraction, are performed to produce the floorplan variations (Wang, Yang and Zhang, 2018). The evolutionary approach introduces graph-based evolutionary operators, namely cross-over and mutation, in the floorplan generation process (Wong and Chan, 2009; Strug, Grabska and Ślusarczyk, 2014) The deep learning approach is achieved via a Generative Adversarial Networks (GAN), which takes a large dataset of pixel-based floorplans as inputs and generate novel ones by performing a generator and a discriminator on their graph representations (Nauata et al., 2020). Having graph-represented design solutions of floorplans, Strug and Ślusarczyk detected the frequent patterns via graph mining technique (Strug and Ślusarczyk, 2009). These patterns are further used as design features to evaluate the new floorplan design (Strug, 2013).

Through a similar representation approach, the graph-based floorplans also enable space navigation. Skandhakumar et al. constructed a graph from IFC files to represent a floorplan. The space adjacency and accessibility relationship represented in the graph facilitate the BIM in the navigation and access control applications (Skandhakumar et al., 2016). Ślusarczyk et al. proposed a multi-hierarchical graph to represent three-dimension of a floorplan, including spatial arrangement, accessibility, and administrative structure of the building. The shortest path algorithm is then performed for the application of mobile robot control for a mail delivery task (Ślusarczyk et al., 2017).

Although above and many other studies have shown that graph-based modeling is suitable to represent construction projects, few studies tested the graph representation at the granularity of the element level and how it could support construction project management (Isaac and Navon, 2013). Within this domain, one of the main research directions is to detect prefabricated modules for industrialized construction. Previous scholars study modules mostly by case studies (Viana, Tommelein and Formoso, 2017; Peltokorpi et al., 2018), and identify the configuration of modules based on their experience and guidelines (Salama et al., 2017). Graph-based modeling can support the detection of modules automatically or semi-automatically. 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, Bock and Stoliar applied a clustering algorithm to detect optimal configurations of modules (Isaac, Bock and Stoliar, 2016b). Those modules are represented as subgraphs, which share two characteristics: 1. relationships are dense within the subgraph and are sparse between the subgraph. 2. the nodes, representing the building elements, in the subgraph have similar attributes, such as replacing rates. A similar 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 research rarely takes into account the repeatability and manufacturability in discovering the building modules. Besides, the approaches are based on a single project, ending up with the detected modules which might not be representative for standardization. An increasing number of unique modules might deteriorate the production efficiency. Last but not least, the previous studies end up with graph represented solutions, rather than BIMs (Strug and Ślusarczyk, 2009; Isaac, Bock and Stoliar, 2016; Samarasinghe et al., 2019). How could those graph patterns be managed as BIMs remains unsolved. In this research, we detect repeated and representative modules in multiples building projects, and then validate the modules using manufacturing constraints. Finally, the results are visualized and stored in the BIM environment for future reuse.

PROPOSED FRAMEWORK

In this section, the proposed graph mining framework for the module library development is described. The framework is composed of the following steps (Figure 1):.

1. Determining the system boundary by selecting building elements and their attributes from projects as inputs
2. Representing the filtered project information as a graph and conducting the same process for all past projects
3. Performing frequent pattern mining on the graph datasets by defining the size and frequency of the patterns
4. Conducting pattern analysis, which includes the shape analysis and manufacturability analysis
5. Mapping the graph patterns to the original BIM objects for visualization and if necessary, for further management

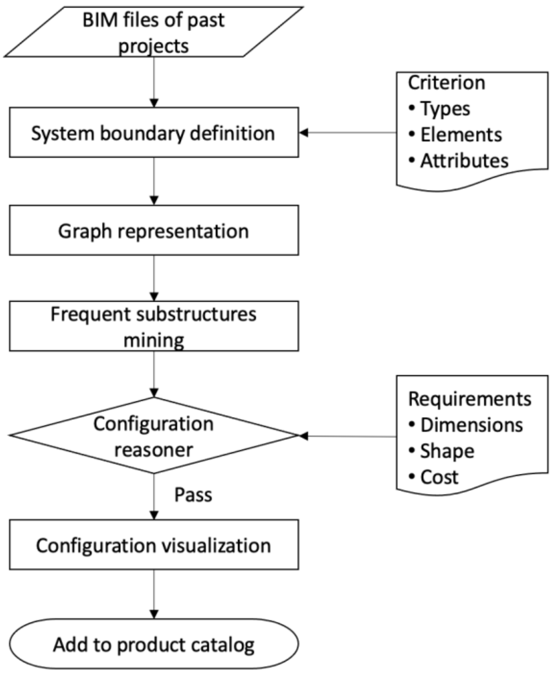

Figure 1: Proposed framework architecture

System boundary definition

The system boundary is restricted in terms of three aspects, building types, elements types, and element attributes. Firstly, since the module library is a collection of components suitable for a specific type of building, such as prefabricated concrete buildings, the BIM file inputs representing projects should be limited within the specified category. Secondly, a project usually contains hundreds of building elements. To narrow down the scope of modules, the users are able to select the element types for the library development. For example, a company doing timber volumetric modules might select wall and floor panels as building parts. Thirdly, the attributes of building elements, such as dimensions, material properties, and cost, can be used to support the graph representation and the pattern analysis in the following configuration reasoner.

Graph representation

The graph model can be used to represent design objects 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 the design representation. 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 sets, representing a set of building elements.
- E is the edge sets, representing a set of relationships between elements, such as adjacency relationship.
- 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.

In order to build such a graph structure, we first utilize the BIM application programming interface (API) to perform data extraction from design input files. The extracted data include the selected element categories and properties. Then, a 3D collision detection algorithm is implemented to determine the spatial relationship between those elements, and returns an adjacency matrix. Finally, the matrix is transformed into a graph structure, and the extracted categories and properties are attached to node labels and attributes.

Frequent pattern mining

Frequent pattern mining is used to discover repeated or similar substructures in the multiple graph structures. It was widely applied in bioinformatics (Mrzic et al., 2018). Let us define the frequent pattern as follows:

Definition 2. A subgraph (g) is a frequent pattern if the number of its occurrence (S), including its belonging nodes and edges, exceed a specified threshold in the entire dataset, where:

- g is a subgraph of the graph G. The node set and edge set of g are denoted as n and e respectively, where $n \subseteq N$ and $e \subseteq E$.

In the construction domain, these frequent substructures represent a cluster of building elements that repeatedly occurred in the examined projects. In some sense, they might represent a standard design or "common knowledge" from previous works (Strug and Ślusarczyk, 2009). In this study, we apply "gSpan" algorithm (Yan and Han, 2002) to potential module identification. The algorithm is implemented in a depth-first search (DFS) approach and generates the subgraphs via the right-most path extension. It combines the subgraph search with the isomorphism testing, thus achieving efficient mining in a large graph set. The work is implemented in the python environment.

Configuration reasoner

The configuration reasoner is aimed at retrieving reasonable configurations from all frequent patterns. This is done in two steps. Firstly, shape analysis is performed on identified frequent subgraphs. By calculating graph properties, such as degrees and circles, the shape of represented substructures can be inferred. For example, a subgraph with a circle, namely the cyclic graph, might represent a closed substructure, which might be a volumetric module. Secondly, with the element attributes embedded in the nodes of a subgraph, a manufacturability analysis is further conducted. For instance, the length and weight of elements can be used to filter out the configurations feasible for manufacturing. In addition, a graph-based structural analysis might be necessary for some substructural system, such as frames. The structural analysis is not in the scope of this research, and might be included in the future step. The related study can be found in Chang's work (Chang and Cheng, 2020).

Configuration visualization

The development of the module library is an iterative process, which incorporates different requirements in the reasoning process. To support specialists' visualization and validation of feasible candidates, we map the subgraph patterns back to building information models. This is achieved by a subgraph matching algorithm, developed by Bonnici et al. (Bonnici et al., 2013). Firstly, the targeted subgraph is compared to all possible subgraphs in the graph datasets, and return any subgraphs which is a subgraph isomorphism. Then, the element ID stored as attributes in the original graph datasets can be obtained. As a result, the considered subgraph can be tracked and visualized as a 3D BIM model. We implemented this function with BIM API, so as to highlight the substructures in the project. Once the configurations are checked by specialists, the substructures will be added into the module library for future reuse.

ILLUSTRATIVE CASE

The main objective of this section is to validate the performance of the proposed framework. In this study, we select residential timber projects as illustrative cases. First, we collect two standard types of modules (Figure 2) from a collaborative company, and use them to create ten customized floorplans in BIM. Figure 3 shows some examples of design. To test the robustness of the framework, we add some changes to each module. These include 1. adding and deleting interior walls to partition rooms; 2. swapping wall types to suit room requirments. For example, the thickness of structural walls are 260 mm, while the thickness of non-structural walls can be 100 mm and 120 mm. Besides, the walls for the bathroom are furnished with tiles. By doing so, we obtain modules with different functions, such as a living room module and a module with two bedrooms and a bathroom. To be noted, this is exactly how the studied company and many other modular companies might do to achieve product diversification. Besides, the implementation also inserts some "noise" to the input data, so as to test whether the framework can still detect the unchanged parts, or the standard parts of the original modules.

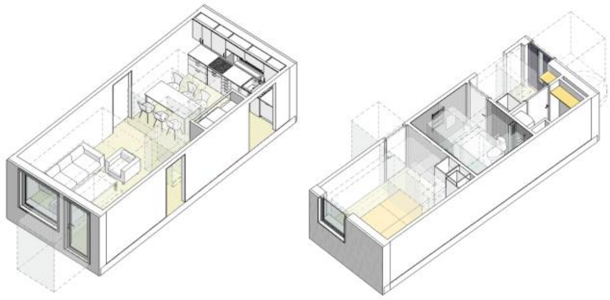

Figure 2: Two standard modules

Figure 3: Examples of customized floorplans

Next, the building elements, as well as their properties are extracted from the BIM files using the developed add-on (Figure 4). In this case, we select walls and floors as element categories, and the length of walls and the element ID as element properties from the design files.

Two documents are generated as outputs automatically: an attribute file and a relationship file, in the .csv format. The attribute file contains the selected types of elements (e.g. OST_Walls), and rows of associated properties (e.g. Length). The relationship file decodes the spatial relationship between elements as an adjacency matrix. In the matrix, the spatial relationships include having physical connections referred to as 1 and having no connection as 0. More types of connections, such as plate connections, can be specified for higher level-of-detail models.

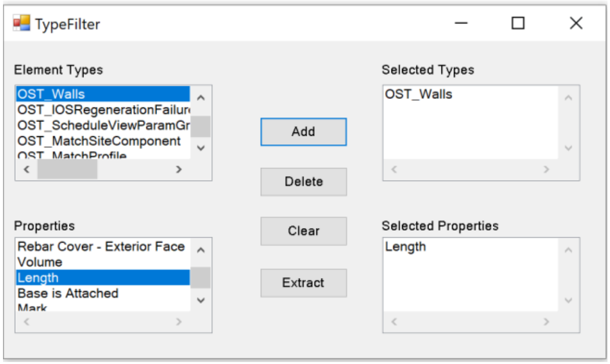

Figure 4: Add-ons for BIM data extraction

After the data extraction, we implemented a python script to transform two documents into graph representations. Each building element is matched to a node in the graph, while the properties of that element are assigned to the node attributes. Besides, the adjacency matrix specifies the edge connection between nodes. An example of a transformed graph is shown in Figure 5. This step is done using the Python library NetworkX.

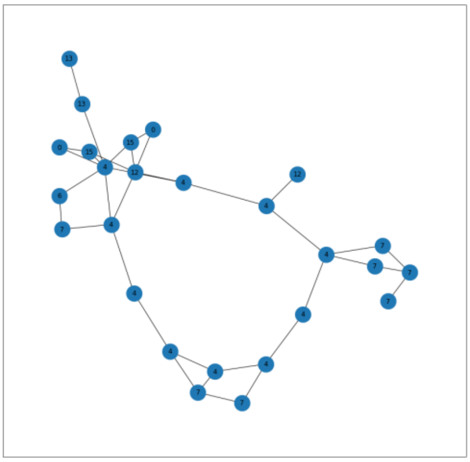
Figure 5: An example of graph representations

The procedure after the graph representation is to detect frequent subgraphs. A number of parameters can be set to support the subgraph mining. Those include the minimal and maximal nodes in the subgraph, and the number of the subgraph's occurrence, namely support. Examples of the identified subgraphs with the number of nodes equals 6 and the frequency equals 10 are displayed in Figure 6. The index on the nodes indicates the type of building elements. In Figure 6, the graphs represent two closed substructures. Both composed of three exterior structural walls (node 1) and two different types of interior walls (node 2, 6, 8), all of which are connected to the floor (node 0). They can be considered as partial structures of a volumetric module. With more elements selected at the system boundary definition step, the patterns will be more complicated and include more features, such as ceilings and openings. Other information of the structure, such as the length and cost, can be obtained from the node attributes.

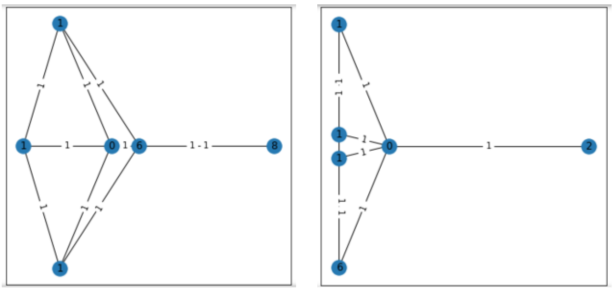
Figure 6: Examples of graph representations

To further validate the modular structure, we apply the manufacturing restrictions on the detected subgraphs. We consider an acceptable timber module is in the range of 3 m in width and 8 m in length and 3.2 m high. Other criteria can be set by selecting the targeted element property and applying math equations. The application window of the developed add-on is depicted in Figure 7. Finally, the detected subgraphs are matched in the original project datasets for visualization. Figure 8 displays the frequent subgraph in Figure 6 (left) in the BIM environment.

Comparing the detected structures with original models, we found that the structures match the original modules in terms of their four side wall types and floor types, and do not match the interior walls which are varied from projects to projects. It indicates that the framework is robust to extract the standard and repeated structures from multiple projects.

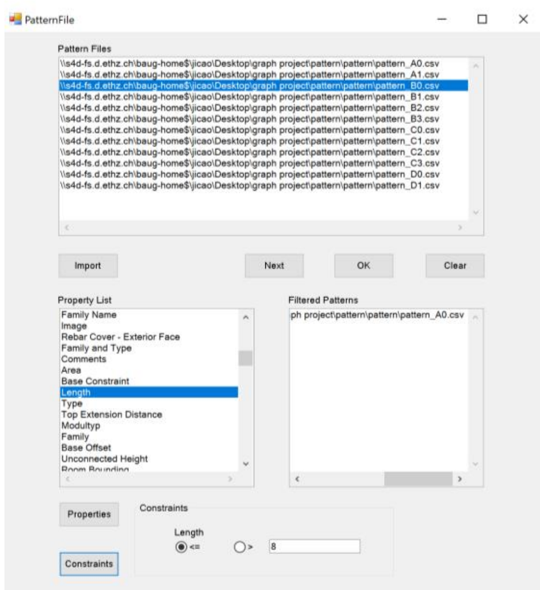

Figure 7: Configuration reasoner

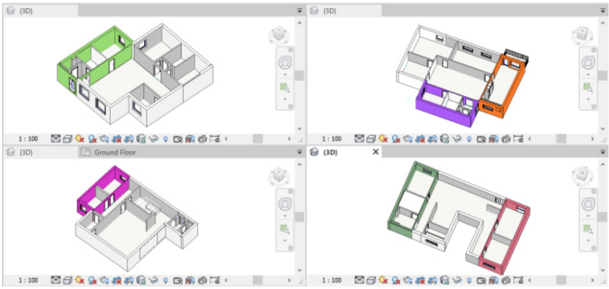

Figure 8: Frequent subgraph visualization in BIM

DISCUSSION & LIMITATION

The proposed framework provides a new method for the module library development by extracting frequent patterns from past projects. This matches the nature of most modular construction which shares a degree of repeatability. Besides, since the detected modules exist frequently in past projects, it might indicate a standard structure or process. Therefore, by developing such module libraries, the industrialized construction firms could not only obtain their featured product catalog but boost their productivity through standardization.

The above work emphasizes the opportunity for productization of traditional construction projects. Past scholarship has suggested the need to achieve a higher level of product modularization in order to minimize the total number of components (Khalili and Chua, 2013). Similar research related to the optimal configuration of modules also highlights the importance of partitioning the building structures into independent components, so as to eliminate the interference caused by design change (Isaac and Navon, 2013). However, these approaches are project-based module identification. The modules might have different typologies in projects. With an increasing number of unique modules, the company might lose the production efficiency guaranteed by mass production.

By comparison, the framework presented in this study requires efforts to build a reusable module library. Since the modules are identified from past projects, the supply chain information, such as cost, quality and production time, etc., is easy to be obtained. When a new project comes in, the company can switch out one module for an alternative if required and notice the performance change immediately. In addition, the reuse of those modules will provide a continuous improvement cycle for companies to upgrade their projects and gain the return of investment from a single project (Tetik et al., 2019). As a result, the application of reusable modules can help the industry achieve economy of scale and integration along the value chain.

However, the framework has certain limitations. First, the detected patterns contain many irregular configurations. The same problem also exists in Khalili's research (Khalili and Chua, 2013). Existing shape analysis functions need to be improved. Secondly, historical project data might not share a consistent naming convention, causing lots of trouble to identify identical element types during the data preprocessing. A possible solution is to identify the same element by their common properties. Thirdly, the study does not incorporate structural analysis. Local boundary conditions, as well as connection types, are not specified clearly in most BIM files. Finally, the illustrative case demonstrates that the framework can detect standard and repeated patterns in the projects. More project settings need to be done, especially for industrialized construction firms without module libraries.

CONCLUSION

Modular construction demonstrates a series of benefits over traditional construction for appropriate project settings (Ferdous et al., 2019). More stakeholders adopt this approach by delivering their products from design to manufacturing and assembly. A module library is an important property owned by industrialized construction firms. The definition and categorization of modules have been illustrated in previous studies (Gosling et al., 2016) but the development of a reusable module library is limited due to the neglect of repeatability. To enable the adoption of modular construction by a greater segment of the industry, this research proposes a graph-based framework to identify modules from historical data automatically.

This paper attempts to make several contributions to the literature. First, it enriches the existing study of graph-based approach in the modular building design. The proposed framework represents design files as graphs and applied frequent pattern mining to identify the potential reusable modular structures. Second, it proves that manufacturability of the design can be integrated into the module development. The configuration reasoner could identify volumetric modules which satisfy manufacturing constraints. Third, the identified modules are tracked in the BIM environment for visualization and management. A BIM-based module library can benefit the integration of data from multi-stakeholders throughout the project life cycle.

Future research should investigate additional questions such as:

1. How could the module library be used to replace the existing design, to achieve economy of scales? This might require techniques of pattern matching.
2. How could design flexibility be implemented with a module library? A possible solution can be dividing the design into standard parts detected by our framework and remaining part for customization.
3. How scalable of the framework is regarding other project contexts? We plan to test the framework in different contextual settings.

REFERENCE

Boccaletti, S. et al. (2006) Complex networks: Structure and dynamics, *Physics Reports*, 424(4–5), pp. 175–308.

Bonnici, V. et al. (2013) A subgraph isomorphism algorithm and its application to biochemical data, *BMC Bioinformatics*. BioMed Central Ltd, 14(SUPPL7), p. S13.

Chang, K.-H. and Cheng, C.-Y. (2020) Learning to simulate and design for structural engineering. arXiv:2003.09103[cs.LG]

Cui, Y. et al. (2020) Creation and Diversified Applications of Plane Module Libraries for Prefabricated Houses Based on BIM, *Sustainability*, 12(2), p.453.

Feloni, R. (2014) How Lego Came Back From The Brink Of Bankruptcy, viewed Dec 2020, < https://www.businessinsider.com/how-lego-made-a-huge-turnaround-2014-2?r=US&IR=T>.

Ferdous, W. et al. (2019) New advancements, challenges and opportunities of multi-storey modular buildings – A state-of-the-art review, *Engineering Structures,* 183(January), pp. 883–893.

Gosling, J. et al. (2016) Defining and Categorizing Modules in Building Projects: An International Perspective, *Journal of Construction Engineering and Management*, 142(11), pp. 1–11.

Hamid, M., Tolba, O. and El Antably, A. (2018) BIM semantics for digital fabrication: A knowledge-based approach, *Automation in Construction*, 91(March), pp. 62–82.

Hiyama, K. et al. (2014) A new method for reusing building information models of past projects to optimize the default configuration for performance simulations, *Energy and Buildings*, 73, pp. 83–91.

Hwang, B., Shan, M. and Looi, K. (2018) Key constraints and mitigation strategies for prefabricated pre fi nished volumetric construction, *Journal of Cleaner Production*, 183, pp. 183–193.

Isaac, S., Bock, T. and Stoliar, Y. (2016) A methodology for the optimal modularization of building design, *Automation in Construction*, 65, pp. 116–124.

Isaac, S. and Navon, R. (2013) A graph-based model for the identification of the impact of design changes, *Automation in Construction*, 31, pp. 31–40.

Khalili, A. and Chua, D. K. H. (2013) IFC-Based Framework to Move beyond Individual Building Elements toward Configuring a Higher Level of Prefabrication, *Journal of Computing in Civil Engineering*, 27(3), pp. 243–253.

McKinsey Capital Projects (2019) Modular construction : From projects to products, viewed Dec 2020, <https://www.mckinsey.com/business-functions/operations/our-insights/modular-construction-from-projects-to-products>.

Mrzic, A. et al. (2018) Grasping frequent subgraph mining for bioinformatics applications, *BioData Mining*, 11(1), pp. 1–24.

Murtaza, M. B., Fisher, D. J. and Skibniewski, M. J. (1993) Knowledge-Based Approach to Modular Construction Decision Support, *Journal of Construction Engineering and Management*, 119(1), pp. 115–130.

Nath, T. et al. (2015) Automation in Construction Productivity improvement of precast shop drawings generation through BIM-based process re-engineering, *Automation in Construction*, 54, pp. 54–68.

Nauata, N. et al. (2020) House-GAN: Relational Generative Adversarial Networks for Graph-constrained House Layout Generation. arXiv:2003.06988[cs.CV]

Peltokorpi, A. et al. (2018) Categorizing modularization strategies to achieve various objectives of building investments, *Construction Management and Economics*, 36(1), pp. 32–48.

Salama, T. et al. (2017) Near optimum selection of module configuration for efficient modular construction, *Automation in Construction*, 83, pp. 316–329.

Samarasinghe, T. et al. (2019) Dependency Structure Matrix and Hierarchical Clustering based algorithm for optimum module identification in MEP systems, *Automation in Construction*, 104(May 2018), pp. 153–178.

Skandhakumar, N. et al. (2016) Graph theory based representation of building information models for access control applications, *Automation in Construction*, 68, pp. 44–51.

Ślusarczyk, G. et al. (2017) An extended hierarchical graph-based building model for design and engineering problems, *Automation in Construction*, 74, pp. 95–102.

Strug, B. (2013) Automatic design quality evaluation using graph similarity measures, *Automation in Construction*, 32, pp. 187–195.

Strug, B., Grabska, E. and Ślusarczyk, G. (2014) Supporting the design process with hypergraph genetic operators, *Advanced Engineering Informatics*, 28(1), pp. 11–27.

Strug, B. and Ślusarczyk, G. (2009) Reasoning about designs through frequent patterns mining, *Advanced Engineering Informatics*, 23(4), pp. 361–369.

Tetik, M. et al. (2019) Direct digital construction: Technology-based operations management practice for continuous improvement of construction industry performance, *Automation in Construction*, 107(July), p. 102910.

Viana, D., Tommelein, I. and Formoso, C. (2017) Using Modularity to Reduce Complexity of Industrialized Building Systems for Mass Customization, *Energies*, 10(10), p. 1622.

Wang, X., Yang, Y. and Zhang, K. (2018) Customization and generation of floor plans based on graph transformations, *Automation in Construction*, 94(December 2017), pp. 405–416.

WikiHouse (2019) The DfMA Housing Manual, viewed Dec 2020, <https://docs.google.com/document/d/1OiLXP7QJ2h4wMbdmypQByAi_fso7zWjLSdg8Lf4KvaY/edit>.

Wong, S. S. Y. and Chan, K. C. C. (2009) EvoArch: An evolutionary algorithm for architectural layout design, *CAD Computer Aided Design*, 41(9), pp. 649–667.

Yan, X. and Han, J. (2002) gSpan: Graph-based substructure pattern mining, *Proceedings of the IEEE International Conference on Data Mining*, ICDM, pp. 721–724.

Zawiślak, S. and Rysiński, J. (2017) *Graph-Based Modelling in Engineering*. Springer, Cham.