Research on Self-adaptive Reinforcement Plug-in of Prefabricated Concrete Component based on BIM

Fubin Zhang[a], Yebo Shen[b], Li Tan[c], Ruoyu Jin[d,*], Yu Han[e]

[a]Associate Professor, Faculty of Civil Engineering and Mechanics, Jiangsu University, Zhenjiang, China.

ORCID ID: 0000-0003-4382-5830

[b]Assistant Teaching Professor, School of Civil Engineering, Nantong Institute of Technology, Nantong, China

ORCID ID: 0009-0005-7940-5987

[c]Graduate Research Assiatant, Faculty of Civil Engineering and Mechanics, Jiangsu University, Zhenjiang, China

ORCID ID: 0009-0007-9539-8486

[d]Reader, Department of Civil & Environmental Engineering, Brunel University London, Kingston Lane, Uxbridge, UB8 3PH, United Kingdom

ORCID ID: 0000-0003-0360-6967

[e] Professor, Faculty of Civil Engineering and Mechanics, Jiangsu University, Zhenjiang, China

ORCID ID: 0000-0002-5513-0393

*: Corresponding author, email: Ruoyu.Jin@brunel.ac.uk

Abstract

Prefabricated concrete structure has the advantages of faster construction, labor saving, pollution reduction, and enhanced quality. It has been more widely adopted in recent years, but the cost is usually higher at its design or pre-construction stages due to component disassembly and detailed design. Building information modeling (BIM) could improve the design efficiency and reduce design cost, hence promoting the development of prefabricated buildings. However, due to the complexity of reinforcement modeling, existing BIM authoring tools still have low operation efficiency. Even though the latest technologies can achieve rapid reinforcement modeling, it often fails to realize the adaptive adjustment of reinforcement. In the case of changes in prefabricated components, the internal

---

reinforcement needs to be remodeled, with reduced efficiency and increased the risk of errors. Therefore, this research proposes a self-adaptive reinforcement plug-in based on Autodesk Revit for prefabricated concrete components. The research achieved fast modeling and adaptive reinforcement. Firstly, the parametric modeling of column, beam and slab components was completed by using Excel to drive family parameters. Then, by adopting the Revit Application Program Interface (API), programming language C# and programming platform Visual Studio (VS), the secondary development within Revit achieved the rapid adaptive configuration of reinforcement. At the same time, the user interface was developed based on Windows Presentation Foundation (WPF), and the Ribbon function was adopted to expand the Revit function area to realize the visual regulation of key parameters of reinforcement, such as the reinforced type, cover thickness, spacing, and reinforcement ratio, etc. Finally, the program was developed by integrating different modules and plug-ins were established. A typical prefabricated frame structure office building was used as a case study to test the self-adaptive reinforcement plug-in. The results showed that the modeling efficiency of the developed plug-in was nearly 3 times higher than that of manual modeling. After changing the section size parameters of the members, the reinforcement could successfully achieve self-adaptive adjustment, hence significantly saving the modeling steps and time. The developed self-adaptive reinforcement plug-in contributed to the fully interoperable and multi-disciplinary coordination in prefabricated building design.

Keywords: Prefabricated buildings; Building information modeling; Secondary development; Self-adaptive reinforcement

1. Introduction

Prefabrication, as the emerging research and professional practices of the construction industry in recent years, has the advantages of enhancing quality, improving efficiency, reducing pollution and emissions, and potential for even wider applications (Liu et al. 2019a; Silva et al. 2013; Wu 2021; Li et al. 2022; Zhang et al. 2023; Zhang et al. 2024). Many researchers have analyzed and compared practical cast-in-situ buildings and prefabricated buildings from different perspectives such as safety, economy, society and industrial chain, and generally concluded that prefabricated buildings are superior to traditional cast-in-place buildings (Masood et al. 2022; Dong et al. 2015; Breccolotti et al. 2016; Ferrara et al. 2004; Li et al. 2014). However, the cost of prefabricated building is always one of the key issues restricting the development of prefabricated buildings (Navaratnam et al. 2019; Du et al. 2021; Hong et al. 2018; Lou et al. 2020). For prefabricated buildings, the main processes include: structural design, component production and site assembly. The structural design can be broken down into concept design, preliminary design, construction drawing development and detailed design. Compared to the cast-in-situ building, the prefabricated building design not only increases the workload of component design, but also changes the design concept, process and method accordingly (Xiao et al. 2021). Therefore, The degree of design standardization has become the most important factor affecting the cost of prefabricated buildings (Lou et al. 2020).

The concept of BIM was first proposed by Eastman (Wang 2012; Eastman et al. 2009) in 1975, with the purpose of facilitating the visualization and quantitative analysis of construction projects, reducing the cost of construction process, and acting on the whole life cycle of the project. In order to address the cost problem, many researchers have attempted to integrate prefabricated buildings with BIM to improve the level of standardization and industrialization of components (Tan et al. 2019; Li et al. 2018; Yoo et al. 2019). In prefabricated buildings, BIM can be used for standardized design, and then the design information in BIM is directly imported into the central control system of the factory and converted into production data information that can be read for producing components. BIM, which is deeply integrated with computers, has promising potentials to promote the development of prefabricated buildings (Mostafa et al. 2018; Li et al. 2017; Liu et al. 2019b; Tan et al. 2020; Bortolini et al. 2019).

However, the efficiency of BIM in prefabricated building design has not been fully achieved, such as reinforcement configuration in the detailed design. Manual efforts are still largely required to draw the reinforcement model in the BIM interface and to define different types of settings. The complexity of software operation leads to low design efficiency and high error rate, and thus increasing the cost of labor and time in design (Mangal et al. 2018). Engineers or designer could often face the problem of incorrect splitting and adjustment in the process of detailed design of prefabricated buildings. For instance, when the section size of components changes, the reinforcement cannot change adaptively with the changes of components, and hence the design efficiency is still low.

In recent years, in order to meet the practical needs of prefabricated construction, many scholars have conducted secondary development research on rapid remodeling based on different BIM tools (e.g., Autodesk Revit, CATIA, AutoCAD, etc.), including reinforcement modeling design to improve efficiency and to reduce costs (Liu et al. 2018; Huang et al. 2012; Lu 2012; Kamat et al. 2002; Bai et al. 2019; Yuan et al. 2018; Liu 2021). As the most typical BIM authoring tool, Autodesk Revit provides a large number of open features to meet users' personalized needs, and has significant advantages and applications for secondary development. There are also several studies (Mangal et al. 2018; Liu et al. 2021; Borges 2018) adopting algorithms to optimize the design and establish models of reinforcement, such as hybrid genetic algorithm and decomposed optimization algorithm, so that efficiency and accuracy of reinforcement detailed design are enhanced. While these previous studies have explored various methods of rapid reinforcement placement, the adaptive function of reinforcement changing with components is still lacking. The problem of repeated manual operation and consuming more manpower and time due to the change of component size during the detailed design of prefabricated building have not been fully addressed. Currently there have been insufficient studies on the adaptive reinforcement modeling of prefabricated building components. This research aims to develop a self-adaptive reinforcement platform for prefabricated concrete structure through secondary development. Adopting Autodesk Revit, a developed plug-in is used to realize fast modeling and adaptive reinforcement of basic components for prefabricated concrete frame structure. This study contributes to the improvement of existing adaptive reinforcement system in BIM, and improves the interoperable and multi-disciplinary coordination in prefabricated building design.

This paper is structured as follows. In Section 2, a literature review is presented with research objectives. Section 3 presents methodology including parametric modeling and secondary development. Section 4 exhibits the development process of the developed self-adaptive reinforcement plug-in, inclusive of covering the creation and optimization of the reinforcement, and the development of the window interface. In Section 5, the validity of the plug-in is demonstrated by analyzing the efficiency of manual modeling and plug-in modeling of a real-life project. Conclusions and future work are introduced in Section 6.

2. Literature review

Numerous studies have confirmed that cost is one of the key problems restricting the development of prefabricated buildings. Navaratnam et al. (2019) evaluated the performance of the prefabricated building system in Australia and concluded that the lack of structural design standards and the variety of components lead to increased costs. Du et al. (2021) studied prefabricated buildings from the perspective of supply chain, and claimed that prefabricated buildings had the disadvantage of high component cost. Hong et al. (2018) explored the cost of prefabricated buildings and summarized its influencing factors, including component varieties, design standardization, and integrated management, etc. Lou et al. (2020) applied system dynamics approach to analyze the key influencing factors of cost in the whole process of prefabricated buildings from design to construction, and concluded that the degree of design standardization was the most important factor affecting the cost of prefabricated buildings.

BIM is a collaborative platform for storing information and data exchange of construction projects. There have been successful applications of BIM authoring tools in the workflow of prefabricated building to reduce construction cost (Li et al. 2021; Wang et al. 2020; Wang 2019). For example, based on visualization and parameterization in BIM, Li et al. (2021) used a BIM based design software (i.e., PKPM-PC) for prefabricated building to design and optimize precast shear wall joints, and improved the efficiency of shear wall joint design. Wang et al. (2020) emphasized the significance of BIM for the efficient, accurate and reasonable design of prefabricated building structures, and proposed the importance of establishing a standardized component family library to improve the design efficiency. Wang (2019) adopted BIM to deepen the design of prefabricated buildings, established the node model of prefabricated building steel structure, reasonably embed pipelines and set openings, improved the construction quality and technical level of the project, and reduced the cost of human resources.

However, due to the complexity of real-world projects, existing BIM functions can not fully meet the needs. Several studies targeting on expanding the functions of BIM authoring tools through secondary development to solve the problem of the lack of personalized function. Tang et al. (2020) used the functions of Revit and ABAQUS to propose a data conversion interface for both three-dimensional (3D) visual modeling and structural calculation of asphalt pavements, and that effectively enhanced the structural analysis capability of BIM. PÄRN et al. (2017) studied the method to minimize the cost of updating and maintaining BIM model for the facility management team based on Revit Application Program Interface (API). Ma et al. (2019) carried out secondary development of Revit, collected information of components with Radio Frequency Identification (RFID) tags and transmitted it to BIM model, thus solving the problems of easy omission and query difficulties of precast concrete building components. Liu et al. (2018) proposed a rule-based BIM methodology and developed a prototype system through API on the basis of Revit. Huang et al. (2012) derived the end-face equation of double circular-arc gears, and conducted secondary development of CATIA using Visual Basic language programming, which quickly achieved parametric modeling of components.

Targeting secondary development of reinforcement rapid modeling for the detailed design of prefabricated buildings, Bai et al. (2019) introduced the methodology of establishing local parametric component database and developed plug-in through Revit API. This methodology avoided the tedious modeling process of Revit's original reinforcement modeling tools, but the plug-in did not consider the possible model adjustment problems that may occur in the subsequent design changes. Yuan et al. (2018) also proposed the method of establishing a standard parameterized prefabricated component library through secondary development and family templates. The secondary development focused on meeting the requirements of manufacturing and assembly, and reflected in node connection and collision detection optimization. Liu (2021) developed a plug-in based on Autodesk Robot Structure Analysis that could automatically read the internal forces of structural units of prefabricated subway station under various working conditions and perform reinforcement design and calculation. This plug-in improved the efficiency of reinforcement design. However, there was still a lack of research on how to cope with model adjustment after its establishment.

Mangal et al. (2018) presented a BIM-based framework using the developed three-stage hybrid genetic algorithm (GA) for automated optimization of steel reinforcement, and considered the combination of reinforcement with different diameters to calculate the reinforcement design with the minimum reinforcement area. Liu et al. (2021) proposed an automatic and optimal rebar layout method based on decomposition optimization algorithm. This method could save more than 60% of the computation time compared to other algorithms. Borges (2018) introduced the slab related data into the developed Excel worksheet to support the reinforcement design, and showed the selected reinforcement area in Revit through the visual programming software Dynamo. Even though these studies improved the efficiency of single reinforcement design and modeling, it had not taken into account the behavior of continuous modification and repeated times due to incorrect component splitting during detailed design of prefabricated buildings. These existing results could not achieve the self-adaptive adjustment of the reinforcement after the change of the component. Therefore, it is meaningful to study the adaptive reinforcement of prefabricated concrete components.

3. Methodology

3.1 Parametric modeling

There are various types of components in design and construction practices. If the modeling is conducted according to the shape and size of the cross-section, the workload is huge and the cross-section drawing error is easy to occur. However, based on the standard family provided by Autodesk Revit, parametric design can be modified to automatically generate family instance models, hence improving the efficiency and accuracy of modeling.

Each prefabricated component is created according to the corresponding family template file. After designing the parameters of the prefabricated component family, it is necessary to associate them with the corresponding family parameters, in order to drive the design parameters to adjust the dimensions, materials and other attributes of the prefabricated component family. After the column, beam and slab prefabricated components are created, geometric constraints are carried out to ensure that components are symmetrical to the central axis no matter how the parameters changed. It is important to label the dimensions of each component and add parameters, which are bound to the dimension annotations. Through continuous debugging, parameters are ensured to drive the BIM-based model to undergo adaptive changes.

However, if the parameters of each component are operated manually, it will cause a waste of human resources. There are two ways to drive family parameters, namely data file driver and data built-in driver. The data built-in driver has stronger operability, but it is not conducive to manage parameter information, and it is relatively difficult for users to debug parameters, making it more suitable for model creation work with fewer family types. When there are multiple types of families, the approach of data file driver is widely used. Therefore, this paper used Excel worksheet file to design and manage the relevant parameters of prefabricated component families, and automatically identified the data in the tables through Revit, in order to generate family instance models in batch. The process of batch generation of family instances starting from Excel sheet is shown in Figure 1.

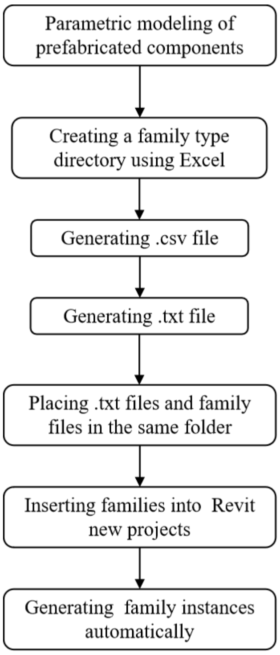


Fig.1. Batch generation family instance flow chart

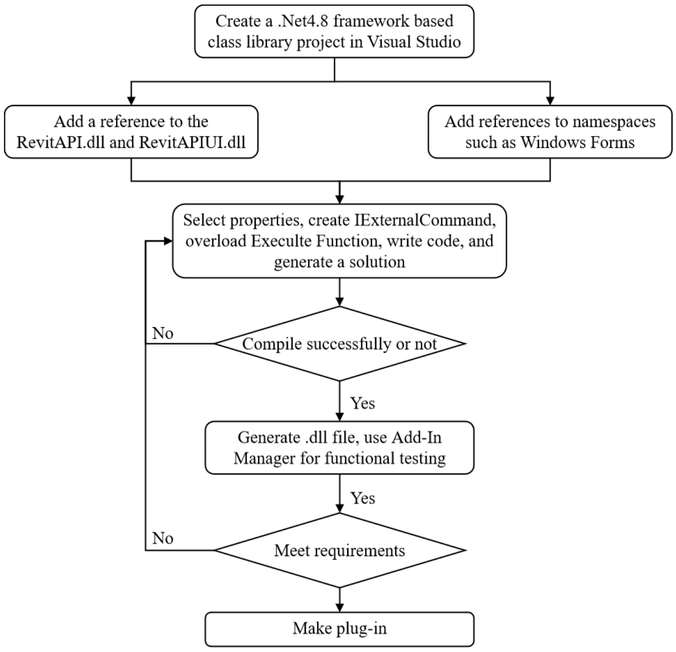


Fig.2. Secondary development flow chart

3.2 Revit API and the secondary development

Autodesk Revit is the world's most widely used BIM authoring tool. In order to meet the personalized needs of users, it provides a large number of API functions to facilitate users to develop plug-ins according to the actual needs of real-life projects. Since Revit itself is developed based on the C#.Net language framework, C# was chosen as the programming language for this study.

In this research, Revit 2019 was selected as the basic tool for the study, and the method functions in the Revit API class library were used to extend and enhance Revit. Tests had found that at least .Net Framework 4.7 was needed for the secondary development of Revit 2019. Therefore, the comprehensive .Net Framework 4.8 language framework was selected for compiling the program.

Visual Studio 2019, which contained the .Net Framework 4.8, was used as the development software.

In Visual Studio 2019, Revit can be expanded only by accessing Revit documents according to two special interfaces, namely IExternalCommand and IExternalApplication, where IExternalCommand is used more frequently in practice. In this study, IExternalCommand was used to compile the code of the required function, and Add-In Manager plug-in was used to run the program to be implemented. The main flow of secondary development is shown in Figure 2.

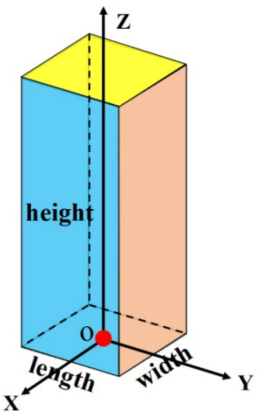
4. Development of self-adaptive reinforcement plug-in

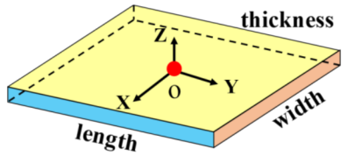
4.1 Creation and optimization of reinforcement

4.1.1 Reading of component parameters

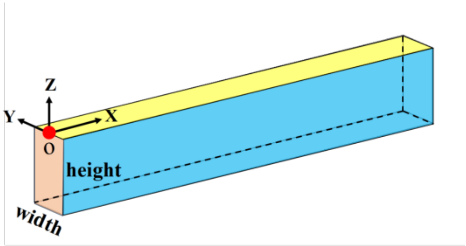
The first step in modeling reinforcement in Revit is to identify the component that needs to be laid out, i.e. the host of the reinforcement model, so that the geometry, dimensions, coordinates, materials and other parameters of the host can be read. LookupParameter function can directly obtain the geometric dimensions of components, but fail to obtain the positioning control points. Although BoundingBoxXYZ function can obtain both, it must be calculated by algorithm and occupy memory. In order to occupy as little memory as possible, LookupParameter function is preferred to obtain the length, width and height parameters of the selected component, and BoundingBoxXYZ function is combined to obtain the control point.

In this study, the centre position of the bottom plane of the column and the centre position of the upper plane of the slab were registered as the anchor points of the column and slab respectively (see Figure 3(a), (b)). Meanwhile, the position line of the beam was usually referred to the center line of the upper surface of beam in Revit, so the anchor point of the beam was set at the beam end of the center line of the upper surface (see Figure 3(c)). In order to ensure that the reinforcement moved spontaneously with the movement of the host element, the relative system was used to describe the anchor point of the reinforcement.





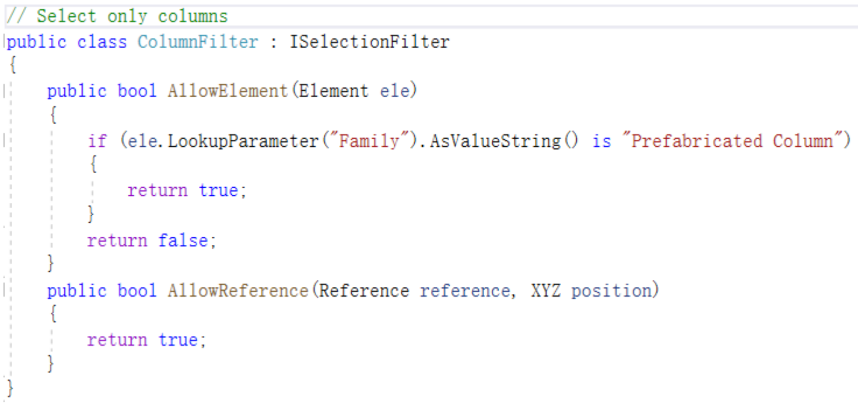
(b) Anchor point of prefabricated slab



(a) Anchor point of prefabricated column (c) Anchor point of prefabricated beam

Fig.3. Anchor points of prefabricated components

Due to the wide variety of components, it would consume significant effort to click on each one to select the specified type of components for reinforcement, so the API filtering was used to quickly acquire the same type of components by box selection. The parameter name and type of the specified component were obtained by Revit Lookup plug-in. After the program was traversed and searched, two functions mentioned above were used to read the length, width and advanced parameters of the selected component to create data structures. The core code of traversal and selection is shown in Figure 4. Due to the uncertainty of the normal vector of the beam section, the spatial coordinates of the beam needed to be converted into relative coordinates to ensure that the reinforcement changes with the control line of the beam.



```
// Select only columns
public class ColumnFilter : ISelectionFilter
{
    public bool AllowElement(Element ele)
    {
        if (ele.LookupParameter("Family").AsValueString() is "Prefabricated Column")
        {
            return true;
        }
        return false;
    }
    public bool AllowReference(Reference reference, XYZ position)
    {
        return true;
    }
}
```

Fig.4. The core code of traversal and selection

4.1.2 Reinforcement positioning rule

For the reinforcement in the column, it was necessary to create stirrup and through-length bars in the column, including the through-length bars on both sides of the column B side and H side. The horizontal right of the column section was the positive direction of X axis, and the vertical upward was the positive direction of Y axis. The section dimensions of the host element were b and h. The section center point was the origin. The thickness of the protective layer was c. The diameter of the stirrup was denoted as d_g, the distance between the center of the stirrup and the edge of the column was $d = c + d_g/2$, and the distance between the stirrup at both ends of the column and the column end was also $d = c + d_g/2$. A stirrup in Revit was represented as a quadrilateral with radian angles, and a stirrup could be described by four points as shown in Figure 5. The control points of the sample reinforcement were connected in pairs to form the IList<Curve>. The rebar style was set to RebarStyle.Standard, and the bending angle of the start and end hooks was set to 135 degrees.

The through-length bar of the column depicted the shape of a single steel bar with two points 1 and 2 as shown in Figure 6, and the curve of the sample reinforcement was the curve set formed by the connection of the above two points. The type and arrangement of through-length bars on the B side and H side of the column were distinguished. A through-length bar was positioned at each of the four corners of the column to facilitate the subsequent creation of through-length bars and self-adaptive change of reinforcement as shown in Figure 7 through correlation functions. The coordinates of the positioning control points of the four through-length bars are shown in Table 1.

The performance form and positioning mode of beam and slab reinforcement were similar to that of column. The reinforcement in the beam included stirrup, upper tensile reinforcement, lower compression reinforcement and waist reinforcement. The shape of stirrup and through-length bar was set as 01 and 33 respectively. The waist reinforcement type was set as 10 HPB300 and positioned in the middle of the H side of the beam. The reinforcement in the slab included longitudinal bars and transverse bars, which were located in the appropriate position through the functional relation. The upper and lower longitudinal and transverse bars of the slab were symmetrical.

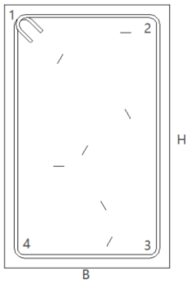


Fig.5. Stirrup sample

Fig.6. Through-length bar sample

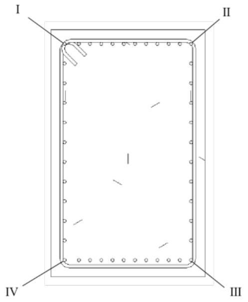


Fig.7. Positioning of through-length bar

Table 1. Coordinates of column longitudinal bar control points

Number	Control points	X coordinate	Y coordinate	Z coordinate
I	1	$-b / 2 + d_B$	$h / 2 - d_H$	L
	2	$-b / 2 + d_B$	$h / 2 - d_H$	0
II	1	$b / 2 - d_B$	$h / 2 - d_H$	L
	2	$b / 2 - d_B$	$h / 2 - d_H$	0
III	1	$b / 2 - d_B$	$-h / 2 + d_H$	L
	2	$b / 2 - d_B$	$-h / 2 + d_H$	0
IV	1	$-b / 2 + d_B$	$-h / 2 + d_H$	L
	2	$-b / 2 + d_B$	$-h / 2 + d_H$	0

b means B-side dimension of column section; d_B means distance between longitudinal bar center and side H column edge, $d_B=c+d_g+d_b/2$; d_H means distance between longitudinal bar center and B side column edge, $d_H=c+d_g+d_h/2$; Among them, c means thickness of protective layer, d_g means stirrup diameter, d_b means diameter of longitudinal reinforcement at side B, d_h means diameter of longitudinal reinforcement at side H; L means column height.

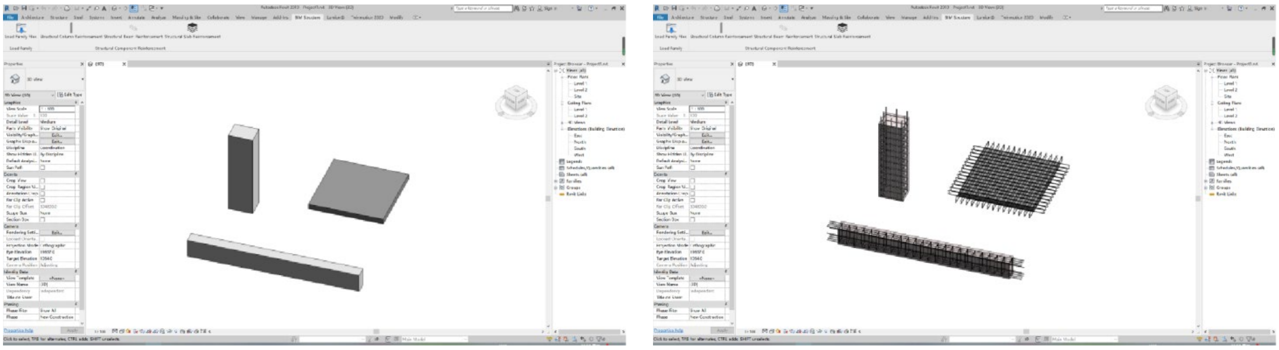
4.1.3 Creation and adaptive adjustment of reinforcement

There are three functional approaches to rebar creation in Revit: “Rebar.CreateFromCurves”, “Rebar.CreateFromRebarShape” and “Rebar.CreateFromCurvesAndShape”. “Rebar.CreateFromRebarShape” and “Rebar.CreateFromCurvesAndShape” are two rebar creation methods used to create rebar shapes and families that come with Revit, and to create custom shaped rebar based on the rebar family that comes with Revit, respectively. The “Rebar. CreateFromCurves method” used in this study was suitable for creating custom shaped rebar, which matched the requirements of this research and was easy to operate and to be widely used. The reinforcement style and hook angle are set by the ”RebarStyle” and “RebarHookType” parameters in the “Rebar. CreateFromCurves” function, respectively. After reading the component parameters and understanding the relevant key points of steel flat method and positioning rules, the parameters of reinforcement are described by Boolean operation, so as to form the parametric model of sample reinforcement. The function expression for the first method is public static Rebar CreateFromCurves(). Using the column stirrup as an example, the core code is as follows:

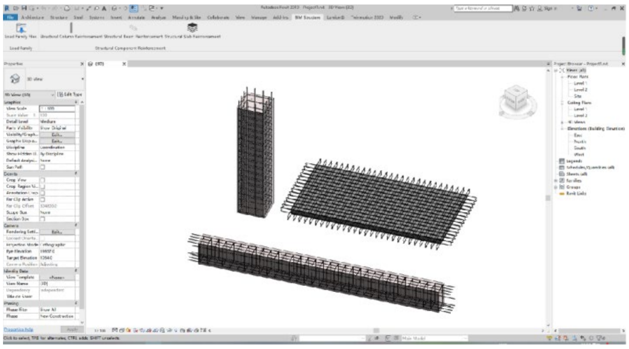
```
//Single stirrup creation
Rebar rbGJ = Rebar.CreateFromCurves(doc, RebarStyle.Standard, rbTypeGJ, hook, hook, column, norm1, curves, RebarHookOrientation.Right, RebarHookOrientation.Right, true, true);
```

The stirrup spacing of each component is different. When creating stirrups in batch, stirrup spacing should be referenced and entered in the form. In order to adapt the stirrup to fit with the change of the host elements, this study used the SetLayoutAsMaximumSpacing class function statement to construct the reinforcement set and used GetShapeDrivenAccessor() to drive. The quick reinforcement results and adaptive results of prefabricated components are shown in Figure 8. The core code is as follows:

```
//stirrup spacing
double space = Convert.ToDouble(W2.textbox2.Text) / 304.8;
//Stirrup batch creation and adaptive adjustment
rbGJ.GetShapeDrivenAccessor().SetLayoutAsMaximumSpacing(space, height - 2 * protect -
rbTypeGJ.BarDiameter, true, true, true);
```



(a) Parametric model of prefabricated components (b) Reinforcement model of prefabricated components



(c) Adaptive adjustment of prefabricated component reinforcement

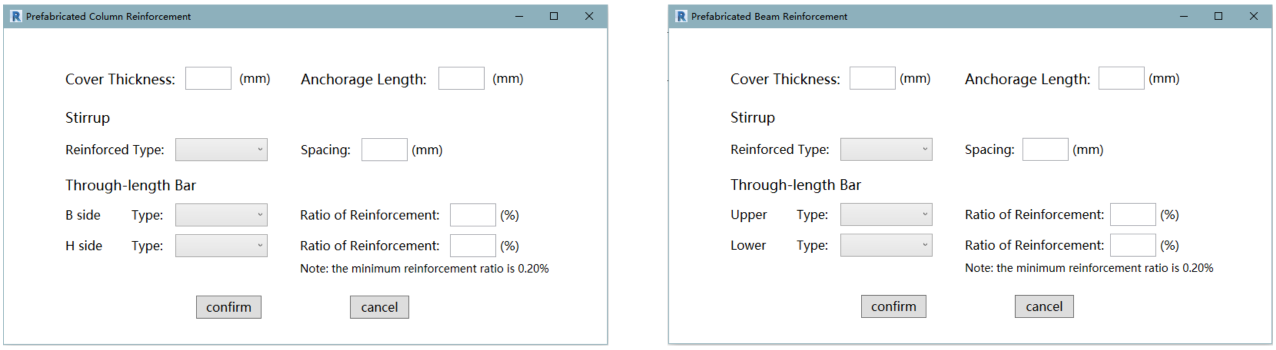
Fig.8. Self-adaptive reinforcement plug-in of prefabricated components display

4.2 Development and integration of window interface

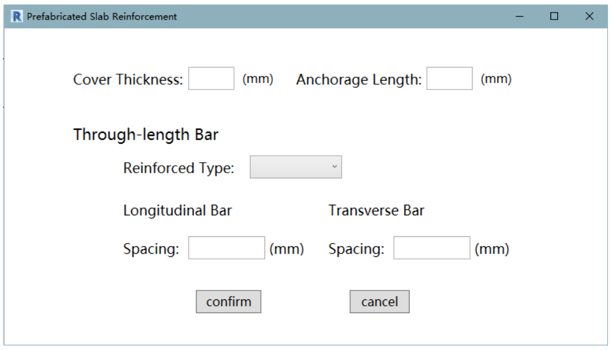
4.2.1 Development of window interface

The Visual Studio family offers two forms applications: WinForm and Windows presentation foundation (WPF). Compared with WinForm, WPF is more complex and difficult to learn, but it has incomparable advantages: WPF's UI is more brief, intuitive and extensible; Both in terms of interface layout and form adaptation, WPF's control adaptation is significantly more effective than that of WinForm's. Interface display and code will be better separated, easier for developers to code. Therefore, the WPF framework was chosen for the development of the forms application.

The size, position and formatting of the “Title”, “Label”, “TextBox” and “ComboBox” needed to be defined exactly when the form was developed. Then the “OK” and “Cancel” as two Button complete forms were set. The parameters entered in the form interface included the reinforced type, cover thickness, spacing, reinforcement ratio and anchorage length. Prefabricated column, beam and slab reinforcement window application is shown in Figure 9.



(a) Prefabricated column reinforcement form application (b) Prefabricated beam reinforcement form application



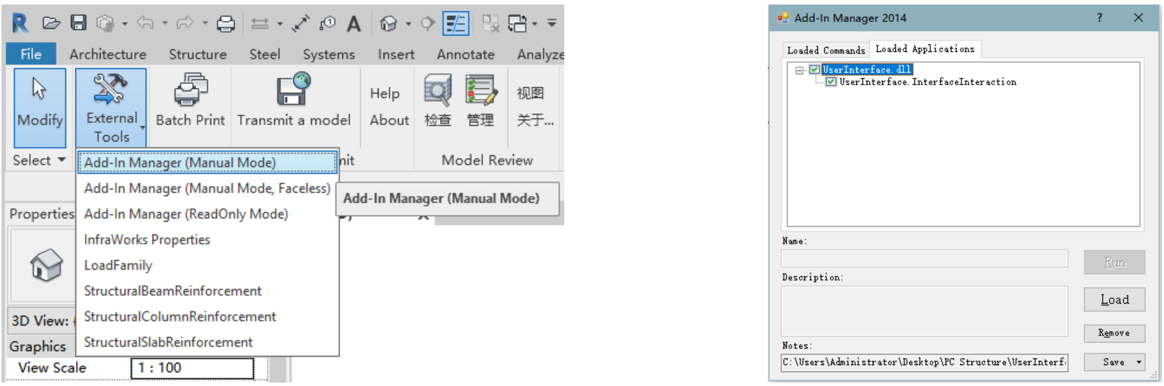
(c) Prefabricated slab reinforcement form application

Fig.9. Prefabricated component reinforcement form application

4.2.2 Ribbon user interface (UI) extension

After writing and testing the reinforcement codes for prefabricated columns, beams and slabs, the final program was aggregated to expand the Ribbon UI. The interface expansion process is as follows. First of all, a “RibbonTab” must be created. Then, one or more “RibbonPanel” must be added to the created “RibbonTab” according to actual requirements. Then, button controls of function commands could be arranged in the corresponding panels. Finally, the external commands were defined using ToolTip and LargeImage functions and the control buttons were displayed. Revit provided a special API to allow users to customize the UI interface. To create a UI, “RevitAPIUI.dll” file needed to be referenced.

The code was regenerated after compilation, and the generated .dll file was loaded through the plug-in Add-In Manager, as shown in Figure 10. Locate the .addin file generated by external applications under the internal .addin file directory of Revit and copy it to the corresponding Revit version. After Revit was restarted, the extended interface appeared (see Figure 11).



(a) Add external command dialog box (b) External commands load and run plug-in

Fig.10. Loading external Applications

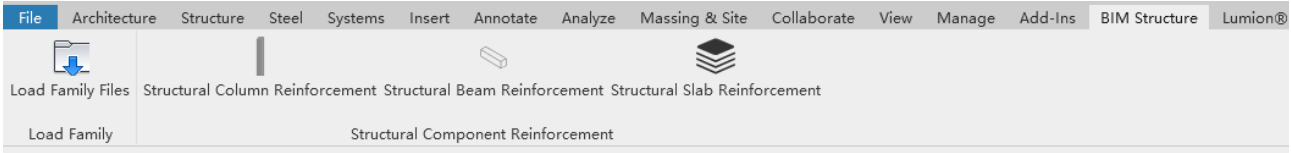


Fig.11. Ribbon interface

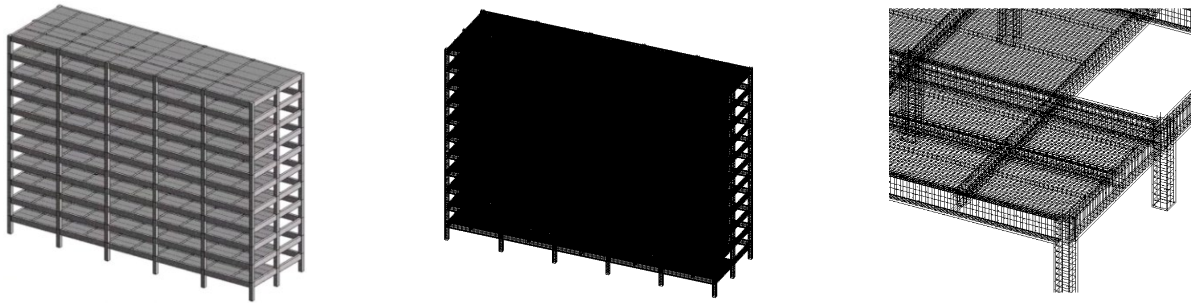
5. Validation and discussion

The proposed self-adaptive reinforcement plug-in is developed to improve the efficiency of reinforcement modeling of prefabricated concrete components and reduce labor and time costs. The research has realized the rapid creation and adaptive adjustment of the reinforcement model of prefabricated concrete column, beam and slab structural members. To validate the developed methodology, it was applied to a real-life project.

5.1 Case study and results

An office building with eleven-story prefabricated frame structure was selected for case study. The seismic fortification intensity of the project is Magnitude 7, and the site category is ii. During the engineering design, parametric families were used to model the columns, beams and slabs. The frame structure model is shown in Figure 12 (a). The reinforcement was then configured using the proposed platform to enable the creation and adaptive adjustment of the reinforcement by entering parameters that control the reinforcement model in the WPF form interface. The integral and partial reinforcement model are shown in Figure 12 (b), (c).

In order to verify the advantages of the plug-in, manual modeling and plug-in automatic modeling efficiency were compared. Six peer assessors proficient in Revit, equipped with high-performance computers, were recruited to record the modeling and reinforcement time of individual components (columns as an example) and the whole building model of the frame-structured office building project. In addition, the self-adaptive test of reinforcement after component size modification was carried out on the plug-in.



(a) Frame structure model (b) Integral reinforcement model (c) Partial reinforcement model

Fig.12. Project Revit model

Table 2 shows the time and efficiency comparison results of manual modeling and self-adaptive reinforcement plug-in modeling for single column of the frame structure office building. As shown in the table, the average time of using manual modeling was 288s, while the average time of using self-adaptive reinforcement plug-in was greatly shortened to 86s. The efficiency of modeling using the developed platform was significantly improved, by an average of 2.3 times, with assessor No.1 and No.2 showing the highest improvement by 2.6 times.

Table 3 shows the results of manual modeling and self-adaptive reinforcement plug-in modeling for the whole frame structure office building. Due to the large number of building-wide model components, the required column, beam and slab modeling time accounted for a large proportion of the time. The overall modelling uplift efficiency was not as large as for the individual components, but it was still a significant improvement, with a maximum uplift efficiency of 3.5 times and an average uplift efficiency of 2.8 times. Through the self-adaptive reinforcement plug-in, all assessors could complete the modeling of the whole building in about 32 minutes. In general, the use of secondary development techniques for adaptive modeling of reinforcement could significantly enhance automation, thus improving efficiency, saving time and cost, and avoiding the waste of human resources.

Table 2. Individual component test results

Tester	Time (s)		Improved efficiency (times)
	Manual modeling	Plug-in modeling	
No.1	305	85	2.6
No.2	285	80	2.6
No.3	260	83	2.1
No.4	275	90	2.1
No.5	293	88	2.3
No.6	312	91	2.4
Mean value	288	86	2.3

Table 3. The whole building test results

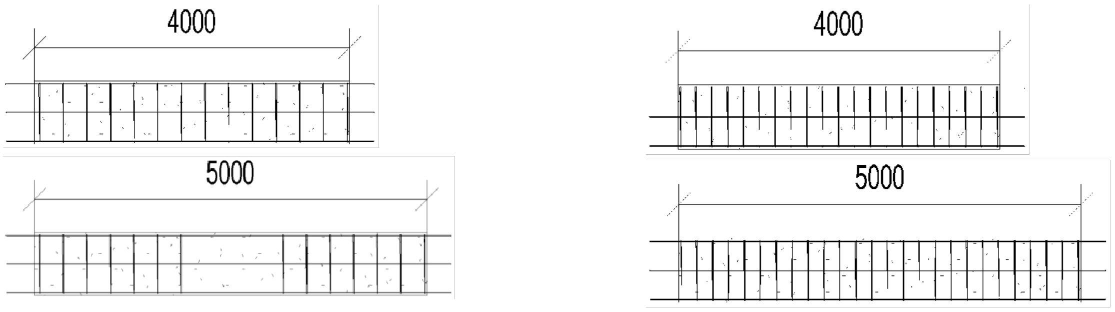
Tester	Time (s)		Improved efficiency (times)
	Manual modeling	Plug-in modeling	
No.1	118	30	2.9
No.2	128	35	2.7
No.3	104	23	3.5
No.4	125	35	2.6
No.5	110	26	3.2
No.6	135	40	2.4
Mean value	120	32	2.8

5.2 Comparison of reinforcement effects after adjustment

In order to modify the size of prefabricated building components and realize the purpose of automatically adjusting the internal reinforcement of components, the platform was developed. In order to test the value of the methodology proposed in this research, a single structural column, beam and slab in the eleven-story prefabricated frame structure office building was randomly selected for the adaptive adjustment test of the reinforcement model. Figure 13~15 show and compare the test process of reinforcement between self-adaptive adjustment and manual reinforcement and plug-in reinforcements. The results showed that by changing the height attribute of precast columns, the length of precast beams, and the width of precast slabs, the reinforcement model of prefabricated concrete components manually created could not be automatically adjusted with the change of size, while the component model created by program could perform adaptive adjustment. Due to the constraints of project location, production standards, construction conditions and other factors, prefabricated components in real-world projects will inevitably have a variety of specifications, and are prone to parameter changes and other problems. Therefore, the research results of this study demonstrated improvements of the efficiency of assembly component modeling to a certain extent, with further potential to reduce labor and save resources.

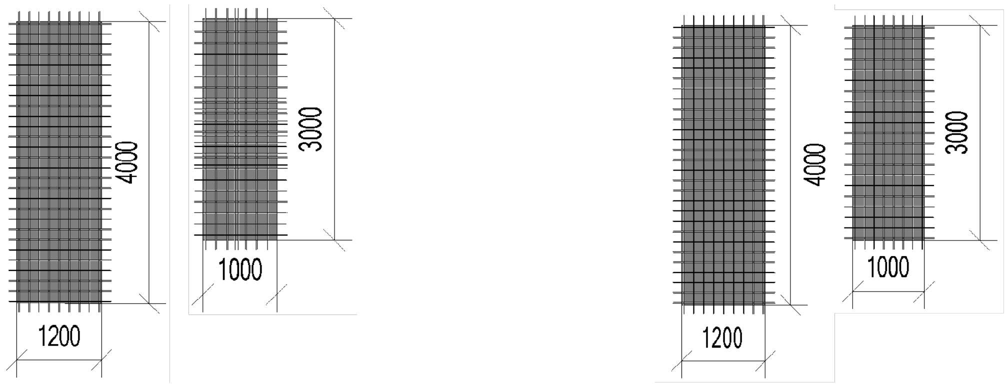
(a) Manual modeling adaptive adjustment test (b) Plug-in modeling adaptive adjustment test

Fig.13. Project structural column reinforcement adaptive test



(a) Manual modeling adaptive adjustment test (b) Plug-in modeling adaptive adjustment test

Fig.14. Project structural beam reinforcement adaptive test



(a) Manual modeling adaptive adjustment test (b) Plug-in modeling adaptive adjustment test

Fig.15. Project structural slab reinforcement adaptive test

6. Conclusion and Perspective

This study focused on prefabricated frame structure building and developed a self-adaptive reinforcement methodology based on BIM technology. The self-adaptive reinforcement platform was started by driving parameters to generate column, beam and slab instance. It completed the self-adaptive configuration of internal reinforcement through secondary development program. The self-adaptive reinforcement platform was tested using a typical prefabricated frame structure office building project. The main contribution of the presented research are as follows:

(1) Based on Revit component family, this study proposed a method to create parameterized family model, and used Excel table to drive family parameters to generate component models in batches in Revit. The method could reduce the repetitive creation of prefabricated component models in prefabricated structural buildings and greatly reduce the error rate.

(2) Based on the relevant functions of Revit API, this study traversed all framed components, filtered out the required components, and read the parameter information of the selected components. The method was efficient for the rapid reinforcement of specified types of components and avoided the waste of time caused by manual fixing of each individual component.

(3) This study used the C#.Net programming language and the Rebar function provided by Revit API to create the reinforcement model through IExternalCommand. The API class library functions and algorithms were also used to directly realize the adaptive adjustment of reinforcement.

(4) Through the WPF window interface development function, this study realized the visual control of the parameters of prefabricated components by considering the key parameters such as cover thickness, anchorage length, reinforced spacing and so on. The Ribbon function was used to expand

Revit functional areas. The compiled program was integrated to generate the self-adaptive reinforcement plug-in platform of prefabricated concrete component.

(5) Through the practical case, the proposed self-adaptive reinforcement plug-in was verified to be of potential use in saving modeling time and improving modeling efficiency. It was with practical value in BIM-based design and can provide extended insights for subsequent research on secondary development of Revit.

(6) Addressing the immature application of Revit secondary development technology in many developing countries and the scarcity of open source data, this study is a preliminary exploration of the application of Revit API for prefabricated buildings. The reinforcement parameters considered in the plug-in only include some typical parameters, while more complex parameters such as reinforcement anchoring form and densification area are not involved. Moreover, the prefabricated components have various structural forms, especially specially-shaped components in real world. Therefore, the parameterized modeling methods for prefabricated components and spatial algorithms of reinforcement layout still need to be explored and improved, and more types of engineering cases need to be tested to verify the effectiveness of the proposed plug-in.

Acknowledgement

The research described here was sponsored by the Program of China Postdoctoral Science Foundation (Grant No.2021M690264 and 2021T140031), the State Key Laboratory of Special Functional Waterproof Materials (Grant No.SKWL-2021KF10), and the Youth Talent Cultivation Program of Jiangsu University

References

Bai QH, Deng SH, Li CG, Qie Z (2019) Application of Bim in the Creation of Prefabricated Structures Local Parameterized Component Database. *Architecture and Engineering* 4(2): 13-21, DOI: 10.23968/2500-0055-2019-4-2-13-21

Borges MM (2018) BIM modelling automation on reinforcement detailing of slabs.

Bortolini R, Formoso CT, Viana DD (2019) Site logistics planning and control for engineer-to-order prefabricated building systems using BIM 4D modeling. *Automation in Construction* 98: 248-264, DOI: 10.1016/j.autcon.2018.11.031

Breccolotti M, Gentile S, Tommasini M, Materazzi AL, Bonfigli MF, Pasqualini B, Colone V, Gianesini M (2016) Beam-column joints in continuous RC frames: Comparison between cast-in-situ and precast solutions. *Engineering Structures* 127: 129-144, DOI: 10.1016/j.engstruct.2016.08.018

Dong YH, Jaillon L, Chu P, Poon CS (2015) Comparing carbon emissions of precast and cast-in-situ construction methods - A case study of high-rise private building. *Construction and Building Materials* 99: 39-53, DOI: 10.1016/j.conbuildmat.2015.08.145

Du Q, Pang QY, Bao TN, Guo XQ, Deng YG (2021) Critical factors influencing carbon emissions of prefabricated building supply chains in China. *Journal of Cleaner Production* 280: 124398, DOI: 10.1016/j.jclepro.2020.124398

Eastman C, Lee JM, Jeong YS, Lee JK (2009) Automatic rule-based checking of building designs. *Automation in Construction* 18(8): 1011-1033, DOI: 10.1016/j.autcon.2009.07.002

Ferrara L, Negro P, Toniolo G (2004) Precast vs. cast-in-situ reinforced concrete industrial buildings under earthquake loading: an assessment via pseudodynamic tests. *13th World Conference on*

Earthquake Engineering, *Vancouver, BC Canada, August* 1-6: 743.

Huang K, Chen WD, Chen Q, Li YM (2012) Research of Double Circular-Arc Gear Fast Modeling Based on CATIA Secondary Development. *2012 Third International Conference on Digital Manufacturing & Automation, Guilin, China, 31 July 02 August 2012*: 658-661, DOI: 10.1109/icdma.2012.156

Hong J, Shen GQ, Li Z, Zhang B, Zhang W (2018) Barriers to promoting prefabricated construction in China: A cost–benefit analysis. *Journal of Cleaner Production* 172: 649-660, DOI: 10.1016/j.jclepro.2017.10.171

Kamat VR, Martinez JC (2002) Scene graph and frame update algorithms for smooth and scalable 3D visualization of simulated construction operations, *Computer-Aided Civil and Infrastructure Engineering* 17(4): 228-245, DOI: 10.1111/1467-8667.00272

Li CZ, Xue F, Li X, Hong JK, Shen GQ (2018) An Internet of Things-enabled BIM platform for on-site assembly services in prefabricated construction. *Automation in Construction* 89: 146-161, DOI: 10.1016/j.autcon.2018.01.001

Li CZ, Zhong RY, Xue F, Xu GY, Chen K, Huang GG, Shen GQ (2017) Integrating RFID and BIM technologies for mitigating risks and improving schedule performance of prefabricated house construction. *Journal of Cleaner Production* 165: 1048-1062, DOI: 10.1016/j.jclepro.2017.07.156

Li W, Xiong L, Liu YJ, Li SJ (2021) Application research on the design of assembled shear wall joint based on BIM technology. *IOP Conference Series: Earth and Environmental Science* 787(1), DOI: 10.1088/1755-1315/787/1/012186

Li XJ, Wang C, Kassem MA, Alhajlah HH, Bimenyimana S (2022) Evaluation Method for Quality

Risks of Safety in Prefabricated Building Construction Using SEM-SDM Approach. *International Journal of Environmental Research and Public Health* 19(9): 5180, DOI: 10.3390/ijerph19095180

Li Z, Shen GQ, Xue X (2014) Critical review of the research on the management of prefabricated construction. *Habitat International* 43: 240-249, DOI: 10.1016/j.habitatint.2014.04.001

Liu GW, Gu TY, Xu PP, Hong JK, Shrestha A, Martek I (2019a) A production line-based carbon emission assessment model for prefabricated components in China. *Journal of Cleaner Production* 209: 30-39, DOI: 10.1016/j.jclepro.2018.10.172

Liu HX, Singh G, Lu M, Bouferguene A, Al-Hussein M (2018) BIM-based automated design and planning for boarding of light-frame residential buildings. *Automation in Construction* 89: 235-249, DOI: 10.1016/j.autcon.2018.02.001

Liu H, Sydora C, Altaf MS, Han S, Al-Hussein M (2019b) Towards sustainable construction: BIM-enabled design and planning of roof sheathing installation for prefabricated buildings. *Journal of Cleaner Production* 235: 1189-1201, DOI: 10.1016/j.jclepro.2019.07.055

Liu JP, Li S, Xu CR, Wu Z, Ao N, Chen YF (2021) Automatic and optimal rebar layout in reinforced concrete structure by decomposed optimization algorithms. *Automation in Construction* 126: 103655, DOI: 10.1016/j.autcon.2021.103655

Liu WJ (2021) The Application of Computer Bim Assisted Secondary Development Technology in The Structural Design of Prefabricated Subway Station. *Journal of Physics: Conference Series* 1744(2), DOI: 10.1088/1742-6596/1744/2/022077

Lou N, Guo JJ, Pesämaa O (2020) Study on Key Cost Drivers of Prefabricated Buildings Based on System Dynamics. *Advances in Civil Engineering* 2020: 1-12, DOI: 10.1155/2020/8896435

Lu MW (2012) The Application of CAD Secondary Development in 3D Parametric Graphic Design. *Applied Mechanics & Materials* 224: 239-242, DOI: 10.4028/www.scientific.net/AMM.224.239

Ma GF, Jiang J, Shang SS (2019) Visualization of Component Status Information of Prefabricated Concrete Building Based on Building Information Modeling and Radio Frequency Identification: A Case Study in China. *Advances in Civil Engineering* 2019: 1-13, DOI: 10.1155/2019/6870507

Mangal M, Cheng JCP (2018) Automated optimization of steel reinforcement in RC building frames using building information modeling and hybrid genetic algorithm. *Automation in Construction* 90: 39-57, DOI: 10.1016/j.autcon.2018.01.013

Masood R, Lim JBP, González VA, Roy K, Khan KIA (2022) A Systematic Review on Supply Chain Management in Prefabricated House-Building Research. *Buildings* 12(1): 40, DOI: 10.3390/buildings12010040

Mostafa S, Kim KP, Tam VWY, Rahnamayiezekavat P (2018) Exploring the status, benefits, barriers and opportunities of using BIM for advancing prefabrication practice. *International Journal of Construction Management* 20(2): 146-156, DOI: 10.1080/15623599.2018.1484555

Navaratnam S, Ngo T, Gunawardena T, Henderson D (2019) Performance Review of Prefabricated Building Systems and Future Research in Australia. *Buildings* 9(2): 38, DOI: 10.3390/buildings9020038

Pärn EA, Edwards DJ (2017) Conceptualising the FinDD API plug-in: A study of BIM-FM integration. *Automation in Construction* 80: 11-21, DOI: 10.1016/j.autcon.2017.03.015

Silva PCP, Almeida M, Bragança L, Mesquita V (2013) Development of prefabricated retrofit module towards nearly zero energy buildings. *Energy and Buildings* 56: 115-125, DOI: 10.1016/j.enbuild.2012.09.034

Tan T, Chen K, Xue F, Lu WS (2019) Barriers to Building Information Modeling (BIM) implementation in China's prefabricated construction: An interpretive structural modeling (ISM) approach. *Journal of Cleaner Production* 219: 949-959, DOI: 10.1016/j.jclepro.2019.02.141

Tan Y, Li S, Wang Q (2020) Automated Geometric Quality Inspection of Prefabricated Housing Units Using BIM and LiDAR. *Remote Sensing* 12(15): 2492, DOI: 10.3390/rs12152492

Tang FL, Ma T, Guan YS, Zhang ZX (2020) Parametric modeling and structure verification of asphalt pavement based on BIM-ABAQUS. *Automation in Construction* 111, DOI: 10.1016/j.autcon.2019.103066

Wang G (2019) Design and optimization of prefabricated component system based on BIM technology. *Journal of Physics: Conference Series* 1345(6): 062054, DOI: 10.1088/1742-6596/1345/6/062054

Wang HY, Zhang ZH (2020) Discussion on Prefabricated Concrete Structure Design Method under BIM Technology. *E3S Web of Conferences* 198, DOI: 10.1051/e3sconf/202019803011

Wang XY (2012) BIM Handbook: A guide to Building Information Modeling for owners, managers, designers, engineers and contractors. *Construction Economics and Building* 12(3): 101-102, DOI: 10.5130/AJCEB.v12i3.2749

Wu XH (2021) Wireless management system of prefabricated construction materials based on BIM technology. *EURASIP Journal on Wireless Communications and Networking* 2021(1), DOI: 10.1186/s13638-021-01955-x

Xiao YW, Bhola J (2021) Design and optimization of prefabricated building system based on BIM technology. *International Journal of System Assurance Engineering and Management* 13(S1): 111-120, DOI: 10.1007/s13198-021-01288-4

Yoo M, Kim J, Choi C (2019) Effects of BIM-Based Construction of Prefabricated Steel Framework from the Perspective of SMEs. *Applied Sciences* 9(9): 1732, DOI: 10.3390/app9091732

Yuan ZM, Sun CS, Wang YW (2018) Design for Manufacture and Assembly-oriented parametric design of prefabricated buildings. *Automation in Construction* 88: 13-22, DOI: 10.1016/j.autcon.2017.12.021

Zhang FB, Luo H, Xiao JZ, Singh A, Xu J, Fang H (2023) Axial compressive behavior of GFRP-timber-reinforced concrete composite columns. *Low-carbon Materials and Green Construction* 1(5):1-13, DOI: 10.1007/s44242-022-00007-x

Zhang FB, Lu ZY, Wang DC, Fang H (2024) Mechanical properties of the composite sandwich structures with cold formed profiled steel plate and balsa wood core. *Engineering Structures* 300: 117256, DOI: 10.1016/j.engstruct.2023.117256