buildings

MDPI

Article

# Research on a Data-Driven Modeling Method for Precast Concrete Balcony Components

**Jie Cai [1], Xin Wang [1,*], Junfeng Shi [1], Xingxing Xie [2], Yu Feng [1] and Yingjun Wu [1]**

[1] School of Civil Engineering, Architecture and Environment, Hubei University of Technology, Wuhan 430068, China
[2] Beijing Olive Hill Software Co., Ltd., Beijing 100095, China
* Correspondence: 102100814@hbut.edu.cn

**Abstract:** In this paper, a data-driven modeling method for precast concrete (PC) balcony components was proposed to solve the problems of low informatization and the difficult modeling of components at the design stage. Through the analysis of the characteristics of PC balcony components and the combination of modular design methods, the paper designed a data structure for the components and developed a data-driven modeling tool for PC balcony components that can realize the input of structural design data, automatically generating component models. First, this paper introduced the data-driven modeling concept and the modeling process. Second, the PC balcony components in common prefabricated residential projects were analyzed to identify their characteristics. By using a modular design approach, these components were divided and a module dataset was created based on the split modules. Consequently, a data structure for the prefabricated balcony component model was established, wherein both conventional parameters and adaptive parameters between modules were interrelated. Finally, the function of data-driven modeling was achieved by developing a modular design tool on the Revit platform using the C# programming language. The application conducted on a prefabricated building project demonstrated that the software tool and modeling method in this paper effectively improve the level of informatization and modeling efficiency of PC balcony components. The modular design approach was satisfied with the standardization and diversification requirements of balcony components, thereby offering insights for modeling other complex components.

**Keywords:** PC balcony component; data-driven modeling; modularization; Revit secondary development

check for updates

**Citation:** Cai, J.; Wang, X.; Shi, J.; Xie, X.; Feng, Y.; Wu, Y. Research on a Data-Driven Modeling Method for Precast Concrete Balcony Components. *Buildings* **2024**, *14*, 96. https://doi.org/10.3390/buildings14010096

Academic Editors: Yung Yau, Youjin Jang, Jeehee Lee and Soowon Chang

Received: 8 November 2023
Revised: 25 December 2023
Accepted: 27 December 2023
Published: 29 December 2023

## 1. Introduction

With the integration of the sustainable development concept in China, the construction industry has embraced green, efficient, and sustainable practices as its fundamental objectives. Prefabricated building, being an innovative construction method, has gained considerable attention from the industry due to its advantages in industrial component production, environment-friendly construction practices, and shortened construction cycles [1,2]. The factory production and site assembly characteristics of precast components offer significant advantages over the traditional on-site casting construction method in terms of enhancing construction efficiency, reducing construction costs, mitigating pollution, and improving the on-site environment [3–6]. However, despite the numerous advantages of prefabricated buildings, their limited standardization in design, underdeveloped production processes, and inadequate part compatibility impede the enhancement of construction efficiency and hinder the full use of prefabricated building benefits, thereby obstructing widespread adoption. Enhancing the standardization level and design excellence of prefabricated constructions, increasing component reusability, and optimizing the production process are crucial measures for enhancing the overall advantages of prefabricated buildings [7].

The modular design of prefabricated components is considered a pivotal approach to augmenting the overall advantages of prefabricated buildings. Prefabricated buildings are constructed by assembling premanufactured components produced in an off-site production facility [4]. A commonly used design approach involves initially conceptualizing the building as a monolithic structure and subsequently dividing it into prefabricated components based on specific guidelines [8]. However, this approach often results in a wide array of premanufactured components, thereby augmenting the intricacy and expenditure associated with construction. With the progression of industrialization, the adoption of modular design methodology in the domain of prefabricated constructions is steadily increasing. The implementation of modular design theory facilitates the establishment of a module library, thereby enhancing component standards, while accommodating their diversity and subsequently improving construction efficiency [9–11]. The implementation of modular design can enhance component reusability, minimize redundant investments in templates and equipment, expedite construction timelines, and optimize component accuracy and consistency. In the context of prefabricated component modular design, achieving standardization in module splitting while ensuring compatibility between split modules has emerged as a pivotal concern. This necessitates adopting a scientific and rational approach to both design and management.

The digital tool building information model (BIM) is extensively used in the design and management of prefabricated buildings, leveraging construction project data to create a model that accurately simulates building information through digital means [12,13]. Yuan et al. combined parametric design and BIM technology to propose a method for the modularization of prefabricated components, aiming to enhance the standardization level of such components [7]. Wang integrated lean construction principles with BIM technology, proposing a novel workflow aimed at enhancing the design quality of prefabricated components [14]. By integrating BIM technology with the concept of Design for Manufacturing and Assembly (DFMA) in the manufacturing industry, researchers have successfully applied it to precast buildings for the division and design of precast foundation components. This integration has significantly enhanced the modeling efficiency of precast concrete elements [15,16], enabling better management of their production process, ensuring superior quality and accuracy, and improving overall production efficiency. Furthermore, this integration aims to facilitate seamless coordination between design models of prefabricated components and production processes [17], thereby promoting industrialization, digitization, and intelligent development within the construction industry.

The use of computer programming languages and the secondary development technology of Revit facilitates the development of specialized software tailored to diverse projects [18,19]. BIM technology integrates procedures and algorithms, thereby enhancing support for data and models in addressing practical engineering challenges [20–22]. The data and information, serving as the fundamental components of BIM models, encompass all the attributes and characteristics of buildings [23]. They provide enhanced precision and efficacy in design and construction through data-driven support. Meng et al. developed a tool for reasoning parameter modeling by using the Revit modeling platform and the computer programming language C# to improve modeling efficiency [24]. Liu et al. combined BIM technology with a data-driven framework to identify BIM model errors [25]. T. Kirchdoerfer and M. Ortiz developed the data-driven computing paradigm with the help of computer languages to illustrate the robustness of data-driven solvers [26]. Data-driven thinking plays a crucial role in the process of BIM secondary development. By integrating algorithms into programs, the focus is on automating and intelligently managing data for design and construction processes [27–29]. Despite the application of data-driven concepts and methods in BIM models, there is limited research on their implementation in the modeling approach for PC balcony components.

As a type of precast element, PC balcony components offer significant advantages in terms of enhancing construction efficiency and reducing costs. However, existing design methods fail to meet the diverse and individual requirements of real-life projects' components, while also lacking effective tools for data-driven modeling. Therefore, this study aims to explore integrating data-driven modeling concepts with BIM technology to address information design needs, enhance design and production efficiency, and facilitate widespread adoption of PC balcony components.

Building on previous research, in this paper, a data-driven component modeling method was proposed, which was used to analyze the characteristics of building components and develop a corresponding software tool (based on a BIM software platform (Revit 2016 version)) to create the component model of a prefabricated balcony. The method was used to solve the problems of low informatization, difficult modeling, and poor practical application of model data. After analyzing the architectural characteristics of the balcony's components, the components were divided into modules according to their architectural characteristics, the modules' characteristic parameters were established, the modules' data structure was designed, and a module dataset was also established. On this basis, using the C# programming language and Visual Studio programming tools, the WinForms application window was developed, and according to the spatial relationship of the modules, the automatic assembly of the program was used to accomplish the establishment of the model. This paper describes, in detail, the optimization of a modeling method for prefabricated balcony components using data-driven techniques. The research presents a novel idea and methodology for addressing information design challenges in prefabricated component systems, offering valuable insights and assistance in tackling practical issues encountered by prefabricated buildings. Furthermore, the findings of this study will serve as a reference for related research endeavors.

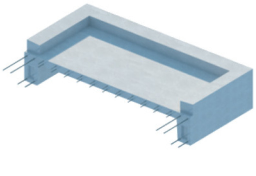
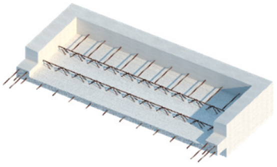
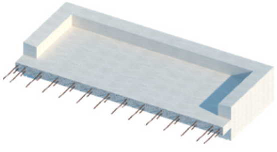
## 2. The Challenges in the Modeling of PC Balcony Components

### *2.1. Classification of PC Balcony Elements*

The PC balcony is an essential component among the prefabricated components of an assembly building. The PC balcony panel is categorized as a beam balcony or a slab balcony based on the type of structural pressure. The slab balcony, in turn, is subcategorized as a complete prefab slab balcony or a stacked prefab slab balcony, depending on the panel's prefabrication method [30].

Prefabricated beam balconies are a type of balcony that transfers the balcony slab and its loads to the beams, walls, columns, and other structures of the main building through the use of a pick-and-suspend beam. At the end of the overhanging beam, an edge beam is installed to support the load of the balcony railing and any other additional loads. The keyway is situated at the termination of the overhanging beam connected to the cast-in-place section. The balcony slab of the prefabricated slab type uses a cantilevered slab design, where the edge is combined with the entire beam slab of the primary structure. The load placed on the slab is then transferred to the main structure's beam slab through the cantilevered slab. Slab balconies tend to be used in small-span balconies, due to limitations stemming from the structural force's form. The components that form a PC balcony are presented in Table 1.

**Table 1.** Classification of PC balcony elements.

| **Beam Balcony** | **Slab Balcony** | |
|---|---|---|
| | Stacked prefab slab balcony | Complete prefab slab balcony |
|  |  |  |

### *2.2. The Challenges in Modeling PC Balcony Elements*

There are two modeling approaches for traditional prefabricated balcony components. In one approach, the floor command is used to draw the balcony plate, while the beam command is used to illustrate both the cantilever beam structure and the edge structure. The other approach involves using lofting, Boolean operations, and other functions to generate a geometric contour drawing of the component based on family creation. Subsequently, a reinforcement model is established in accordance with the reinforcement command. In following this approach, there are the following problems:

1. The information regarding the components is widely scattered, indicating a relatively low level of informatization.

Traditionally, the plate structure model and the beam structure model are established as separate components, along with the geometric contour model and the reinforcement model. This results in dispersed data information within the component models and low accuracy of data. However, for precast balcony component models that require alignment with factory production, scattered data sources can easily lead to deviations during the transmission of model information, resulting in mismatches between data and models that ultimately affect component production efficiency.

2. The software operation is intricate, and the modeling efficiency is suboptimal.

Although the commands for reinforcing bar modeling are included in 3D modeling software, practical application can be challenging due to the diverse and intricate nature of reinforcing bars in precast balcony members. Taking Revit as an example, rebar modeling is required to be based on a specific plane. The built-in plane options include the "current working plane", "near protective layer reference", and "far protective layer reference". The placement direction can be categorized into three forms: "parallel to the working plane", "parallel to the protective layer", and "perpendicular to the protective layer". Prior to drawing the rebar, different protective layers need to be established, requiring frequent switching between planes and sections during rendering. This complexity makes it difficult for users to master the software and results in relatively low modeling efficiency.

To summarize, the current 3D modeling software commonly used in the industry lacks the capability to independently model prefabricated balcony components. In order to address this issue, we proposed the following optimizations: (1) Enhance the modeling approach for prefabricated balcony components by adopting a modular component concept and establishing a module dataset. This will enable us to design a structured data model for balcony components, reducing issues arising from scattered data sources and standardizing data management. (2) Develop a modeling software tool that simplifies the logic used by modelers and integrates algorithms with programmatic capabilities. By leveraging data-driven approaches, we can significantly enhance the efficiency in creating models.

## 3. Methods

In this paper, we optimized the modeling method for PC balcony components by incorporating data-driven principles. First, we conducted an analysis of the characteristics of balcony components and designed a module dataset and data structure using a modular approach. Subsequently, leveraging Revit secondary development technology, we developed a software tool to enable rapid establishment of prefabricated balcony component models. Our aim was to address challenges, such as the low level of informatization in these components, difficulties in modeling, and limited practical application of model data. Ultimately, this will enhance the efficiency of modelers in addressing modeling issues related to prefabricated balcony components. The specific methodology process is illustrated in Figure 1.

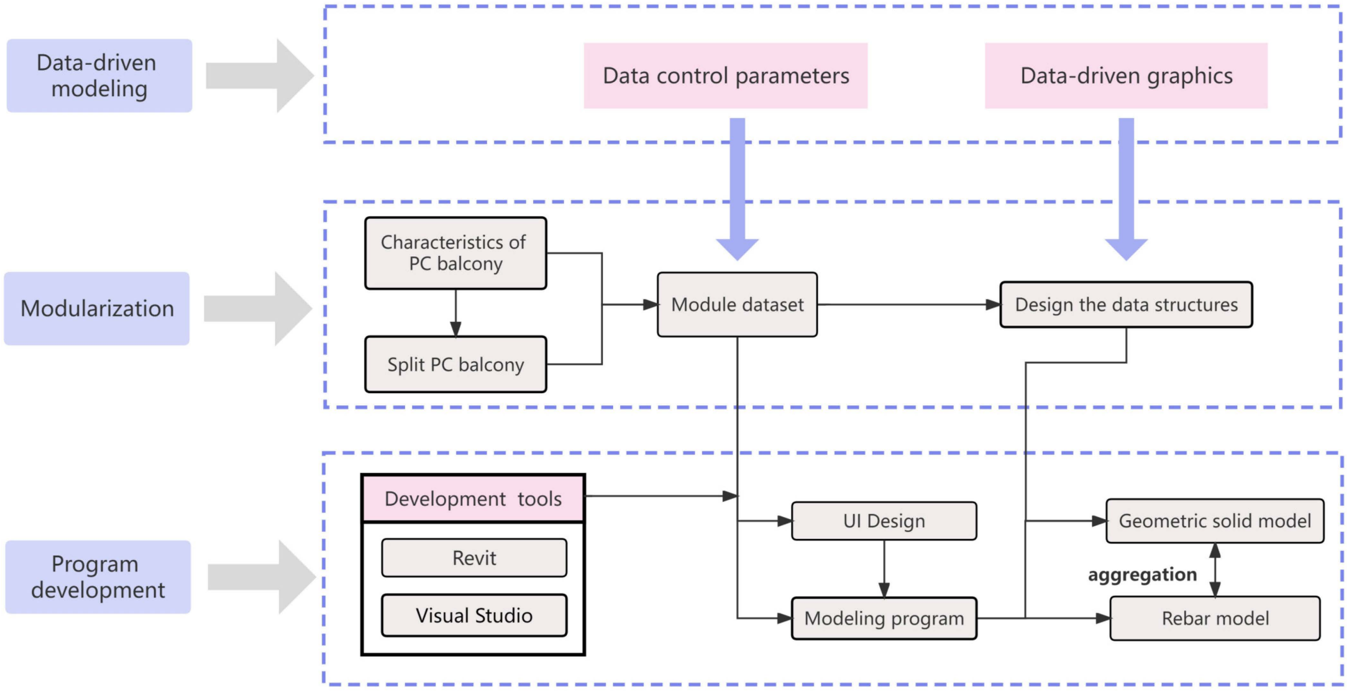


**Figure 1.** Research on the data-driven parametric modeling process.

### 3.1. Concepts and Techniques of Data-Driven Modeling

Data-driven modeling is accomplished using the online and offline data of a controlled system to establish a database, adjust it, judge it, and optimize it, in addition to other functions [31]. The data-driven modeling approach uses data as characteristic parameters for objects, facilitating the automatic control and generation of objects within the model, and modifications made to the feature parameters are accurately reflected in the 3D model.

The principle of traditional parametric modeling is to separate data and graphics, forming a comprehensive model by matching the ID of graphic elements (e.g., the Guid value of graphic elements in Revit). These models acquire information from databases and graphics libraries separately. In contrast to conventional parametric modeling methods, data-driven modeling places data at a more central position. It relies on controlling and constraining actual data using tools, such as parameters, formulas, and tables, to generate and modify building models. Throughout this process, data drive the set of graphs, ensuring that the parameters of the graph layer are entirely dependent on the data source for perfect alignment between the model and its underlying information. The process of comparing two modeling methods is shown in Figure 2.

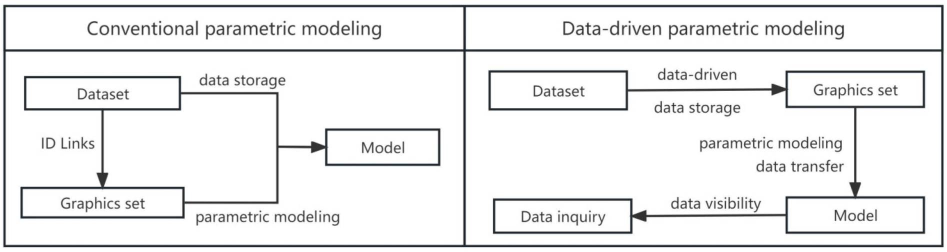


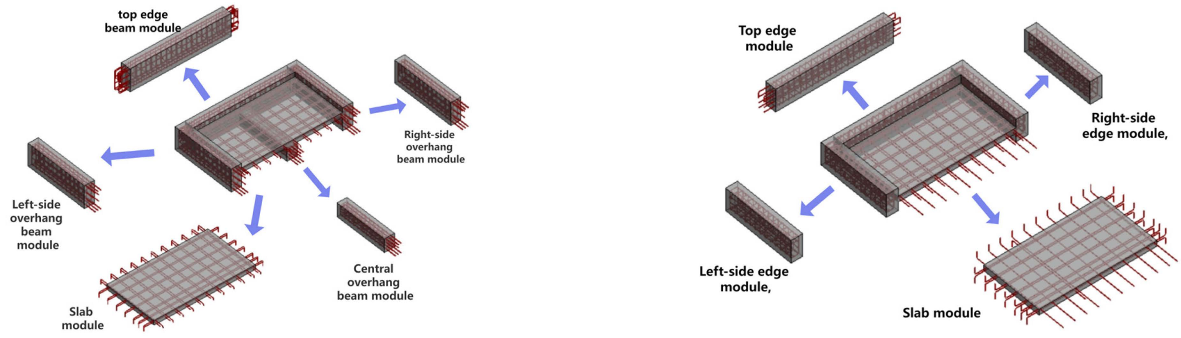
**Figure 2.** Comparison of the processes of parametric modeling vs. data-driven modeling.

### *3.2. Modularization of Prefabricated Balcony Components*

#### 3.2.1. Modular Splitting

The components of an assembly building can be categorized into horizontal and vertical elements. When designing a modular split, two perspectives need to be considered: a top-down partitioning model and a bottom-up composition model.

According to the classification of PC balcony components, modules are categorized based on their characteristics. Specifically, the beam-type balcony can be divided into five modules: slab module, left-side overhang beam module, right-side overhang beam module, central overhang beam module, and top edge beam module. Similarly, the slab-type balcony can be split into four modules: slab module, left-side edge structure module, right-side edge structure module, and top edge structure module. Each of these modules is designed with specific datasets, which are then integrated into a comprehensive whole through parameter association between them. The breakdown of these modules is illustrated in Figure 3.

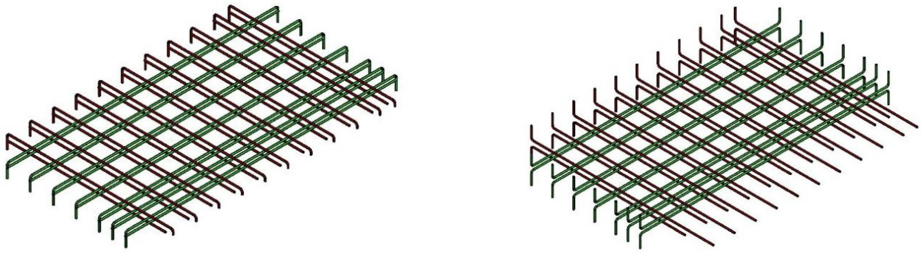


(**a**) (**b**)

**Figure 3.** (**a**) Beam-type balcony module and (**b**) slab-type balcony module.

#### 3.2.2. The Establishment of a Modular Dataset

The slab module dataset was taken as an example in this study. To enhance the representation of the steel reinforcement dataset for the slab module, a three-dimensional form was used to present the steel reinforcement. In Figure 4, green color indicates the steel reinforcement in the overhang direction, while red color represents the steel reinforcement in the vertical direction.



(**a**) (**b**)

**Figure 4.** Reinforcement schematic of the slab module: (**a**) beam balcony and (**b**) slab balcony.

We divided the slab module dataset into three parts: the concrete geometry profile dataset, the rebar overhang direction dataset, and the vertical direction dataset. The parameter names and values defined in the current module were stored in the dataset. These parameters included but were not limited to the thickness, length, and width of the slab, the diameter and spacing of the rebar, etc. The data information of these parameters can provide full flexibility for modeling and adjust and optimize the design and modeling of the balcony plate according to the actual needs. The parameter dataset of the slab module is structured as shown in Table 2.

**Table 2.** The parameter dataset of the slab module.

| Slab Module Dataset | Dataset Name | Parameter Name | Value Type | Meaning of Parameters |
|---|---|---|---|---|
| Concrete Geometry Profile Dataset | BeamFloorBaseInfo | length | Double | Length of slab |
| | | width | Double | Width of slab |
| | | height | Double | Thickness of slab |
| | | depth | Double | Depth of slab end embedding |
| Rebar Overhang Direction Dataset | XRebarInfo | upperDiameter | Double | Diameter of upper rebar |
| | | upperDistance | Double | Spacing of upper rebar |
| | | downDiameter | Double | Diameter of lower rebar |
| | | downDistance | Double | Spacing of lower rebar |
| Rebar Vertical Direction Dataset | YRebarInfo | upperDiameter | Double | Diameter of upper rebar |
| | | upperDistance | Double | Spacing of upper rebar |
| | | downDiameter | Double | Diameter of lower rebar |
| | | downDistance | Double | Spacing of lower rebar |

Note: This paper solely used the slab module as an illustrative example, while the application methodology for other modules remained consistent.

#### 3.2.3. Component Data Structure Design

The definition of data structures is one of the premises that enable data-driven modeling, and its essence lies in the summarizing and sorting of modular datasets for precast balcony components. According to the methodology of data-driven modeling, this study used UML class diagrams for visualizing classes and their relationships in a more professional manner. The class diagram plays a pivotal role in the development process of modeling tools as it visually represents the data structure of the model, serving as an effective means of communication and comprehension for developers. It facilitates a clearer understanding of both the structure and the behavior of the model. The development of modeling tools relies on class diagrams, which are primarily used by developers to write code, complete the modeling process, establish relationships between classes using various programming languages, and ultimately implement the behavioral functionality encapsulated within each class through programming. Therefore, class diagrams play a crucial role in achieving functional realization of modeling tools.

The program design of this study was divided into five components: slab module class, edge member geometry class, beam module geometry class, edge member rebar class, and beam rebar class. Among them, various forms of reinforcement classes were inherited from the reinforcement base class, including common parameter data defined in the reinforcement base class, such as the reinforcement diameter and number, and specific data, such as the anchor length. These classes were composed of the beam module reinforcement class and the edge module reinforcement class. The corresponding geometric module classes were aggregated with the reinforcement data class to form a complete prefabricated balcony module dataset. The data structure of the model was designed. The integration of module datasets from diverse sources enhances the management and application of data, while addressing the problems arising from scattered data sources. Data structure UML class diagrams are shown in Figure 5.

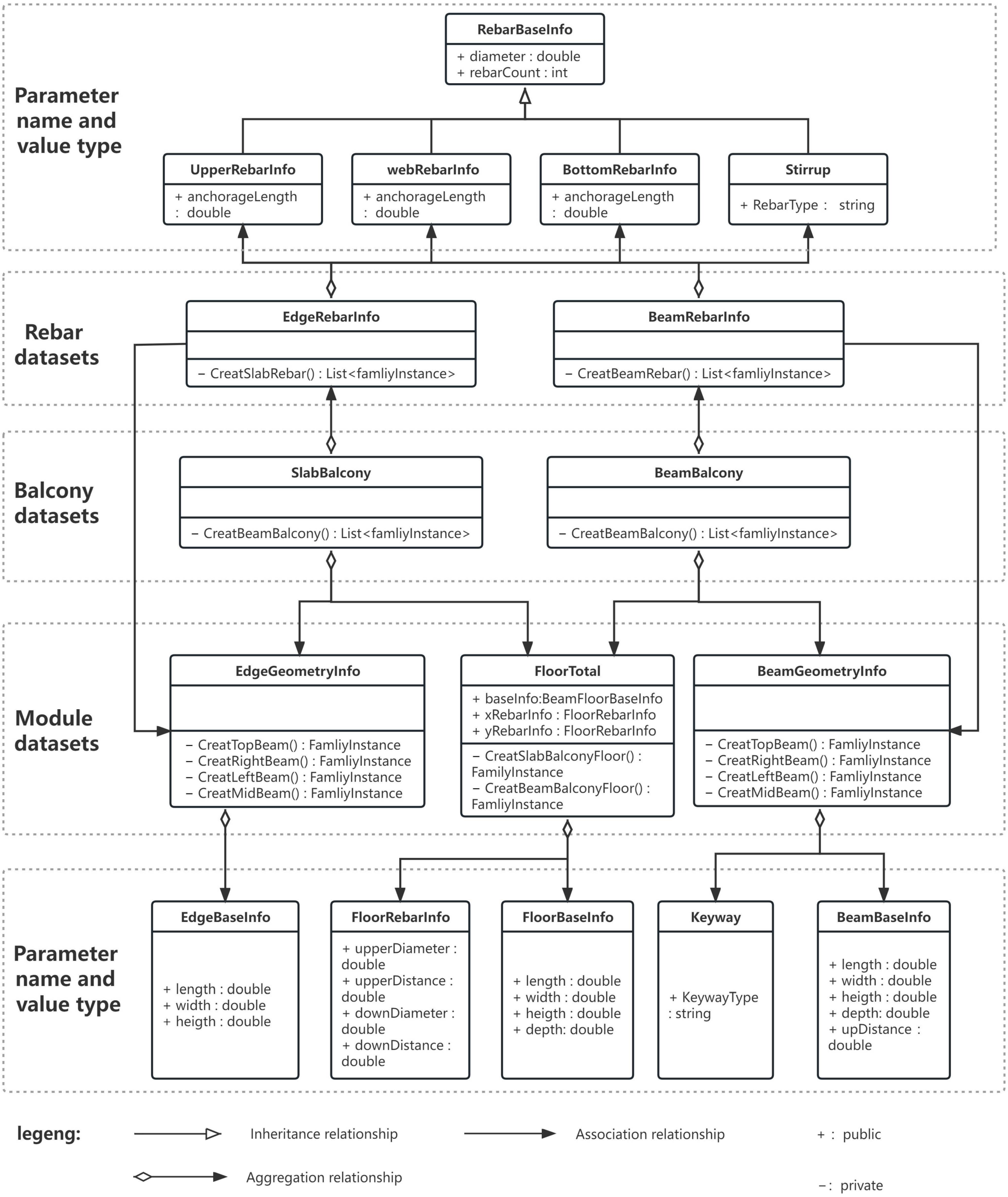

Parameter name and value type
RebarBaseInfo
+ diameter : double
+ rebarCount : int
UpperRebarInfo
+ anchorageLength : double
webRebarInfo
+ anchorageLength : double
BottomRebarInfo
+ anchorageLength : double
Stirrup
+ RebarType : string
Rebar datasets
EdgeRebarInfo
− CreatSlabRebar() : List<famliyInstance>
BeamRebarInfo
− CreatBeamRebar() : List<famliyInstance>
Balcony datasets
SlabBalcony
− CreatBeamBalcony() : List<famliyInstance>
BeamBalcony
− CreatBeamBalcony() : List<famliyInstance>
Module datasets
EdgeGeometryInfo
− CreatTopBeam() : FamliyInstance
− CreatRightBeam() : FamliyInstance
− CreatLeftBeam() : FamliyInstance
− CreatMidBeam() : FamliyInstance
FloorTotal
+ baseInfo:BeamFloorBaseInfo
+ xRebarInfo : FloorRebarInfo
+ yRebarInfo : FloorRebarInfo
− CreatSlabBalconyFloor() : FamilyInstance
− CreatBeamBalconyFloor() : FamliyInstance
BeamGeometryInfo
− CreatTopBeam() : FamliyInstance
− CreatRightBeam() : FamliyInstance
− CreatLeftBeam() : FamliyInstance
− CreatMidBeam() : FamliyInstance
Parameter name and value type
EdgeBaseInfo
+ length : double
+ width : double
+ heigth : double
FloorRebarInfo
+ upperDiameter : double
+ upperDistance : double
+ downDiameter : double
+ downDistance : double
FloorBaseInfo
+ length : double
+ width : double
+ heigth : double
+ depth: double
Keyway
+ KeywayType : string
BeamBaseInfo
+ length : double
+ width : double
+ heigth : double
+ depth: double
+ upDistance : double
legeng:
Inheritance relationship
Association relationship
+ : public
Aggregation relationship
− : private

**Figure 5.** Data structure hierarchy UML class diagrams.

## 3.3. *Revit Secondary Development*

### 3.3.1. Development Platforms and Processes

In this paper, Revit 2016 software was selected for secondary development and Visual Studio 2017 was selected as the development platform; under.NET Framework 4.7, the program was written in the computer language C#, and the functions and interfaces within the Revit API were used. A dynamic link library (.dll) file was compiled and generated to accomplish the extension of the software functions. The development process is shown in Figure 6.

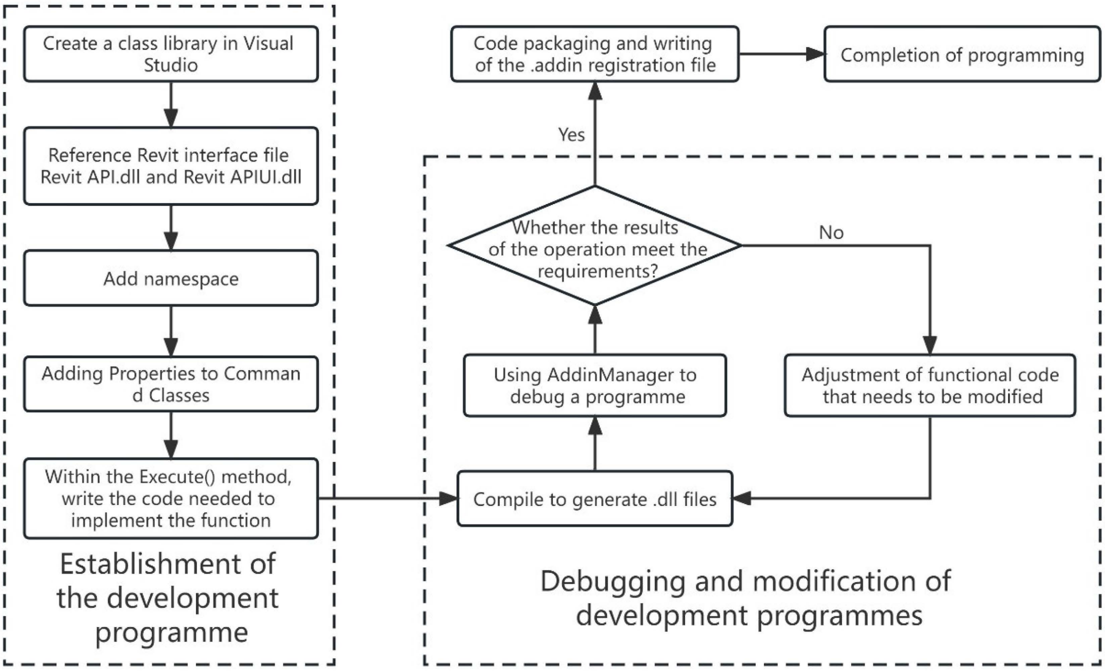

Create a class library in Visual Studio
Reference Revit interface file Revit API.dll and Revit APIUI.dll
Add namespace
Adding Properties to Comman d Classes
Within the Execute() method, write the code needed to implement the function
Establishment of the development programme
Code packaging and writing of the .addin registration file
Completion of programming
Yes
Whether the results of the operation meet the requirements?
No
Using AddinManager to debug a programme
Adjustment of functional code that needs to be modified
Compile to generate .dll files
Debugging and modification of development programmes

**Figure 6.** The development process.

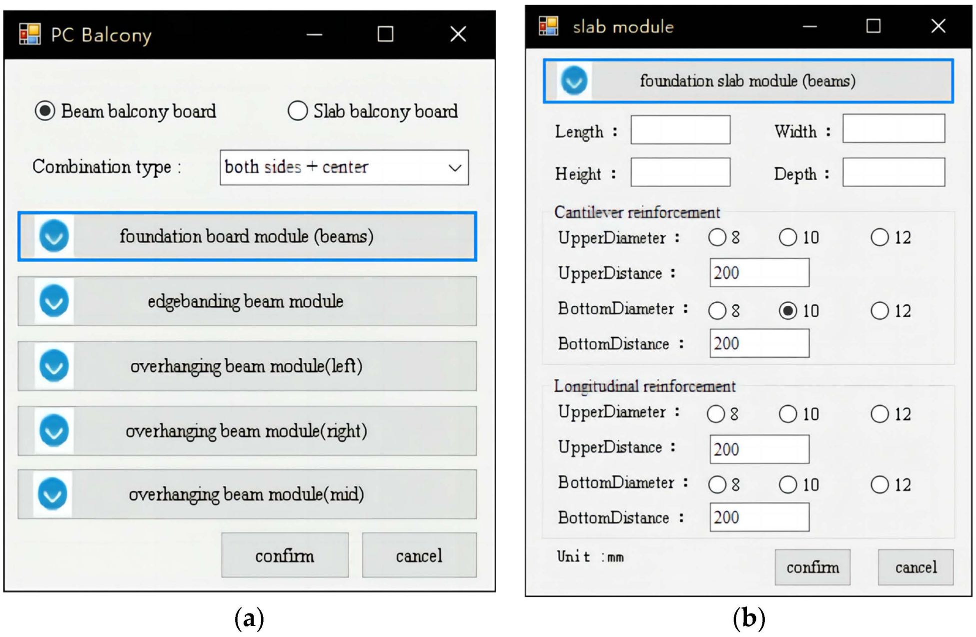
### 3.3.2. The Process of Tool Development

1. User interface (UI) design

The basic logic of interface design must place the interface under the control of the user and reduce the user's memory load. Since PC balcony components are composed of modules, each module was designed as an embedded sub-interface to maintain the overall consistency of the interface when designing the interface. WinForms (Windows Form) is a widely used development tool that is based on the .Net form application. It consists of two parts: front interface design and background code. The user input data can be obtained through the background code. The design of the main interface of the modeling tool and the sub-interface of the corresponding module are shown in Figure 7.

2. Establishment of a solid model of the PC balcony

It is necessary to determine the elevation at which the balcony will be located before creating the balcony elements. The reference coordinate point for generating the balcony component should be selected in the corresponding plane view interface, and the coordinates of this point should be obtained. The model's origin was defined as the location point, which was transformed into origin coordinates using the Element.Location() function in the API. This created the origin Location (0, 0, 0), allowing the re-establishment of the Cartesian coordinate system and relocation calculations for the balcony component's modules. Taking the beam balcony as an example, the plan view of its geometric model is shown in Figure 8. The slab dimensions are denoted as $L_1$ for length and $W_1$ for width, while the widths of the top edge beam and the left- and right-side overhanging beams are, respectively, marked *TW*, *LW*, and *RW*. Additionally, the balcony's Z-direction offset height is defined as $\mathrm{d}_1$ and the depth of its embedment is $T_1$. In Table 3, the computed positioning points and lengths of the remaining modules are presented.



PC Balcony

● Beam balcony board ○ Slab balcony board

Combination type : both sides + center

foundation board module (beams)
edgebanding beam module
overhanging beam module(left)
overhanging beam module(right)
overhanging beam module(mid)
confirm
cancel

slab module

foundation slab module (beams)

Length : Width :

Height : Depth :

Cantilever reinforcement

UpperDiameter : ○ 8 ○ 10 ○ 12

UpperDistance : 200

BottomDiameter : ○ 8 ● 10 ○ 12

BottomDistance : 200

Longitudinal reinforcement

UpperDiameter : ○ 8 ○ 10 ○ 12

UpperDistance : 200

BottomDiameter : ○ 8 ○ 10 ○ 12

BottomDistance : 200

Unit :mm

(**a**) (**b**)

**Figure 7.** The generated window: (**a**) the main interface and (**b**) slab module sub-interface.

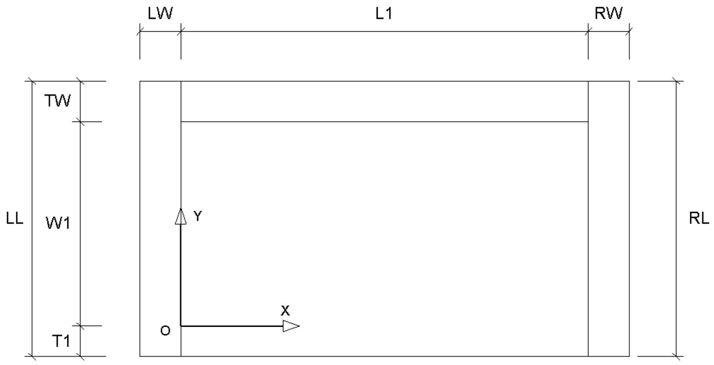


**Figure 8.** Balcony component plan.

**Table 3.** Module positioning points and lengths.

| | Location (X, Y, Z) | Length |
|---|---|---|
| Top edge beam | (0, $W_1$, $d_1$) | $TL = L_1$ |
| Left overhanging beam | (0, $W_1$, $d_1$) | $LL = TW + W_1 + T_1$ |
| Right overhanging beam | (0, $W_1$, $d_1$) | $RL = TW + W_1 + T_1$ |

The function method used to create the geometric solid model of the modules in the balcony component was Creat.NewFamilyInstance (XYZ, FamilySymbol, StructuralType), and this method corresponded to three parameters, which were the positioning point of the instance model, the type of the model, and the structural type. The anchor point of each module could be obtained through the aforementioned calculation, enabling the generation of a single-module model using code-driven parameter data. Using the function List<ElementId>, all module models were aggregated to form a complete geometry model of the balcony.

3. Establishment of rebars model of the PC balcony

As the rebar model differs from the concrete model, the rebar is classified as a component attached to the concrete interior. The prefabricated components of the reinforcement layout have the characteristics of uniformity, standardization, and repeatability. The establishment of rebar parameter logic and the setting of various parameters are essential in

Revit to accommodate different anchorage forms. Reinforcement modeling in data-driven methods relies on coordinate calculation to generate positioning curves and form reinforcement models. Taking the edge girder as an example, this section introduces the program algorithm for generating the reinforcement model.

Program algorithm calculations involved using the aforementioned known geometric point location coordinates of the capping beam. The geometric point location coordinates TopBeamLocation (0, $W_1$, $d_1$) calculated before were transformed into the reference coordinate point of the module steel bar: Location O (Xtop, Ytop, Ztop) established the coordinate system, as shown in the figure. The CreateBound () function was used to create the positioning curve of the rebar. In accordance with the creation of the positioning line, it was then used to generate the longitudinal rebar models.

The thickness of the protective layer of the reinforcement was denoted as *Protect layer thickness* (abbreviated to *Prot* in the formula), the diameter of the stirrup was *Stir dia*, the diameter of the longitudinal reinforcement was *Longi dia*, the anchorage length of the reinforcement was 15 times the diameter of the reinforcement, and the local arrangement and composition of the reinforcement are shown in Figure 9.

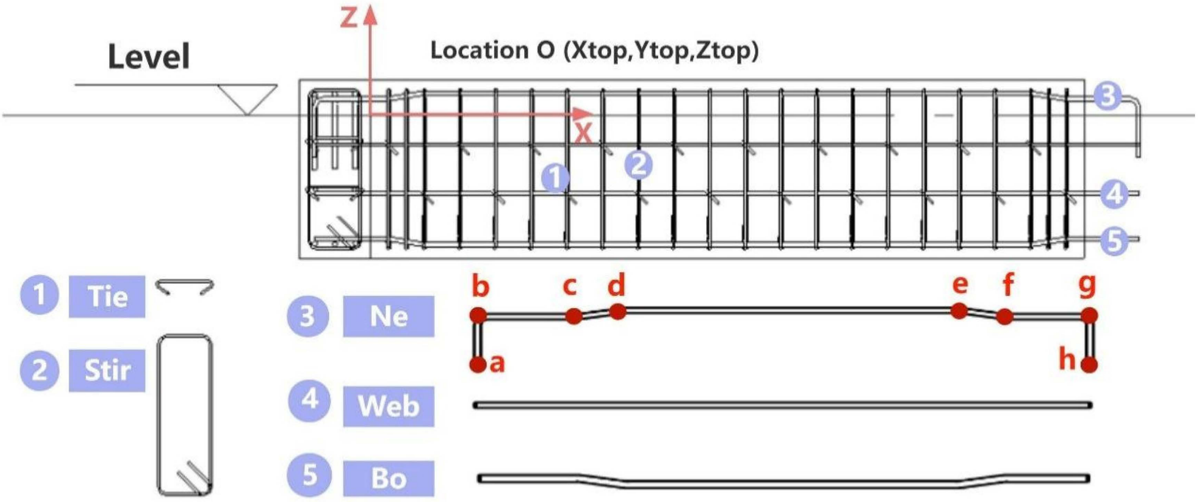


**Figure 9.** Local layout and composition of rebar of the top edge beam.

From Figure 9, it can be seen that the $Y$-axis value of the rebar is fixed, so the following calculation mainly calculates the $X$-axis value and the $Z$*-axis* value of the rebar positioning point. The curves of the rebar *Ne* are represented in Figure 9 by the position of the coordinates of the locating points a~h. The value in the Y-axis direction of the rebar can therefore be calculated as follows:

$$Y_{Ne} = Y_{top} - Prot - Stir\ dia - \frac{Longi\ dia}{2} \tag{1}$$

The coordinates of the locating points of the reinforcement a~h are then calculated as follows:

Location a ($X_a$, $Y_{Ne}$, $Z_a$):

$$\begin{cases} X_a = X_{top} - LW + Prot + \frac{Longi\ dia}{2} \\ Z_a = Z_{top} + d_1 - Prot - (16 + \frac{1}{2})Longi\ dia \end{cases} \tag{2}$$

Location b ($X_b$, $Y_{Ne}$, $Z_b$):

$$\begin{cases} X_b = X_{top} - LW + Prot + \frac{Longi\ dia}{2} \\ Z_b = Z_{top} + d_1 - Prot - \frac{3}{2}Longi\ dia \end{cases} \tag{3}$$

Location c ($X_c$, $Y_{Ne}$, $Z_c$):

$$\begin{cases} X_c = X_{top} + Prot + \frac{Stir\ dia}{2} \\ Z_c = Z_{top} + d_1 - Prot - \frac{3}{2} Longi\ dia \end{cases} \tag{4}$$

Location d ($X_d$, $Y_{Ne}$, $Z_d$):

$$\begin{cases} X_d = X_{top} + Prot + \frac{Stir\ dia}{2} \\ Z_d = Z_{top} + d_1 - Prot - \frac{Longi\ dia}{2} \end{cases} \tag{5}$$

Location e ($X_e$, $Y_{Ne}$, $Z_e$):

$$\begin{cases} X_e = X_{top} + TL - Prot - \frac{Stir\ dia}{2} - 6 \times Longi\ dia \\ Z_e = Z_{top} + d_1 - Prot - \frac{Longi\ dia}{2} \end{cases} \tag{6}$$

As the data for the avoidance curves at both ends of the beam are the same, the coordinates of points f, g, and h can be derived by referring to the calculation of points a, b, and c. Similarly, the positioning coordinates of the rebar Bo can be deduced, and the detailed formulae are not listed here. After generating the geometric shape of the rebar, it is necessary to calculate the arrangement of the rebar within the concrete and several common cross-sectional arrangements of rebar are shown in Figure 10. In the figure, *a* is the value of the center of the reinforcement from the concrete boundary and *b* is the spacing of the reinforcement rows. The double-layer reinforcement form defines $c_1$ as the vertical distance between the upper reinforcement and the top surface, while $c_2$ represents the vertical distance between the lower reinforcement and the top surface.

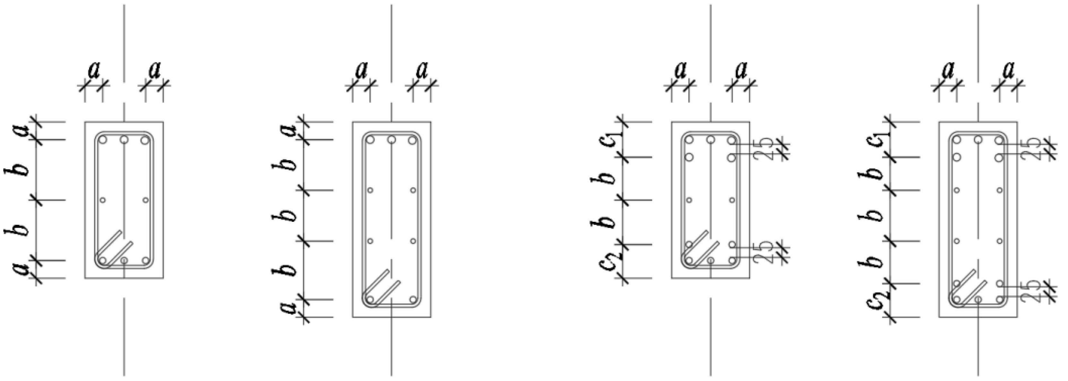


**Figure 10.** Schematic diagram of several common types of reinforcement cross-section arrangement.

A cross section of the center of the closing beam is shown in Figure 11. Since the specific coordinates of *rebar 3* in the edge beam have already been calculated before (and the value in the *X*-axis direction is the length of the reinforcement, which has no relation to the spacing arrangement of the longitudinal reinforcement), only the positioning value in the *Y*-axis direction and the fixed position in the *Z*-axis direction are calculated in the alignment positioning calculation process next.

The length and width of the beam section are *TH* and *TW*, respectively, and the coordinates of the origin are $X_{top}$, $Y_{top}$, and $Z_{top}$. The position coordinates of rebars 1–8 were then identified. First, the coordinates of the rows of rebar 3 in Figure 11 can be calculated as follows:

Rebar 3 Location ($X_{\mathrm{d}}$, $Y_3$, $Z_3$):

$$\begin{cases} Y_3 = Y_{Ne} \\ Z_3 = Z_{top} + d_1 - Prot - \frac{Longi\ dia}{2} - Stir\ dia \end{cases} \tag{7}$$

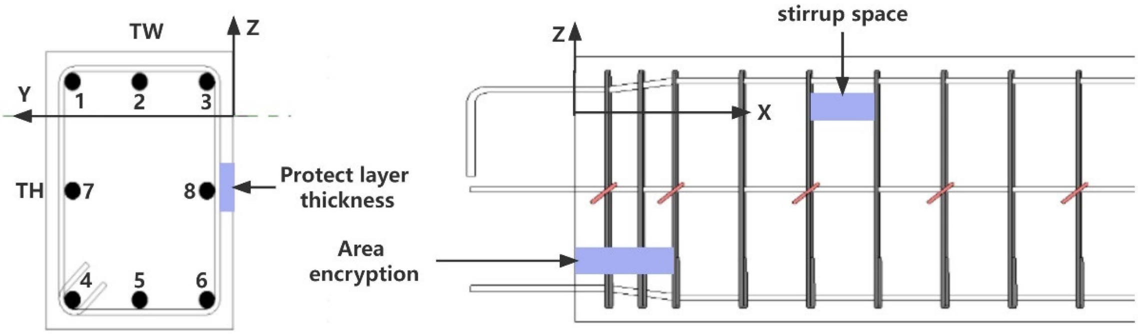


**Figure 11.** Schematic of the beam rebar arrangement.

If the number of bars in the same row is greater than or equal to 3, the ElemengtTransformUtils.CopyElement() method in the Revit API may be used to copy the corresponding bars. By reading the number of top bars in the rebar dataset (*top count*) and the copy spacing of the bars (*Copy space*), the following calculation can be performed:

$$Copy\ space = \frac{TW - 2 \times Prot - 2 \times Stir\ dia - Longi\ dia}{top\ count - 1} \tag{8}$$

Using rebar 3 as a reference, the copy distance and positioning points of rebars 1 and 2 on the diagram may be found. Similarly, the end-point coordinates and copy distances of rebars 4, 5, and 6 can be calculated as described before, and the rebars can be modeled by copying. The number of longitudinal bars at the waist from the rebar dataset (*web count*) and the form of rebar rows for the web longitudinal bars can be determined based on the number of bars. The number of web bar rows is calculated as follows:

$$N_i = \frac{webcount}{2} \tag{9}$$

The spacing of the rebar (*Rebar space*) can then be calculated:

$$Rebar\ space = \frac{TH - 2 \times Prot - 2 \times Stir\ dia - Longi\ dia}{N_i} \tag{10}$$

Using Figure 11 as an example, at this point, the number of web bars is 2; this means that bars 7–8 are in the same row, from which the positioning point of bar 8 can be calculated:

Rebar8 *Ylocation:*

$$Ylocation = Y_{Ne} \tag{11}$$

Rebar8 *Zlocation:*

$$Zlocation = Z_3 - Rebar\ space \tag{12}$$

The method of calculating the position of the other web rebar was essentially the same as that described before. After generating the stirrup and tiepiece, the balcony module may have a rotation angle, so it was necessary to use the List<ElementId> function to collect the IDs of the generated rebar and add the model of the rebar with the corresponding IDs to the balcony module generated in the previous section. Finally, the rotation was performed using the ElementTransformUtils.RotateElements() method. The generation path of the stirrup was as follows. First, the coordinates of the four corners of the stirrup were obtained and the curve shape was drawn through the four corners. The coordinates of these four corners were calculated using the cross-section size of the beam and the thickness of the protective layer. The protective layer of the reinforcement (*Rebar cover*), the diameter of the reinforcing bar (*Stir dia*), and the coordinates of the upper-right corner of the rebar in the Y-direction and in the Z-direction can be calculated based on the aforementioned rebar coordinate system, as follows:

$$\begin{cases} Y_{stir} = Y_{top} + (Rebar\ dia + \frac{Stir\ dia}{2}) \\ Z_{stir} = Z_{top} + d_1 - (Rebar\ cover + \frac{Stir\ dia}{2}) \end{cases} \tag{13}$$

Because the currently acquired coordinates of the origin are the coordinates of the left side of the beam, the coordinates in the x-direction are the coordinates in the longitudinal direction of the beam, and the coordinate position $X_{stir}$ can be calculated as follows:

$$X_{stir} = X_{top} + (Rebar\ dia + \frac{Stir\ dia}{2}) \tag{14}$$

The stirrup layout in the X-direction, the center row spacing (*Stirrup space*), the number of stirrups ($N_1$), and the beams at both ends of the existence of stirrup encryption area (*Encr area*) were calculated; the encryption area's stirrup spacing is generally half the spacing of the intermediate section, and the number of stirrups in this area ($N_2$) were then found. These two values were then added together to obtain the total number of stirrups, $N_{stir}$. Using the For Loop function, the stirrup model was copied to the appropriate position according to the spacing and number of rebar, and the model was added to the set of components as follows:

$$\begin{cases} N_{stri} = N_1 + N_2 \\ N_1 = \frac{TL - 2 \times Encr\ area}{stirrup\ space} - 1 \\ N_2 = 2 \times (\frac{Encr\ area - Rebar\ cover - \frac{Stir\ dia}{2}}{\frac{Stirrup\ space}{2}} + 1) \end{cases} \tag{15}$$

The main role of the tiepiece is to fix the web longitudinal rebar and the longitudinal rebar, the number of tiepieces, and the positioning of the whole based on the number of longitudinal rebars and the web longitudinal rebar. The number of rows of beams ($N_i$) and the number of stirrups ($N_{stir}$) were known for beam reinforcement. The parity of $N_i$ values was first calculated. When $N_i$ is an odd number, the tiepiece is a single row and the spacing is twice the spacing of the stirrup. When $N_i$ is an even number, the tiepiece for the upper and lower rows of staggered rows is a single row with twice the spacing of the stirrups, staggered adjacent to the spacing of the tie piece, and the spacing of the stirrups is the same.

The balcony modules may have rotation angles, so it is necessary to use the List<ElementId> function to collect the IDs of the generated rebars. Next, the rebar models corresponding to these IDs were added to the previously generated balcony module. Finally, the ElementTransformUtils.RotateElements() method was used for rotation.

Using the modeling approach outlined before, algorithms and codes were integrated and encapsulated, creating an .addin file on the Revit software platform. This realized the development of a specialized software tool for modeling PC balcony components. Through user interface input, this tool enabled the automatic generation of steel-bar-reinforced PC balcony components at specified positions based on provided data.

### 3.3.3. Dataset Storage and Inquiry for Models

In engineering design, modifications to the parameters and types of reinforcing bars are often required. During program design, the current model and data structure of the components must be read, and based on the modified WinForms information, the balcony component data must be regenerated. Adhering to the existing data structure when designing the program can help to establish the principles of information storage.

1. The function for the dictionary dataset must be designed as follows: Dictionary<string, string> to create the relevant dataset based on the type of balcony.
2. In the corresponding dictionary dataset function, the module dataset name is set to Field, while the parameter name and its respective attribute values are allocated to the following value field: Serializes Object.
3. Unique data tag GUID values are stored in the generated instance model data.
4. Finally, all the data are stored on the model and supplied to the user for querying.

As Revit Lookup can access the stored dataset and aid subsequent development efforts for querying information data, the process of querying data is as follows.

1. Revit Lookup may be used to read the unique data tag GUID value stored in the model data in this instance.
2. Since each GUID value is unique, it can be used to look up the dictionary dataset function Dictionary<string, string> for this model.
3. The module dataset names and their corresponding attribute values in the dictionary dataset function can be inquired about, and their specific correspondence is shown in Table 4.

**Table 4.** Balcony dictionary dataset structure.

| Field | Serializes Object | Can It Be Null? |
|---|---|---|
| Floor ID | Number | No |
| Style | Beam/Slab | No |
| Slab | FloorBaseInfo<br>yRebarInfo<br>XRebarInfo | No<br>No<br>No |
| Top | TopBeamInfo<br>TopInfo | Yes<br>Yes |
| Left | LeftBeamInfo<br>FbLeftInfo | Yes<br>Yes |
| Right | RightBeamInfo<br>FbRightInfo | Yes<br>Yes |
| Mid | MidBeamInfo | Yes |

## 4. Case Study

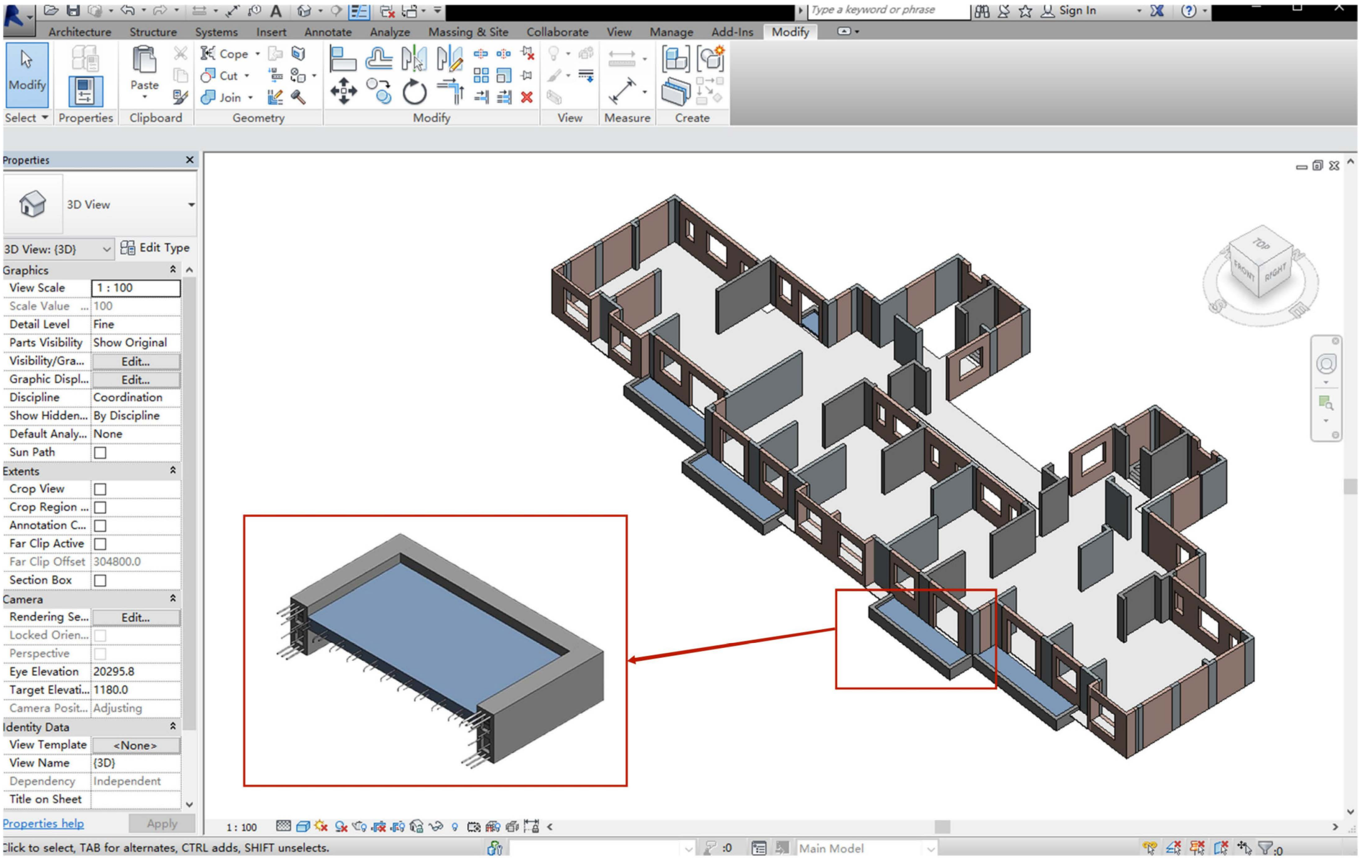
### *4.1. Project Overview and Modeling Process*

The modeling method proposed in this paper and the developed software tool were applied to the case study at hand. This study provided an illustration of an assembly residential project in Hongshan District, Wuhan City. The project comprises 29 residences with a total floor area of 803,000 m$^2$ and 26 to 34 floors each, all of which were constructed using an assembled monolithic shear-wall structure. According to our statistics, there were a total of 3776 PC balcony components involved in the entire project. The fifth building was selected for the data-driven modeling method of this paper. Its structure comprises 26 above-ground floors and 1 underground floor, with floor space measuring 9780 m$^2$ and a monolithic prefabrication rate of 52%. There are five PC balconies on the standard floor.

Standard floor balconies were detailed in two drawings: balcony special drawings containing formwork drawings, and reinforcement drawings. The parameter dataset of the balcony was determined in accordance with the drawings. The parameter data for the drawing in the corresponding sub-window was then entered as follows.

1. Load the modeling tool to the Revit TAB and access the interface of the modeling tool.
2. According to the drawing information, select the beam balcony model, and upon the selection of 'Both sides form' from the drop-down menu, the corresponding parameter set interface button appears automatically.

3. Click the corresponding button and input data in the respective sub-interface that appears.
4. Select the reference coordinate point for generating the balcony component. The driver generates the PC balcony component at the corresponding position to complete the establishment of a single model.
5. Repeat the aforementioned steps to finalize the installation of the remaining PC balcony components on the standard floor. The modeling outcomes of the standard layer are illustrated in the subsequent Figure 12.



Properties

3D View

3D View: {3D} Edit Type

| | |
|---|---|
| **Graphics** | |
| View Scale | 1 : 100 |
| Scale Value ... | 100 |
| Detail Level | Fine |
| Parts Visibility | Show Original |
| Visibility/Gra... | Edit... |
| Graphic Displ... | Edit... |
| Discipline | Coordination |
| Show Hidden... | By Discipline |
| Default Analy... | None |
| Sun Path | □ |
| **Extents** | |
| Crop View | □ |
| Crop Region ... | □ |
| Annotation C... | □ |
| Far Clip Active | □ |
| Far Clip Offset | 304800.0 |
| Section Box | □ |
| **Camera** | |
| Rendering Se... | Edit... |
| Locked Orien... | □ |
| Perspective | □ |
| Eye Elevation | 20295.8 |
| Target Elevati... | 1180.0 |
| Camera Posit... | Adjusting |
| **Identity Data** | |
| View Template | <None> |
| View Name | {3D} |
| Dependency | Independent |
| Title on Sheet | |

Properties help Apply

Click to select, TAB for alternates, CTRL adds, SHIFT unselects.

**Figure 12.** Establishment of balconies on standard floors.

## *4.2. Application Results*

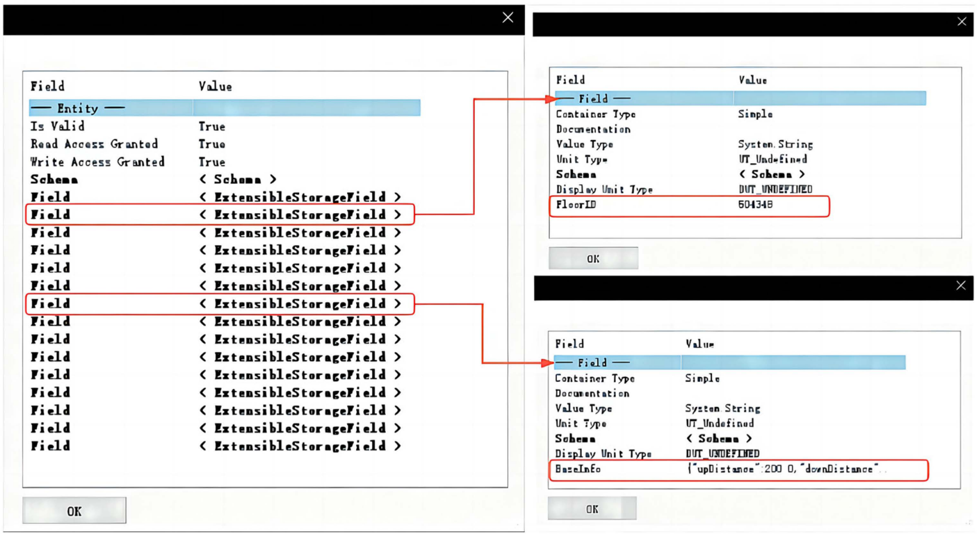
The aforementioned modeling steps are not only applicable to this residential building, but also extend to the establishment of prefabricated models for all residential buildings within this project. The parameters should be adjusted and the appropriate type of PC balcony selected based solely on the corresponding design drawing information. Following the same operational process, the model can then be established. This method is universally applicable and flexible, capable of accommodating diverse residential building projects.

Through the utilization of the software tool developed in this study, PC balcony component models were established for the entire project. The total number of PC balcony components in the project amounts to 3776, comprising 2772 prefabricated beam balconies and 1004 prefabricated slab balconies. The modeling tool is capable of generating 2607 beam balcony components and 905 slab balcony components. A total of 3512 PC balcony component models were successfully completed, resulting in an overall modeling completion rate of approximately 93.01%, as shown in Table 5. The data demonstrate that the modeling method used in this paper effectively accomplishes the construction of the majority of prefabricated balcony components.

**Table 5.** Statistical results of modeling completion.

| Type | Number of Components (PCS) | Number of Components That Can Be Generated (PCS) | Completion Rate (%) |
|---|---|---|---|
| Beam balcony | 2772 | 2607 | 94.04% |
| Slab balcony | 1004 | 905 | 90.14% |
| Total | 3776 | 3512 | 93.01% |

We verified the stability and data reliability of the data structure designed by the modeling method in this paper. The query interface, as depicted in Figure 13, allows for accessing information related to the balcony. Through this interface, module datasets and values within the data structure of the balcony component model can be visually examined. By querying the module datasets and data values in the data structures of 3512 balcony component models generated using the modeling tool, we compared the error between the query results and the input values. The comparison results are presented in Table 6, indicating a 100% success rate in querying the complete module dataset within the data structure. Furthermore, we observed a maximum error range of ±0.35% between the queried data values and their corresponding inputs, which confirms both the high reliability of our dataset and the improved informatization degree of our model. The specific data can be viewed in Figure 14. This approach effectively addresses any discrepancies caused by data deviation.



```
Field                     Value
── Entity ──
Is Valid                  True
Read Access Granted       True
Write Access Granted      True
Schema                    < Schema >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
Field                     < ExtensibleStorageField >
```

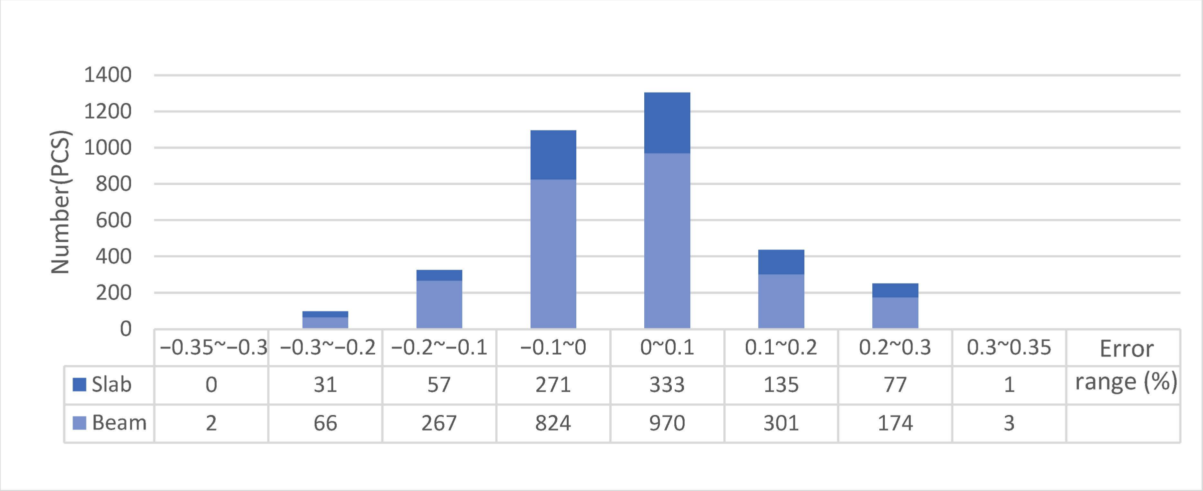
```
Field                     Value
── Field ──
Container Type            Simple
Documentation
Value Type                System.String
Unit Type                 UT_Undefined
Schema                    < Schema >
Display Unit Type         DUT_UNDEFINED
FloorID                   504348
```

```
Field                     Value
── Field ──
Container Type            Simple
Documentation
Value Type                System.String
Unit Type                 UT_Undefined
Schema                    < Schema >
Display Unit Type         DUT_UNDEFINED
BaseInfo                  {"upDistance":200.0,"downDistance"..
```

**Figure 13.** WinForms, which uses the Lookup function to inquire about data.

**Table 6.** Statistical results of data readings.

| Type | Number of Models Generated (PCS) | Number of Components That Can Read Data (PCS) | Data Readability Rate (%) |
|---|---|---|---|
| Beam balcony | 2607 | 2607 | 100% |
| Slab balcony | 905 | 905 | 100% |
| Total | 3512 | 3512 | 100% |



| | −0.35~−0.3 | −0.3~−0.2 | −0.2~−0.1 | −0.1~0 | 0~0.1 | 0.1~0.2 | 0.2~0.3 | 0.3~0.35 | Error range (%) |
|---|---|---|---|---|---|---|---|---|---|
| Slab | 0 | 31 | 57 | 271 | 333 | 135 | 77 | 1 | |
| Beam | 2 | 66 | 267 | 824 | 970 | 301 | 174 | 3 | |

**Figure 14.** Data error distribution diagram of balcony components.

In order to validate the modeling efficiency of the developed tool in this study for practical projects, a comparison was made between the conventional modeling method and the data-driven modeling method proposed herein for prefabricated balcony components of 5#. The average time spent on modeling and model quality were assessed. The results of the modeling time comparison are presented in the subsequent Table 7. By conducting a comparative analysis, it became evident that the use of the proposed modeling method in this paper can lead to a reduction in the average modeling time and an enhancement in modeling efficiency. The quality of the model was assessed using the level of development (LOD) accuracy level definition established by the BIMForum group of the Associated General Contractors of America (AGC). As shown in Table 8, the model established using the proposed method can reach an accuracy level of LOD400, surpassing the model established using the traditional method. This enhances the quality of the model and resolves challenges in modeling PC balcony components.

**Table 7.** Comparison of average modeling time.

| | **Average Modeling Time of Beam Balcony (min)** | **Average Modeling Time of Slab Balcony (min)** |
|---|---|---|
| Traditional method | 22 | 18 |
| Data-driven modeling method | 5 | 4 |
| Time saving | 17 | 14 |

**Table 8.** Evaluation of model LOD accuracy level.

| **LOD Accuracy Level** | **LOD100** | **LOD200** | **LOD300** | **LOD350** | **LOD400** | **LOD500** |
|---|---|---|---|---|---|---|
| | **Conceptual Model** | **Approximate Geometry** | **Precise Geometry** | **Construction Documentation** | **Production and Construction Information** | **As Built** |
| Traditional method | √ | √ | √ | √ | × | × |
| Data-driven modeling method | √ | √ | √ | √ | √ | × |

Note: The symbol "√" in the table indicates that the method satisfies the model accuracy requirements of the LOD accuracy level, while "×" signifies that the method fails to meet the model accuracy requirements of the LOD accuracy level.

## 5. Discussion

Currently, the research on precast component models primarily focuses on major structural elements, such as precast walls, slab, and columns, with relatively limited attention given to precast balcony components. The existing research also incorporates modular and DFMA methods for component splitting and design, although the focus primarily lies on the manufacturing and installation aspects of prefabricated components, with limited exploration into the methodology of component modeling [32]. This study addressed the challenge in complex modeling for precast balcony components by proposing an innovative modeling approach that enhances the efficiency of traditional methods. A data structure was established through the analysis of modularization. The concrete algorithm for reinforcement modeling was elaborated in detail, and a modeling tool was developed to address the lack of direct generation function for balcony components in 3D modeling software. The practical application of the proposed modeling tool and the analysis of modeling results confirm its ability to enhance modeling efficiency. Additionally, they validate the high reliability of the data structure established in this paper, thereby providing dependable data for future production. Therefore, the modeling research on prefabricated balcony components holds significant practical significance and promotional value.

The modeling method used in this paper enhances the efficiency of prefabricated balcony component modeling. However, it is important to acknowledge that the research methodology used herein has certain limitations. First, the datasets and data structures mentioned here may prove highly effective in the Chinese context, yet their direct applicability to modeling prefabricated balcony members in other countries remains uncertain. The variation in building codes, standards, and technical requirements across countries necessitates adaptation and optimization to align with the specific standards of each country or region. Second, the modeling tool used in this study was developed on the Revit platform, and optimization of the data structure may be necessary due to variations in data formats used by different 3D modeling software platforms for storing model information. Additionally, the diverse operation methods and public APIs used by different software platforms result in compatibility issues and unsuitability during programming code-automated operations. Future research can apply the modeling ideas in this paper to other modeling software for further optimization of the data structure and enhancement of its universality, thereby facilitating better adaptation to a wider range of engineering practices and production requirements. This will significantly contribute to improving the efficiency and accuracy of prefabricated balcony component modeling, as well as fostering advancements in related fields. Moreover, investigating how these modeling concepts can be effectively applied to more intricate assembly components warrants further exploration.

## 6. Conclusions

This study proposed a data-driven modeling approach for prefabricated balcony components, developed a software tool for modeling, and ultimately applied it to an actual project. Based on the aforementioned research, we derived the following findings:

1. The combination of data-driven modeling methods and BIM technology enhances the efficiency and quality of modeling for PC balcony components. In terms of modeling time, the beam balcony and slab balcony modeling times reduce by 17 and 14 min, respectively, compared to the traditional modeling method. In terms of model quality, the LOD accuracy level is used to assess the model's quality. The proposed method achieves a model accuracy level of LOD400, surpassing the LOD350 accuracy level achieved using the traditional method, thereby validating the efficacy of this novel approach in enhancing modeling efficiency.
2. In this study, a modeling software tool for PC balcony model was developed based on the Revit software platform, which used a data driver program to establish the model. This software was successfully applied in an actual project, achieving a completion rate of 93.01% for PC balcony components. Consequently, it effectively addresses the issue of 3D modeling software's inability to directly generate such components and significantly reduces operational complexities for modelers.
3. The study adopted a data-driven modeling approach and used a modular strategy to divide the PC balcony components. It also designed the corresponding module

dataset and model data structure. By analyzing the model data and statistically evaluating the error values, a 100% data reading rate was achieved, with a maximum error within ±0.35%. This confirms the reliability of the designed data structure stability in this study and effectively resolves any discrepancies between the data and the model caused by deviations. It serves as a dependable data source for the production of prefabricated components, mitigating the challenges arising from scattered information sources in the model and enhancing the level of model information.

The modeling method used in this paper enhances the efficiency of prefabricated balcony component modeling, while the developed software tool fills a void by providing direct generation functionality for balcony components in 3D modeling software. The results of this study can offer insights for the development of models for other complex prefabricated components and the advancement of modeling software, while it can also offer valuable references for subsequent research and practice in the related fields of prefabricated component manufacturing, design, and construction management. The research method used in this paper, however, exhibits certain limitations. First, the module dataset and data structure used in this study are specifically tailored to conform with China's domestic standards for PC balcony components, thereby limiting their applicability when modeling similar components in other countries. Second, the modeling tool used in this study was developed on the Revit platform, which may possess limited compatibility with other 3D modeling software. Future research can extend the application scope and field of this paper by implementing the modeling approach in other software platforms, while also optimizing the data structure for enhanced efficiency.

**Author Contributions:** Conceptualization, X.W. and J.S.; methodology, J.C. and X.W.; software, X.W. and X.X.; validation, X.W., X.X. and Y.F.; investigation, J.C., X.W. and Y.W.; data curation, X.W. and Y.F.; writing—original draft, X.W.; writing—review and editing, J.C. and J.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The data used to support the findings of this study are available from the corresponding author upon request. The data are not publicly available due to its ownership by a partner organization and adherence to commercial confidentiality agreements.

**Conflicts of Interest:** Author Xingxing Xie was employed by the company Beijing Olive Hill Software Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Cao, X.; Li, X.; Zhu, Y.; Zhang, Z. A Comparative Study of Environmental Performance between Prefabricated and Traditional Residential Buildings in China. *J. Clean. Prod.* **2015**, *109*, 131–143. [CrossRef]
2. Liu, W.; Zhang, H.; Wang, Q.; Hua, T.; Xue, H. A Review and Scientometric Analysis of Global Research on Prefabricated Buildings. *Adv. Civ. Eng.* **2021**, *2021*, 8869315. [CrossRef]
3. Banihashemi, S.; Tabadkani, A.; Hosseini, M.R. Integration of Parametric Design into Modular Coordination: A Construction Waste Reduction Workflow. *Autom. Constr.* **2018**, *88*, 1–12. [CrossRef]
4. Kamali, M.; Hewage, K. Life Cycle Performance of Modular Buildings: A Critical Review. *Renew. Sustain. Energy Rev.* **2016**, *62*, 1171–1183. [CrossRef]
5. Tam, V.W.Y.; Fung, I.W.H.; Sing, M.C.P.; Ogunlana, S.O. Best Practice of Prefabrication Implementation in the Hong Kong Public and Private Sectors. *J. Clean. Prod.* **2015**, *109*, 216–231. [CrossRef]
6. Yang, S.; Hou, Z.; Chen, H. Network Model Analysis of Quality Control Factors of Prefabricated Buildings Based on the Complex Network Theory. *Buildings* **2022**, *12*, 1874. [CrossRef]
7. Yuan, Z.; Sun, C.; Wang, Y. Design for Manufacture and Assembly-Oriented Parametric Design of Prefabricated Buildings. *Autom. Constr.* **2018**, *88*, 13–22. [CrossRef]
8. Li, Y.; Ni, Y.; Zhang, N.; Liu, Z. Modularization for the Complex Product Considering the Design Change Requirements. *Res. Eng. Des.* **2021**, *32*, 507–522. [CrossRef]
9. Qi, B.; Costin, A. BIM and Ontology-Based DfMA Framework for Prefabricated Component. *Buildings* **2023**, *13*, 394. [CrossRef]
10. Dong, D.; Zou, Y.; Pan, H.; Zhou, G.; Feng, Y.; Tang, Y. DFMA-Oriented Modular and Parametric Design and Secondary Splitting of Vertical PC Components. *Sci. Rep.* **2023**, *13*, 3457. [CrossRef]

11. Choi, J.O.; O'Connor, J.T.; Kwak, Y.H.; Shrestha, B.K. Modularization Business Case Analysis Model for Industrial Projects. *J. Manag. Eng.* **2019**, *35*, 04019004. [CrossRef]
12. Mansuri, D.; Chakraborty, D.; Elzarka, H.; Deshpande, A.; Gronseth, T. Building Information Modeling Enabled Cascading Formwork Management Tool. *Autom. Constr.* **2017**, *83*, 259–272. [CrossRef]
13. Marzouk, M.; Daour, I.A. Planning Labor Evacuation for Construction Sites Using BIM and Agent-Based Simulation. *Saf. Sci.* **2018**, *109*, 174–185. [CrossRef]
14. Wang, Y.; Thangasamy, V.K.; Tiong, R.L.K.; Zhang, L. Improved Workflow for Precast Element Design Based on BIM and Lean Construction. *J. Constr. Eng. Manag.* **2022**, *148*, 04022065. [CrossRef]
15. Bakhshi, S.; Chenaghlou, M.R.; Pour Rahimian, F.; Edwards, D.J.; Dawood, N. Integrated BIM and DfMA Parametric and Algorithmic Design Based Collaboration for Supporting Client Engagement within Offsite Construction. *Autom. Constr.* **2022**, *133*, 104015. [CrossRef]
16. Li, M.; Wong, B.C.L.; Liu, Y.; Chan, C.M.; Gan, V.J.L.; Cheng, J.C.P. DfMA-Oriented Design Optimization for Steel Reinforcement Using BIM and Hybrid Metaheuristic Algorithms. *J. Build. Eng.* **2021**, *44*, 103310. [CrossRef]
17. Tan, T.; Mills, G.; Hu, J.; Papadonikolaki, E. Integrated Approaches to Design for Manufacture and Assembly: A Case Study of Huoshenshan Hospital to Combat COVID-19 in Wuhan, China. *J. Manag. Eng.* **2021**, *37*, 05021007. [CrossRef]
18. Honghong, S.; Gang, Y.; Haijiang, L.; Tian, Z.; Annan, J. Digital Twin Enhanced BIM to Shape Full Life Cycle Digital Transformation for Bridge Engineering. *Autom. Constr.* **2023**, *147*, 104736. [CrossRef]
19. Chen, J.; Hu, R.; Guo, X.; Wu, F. Building Information Modeling-Based Secondary DevelopmentSystem for 3D Modeling of Underground Pipelines. *Comput. Model. Eng. Sci.* **2020**, *123*, 647–660. [CrossRef]
20. Tang, F.; Ma, T.; Guan, Y.; Zhang, Z. Parametric Modeling and Structure Verification of Asphalt Pavement Based on BIM-ABAQUS. *Autom. Constr.* **2020**, *111*, 103066. [CrossRef]
21. Girardet, A.; Boton, C. A Parametric BIM Approach to Foster Bridge Project Design and Analysis. *Autom. Constr.* **2021**, *126*, 103679. [CrossRef]
22. Sheik, N.A.; Veelaert, P.; Deruyter, G. Exchanging Progress Information Using IFC-Based BIM for Automated Progress Monitoring. *Buildings* **2023**, *13*, 2390. [CrossRef]
23. Tirella, V.; Fabbricatore, C.; Carpino, C.; Arcuri, N.; Barreca, F. Configuration Optimization for Sustainable Temporary Houses Employing BIM Procedure. *Buildings* **2023**, *13*, 2728. [CrossRef]
24. Meng, W.; Zhang, H.; Ai, Q.; Bao, T.; Yan, J. CBR-RBR Fusion Based Parametric Rapid Construction Method of Bridge BIM Model. *Adv. Eng. Inform.* **2023**, *57*, 102086. [CrossRef]
25. Liu, H.; Cheng, J.C.P.; Gan, V.J.L.; Zhou, S. A Novel Data-Driven Framework Based on BIM and Knowledge Graph for Automatic Model Auditing and Quantity Take-Off. *Adv. Eng. Inform.* **2022**, *54*, 101757. [CrossRef]
26. Kirchdoerfer, T.; Ortiz, M. Data-Driven Computational Mechanics. *Comput. Methods Appl. Mech. Eng.* **2016**, *304*, 81–101. [CrossRef]
27. Raco, F.; Stefani, M.; Balzani, M.; Ferrari, L. Towards effective project documentation, transparency, and data-driven decision-making through bim-blockchain based applications. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2021**, *46*, 437–444. [CrossRef]
28. Wu, Z.; Wang, Q.; Hu, J.; Tang, Y.; Zhang, Y. Integrating Model-Driven and Data-Driven Methods for Fast State Estimation. *Int. J. Electr. Power Energy Syst.* **2022**, *139*, 107982. [CrossRef]
29. Diao, Y.; Shwartz, L. Building Automated Data Driven Systems for IT Service Management. *J. Netw. Syst. Manag.* **2017**, *25*, 848–883. [CrossRef]
30. Onyszkiewicz, J.; Sadowski, K. Proposals for the Revitalization of Prefabricated Building Facades in Terms of the Principles of Sustainable Development and Social Participation. *J. Build. Eng.* **2022**, *46*, 103713. [CrossRef]
31. Wang, L.; Zhao, B.; Ye, Q.; Feng, A.; Feng, W. A BIM-Based Data-Driven Modeling Method. In *ICCREM 2021: Challenges of the Construction Industry under the Pandemic*; American Society of Civil Engineers: Beijing, China, 2021; pp. 319–330.
32. Lu, W.; Tan, T.; Xu, J.; Wang, J.; Chen, K.; Gao, S.; Xue, F. Design for Manufacture and Assembly (DfMA) in Construction: The Old and the New. *Archit. Eng. Des. Manag.* **2021**, *17*, 77–91. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.