

APPLICATION OF BIM IN THE CREATION OF PREFABRICATED STRUCTURES LOCAL PARAMETERIZED COMPONENT DATABASE

Qinghan Bai¹, Sihua Deng¹, Chenguang Li², Ze Qie¹

¹ Beijing University of Civil Engineering and Architecture
Zhanlanguan Rd, 1, Dong Wu Yuan, Beijing, China

² Beijing Building Construction Research Institute Co.
Beijing 100084, China

¹ Corresponding author: 13366015526@163.com

Abstract

Introduction: As a research focus in the construction industry in recent years, prefabricated structures have advantages such as environmentally friendly and economical and so on, but it also increases the complexity of design and construction and the strict requirements for accuracy. **Methods:** Aiming at the shortages of prefabricated structures in design and construction, one solution is creating a local parameterized component database. Collecting the information of the required prefabricated components firstly, and based on this, we put forward a method of creating local prefabricated component database, including secondary development on API and customize parameterized component database, which based on the template file. **Results and discussion:** Combine BIM technology and prefabricated structures better to support the development of prefabricated structures.

Keywords

Prefabricated structure, BIM, Local parameterized component database.

Introduction

At present, the pre-assembly simulation works of prefabricated structures are remains to be studied. The BIM use for cost and project management has received more attention in the last few years and has now become a main focus of project owners (Feng, 2013). Although BIM has made great contributions to the construction industry so far, most of existing software tools are developed in the context of traditional non-prefabricated buildings and do not well take into account the new process that building components are produced in factory and moved to construction site for assembly (Leminen et al., 2013). Since the BIM software used by many domestic design institutes is mostly imported foreign products, there is a problem of software localization, and the parameterized component database should be localized as a core part of the software.

The first thing is collecting the data of components. Collecting detailed information of local prefabricated components from standard design atlas, design institute and prefabricated component factory, etc., to prepare for the localization of BIM software. The information of prefabricated components includes component size, material, price and thermal insulation performance, which lays a foundation for later information processing and analysis.

BIM serves as a useful platform for facilitating the on-site assembly services (OAS) of prefabricated

construction for its benefits of providing collaborative working teams and decision makers with the physical and functional representations of prefabricated component (Chen et al., 2015). After collecting the product data from the component factory, parameterized component database in the modeling software can be localized. The process of localization is mainly established by secondary development or based on internal parameterized component database in BIM software. Since the software was designed to consider the expansion and compatibility issues in the early stage of development, this process can be directly performed. This component database is parameterized with well compatibility, so all kinds of components can be evolved by several basic types of components.

Since there is still no software that can contain all the building information, the common method is: basing on the Industry Foundation Classes (IFC) data model. IFC is a standard format which is open to the BIM data that are exchanged and shared among the software applications used by the participants to building sector or facility management (Richard, Jon, 2010). The industry foundation classes (IFC) standard achieves the data interchange between BIM tools (Wang et al., 2015). Using BIM software and BIM peripheral auxiliary software as the tool, coordinate multi-software and multi-specialty work, transfer data efficiently (Pei, 2010; Singh et al., 2011; Jung, Joo, 2011).

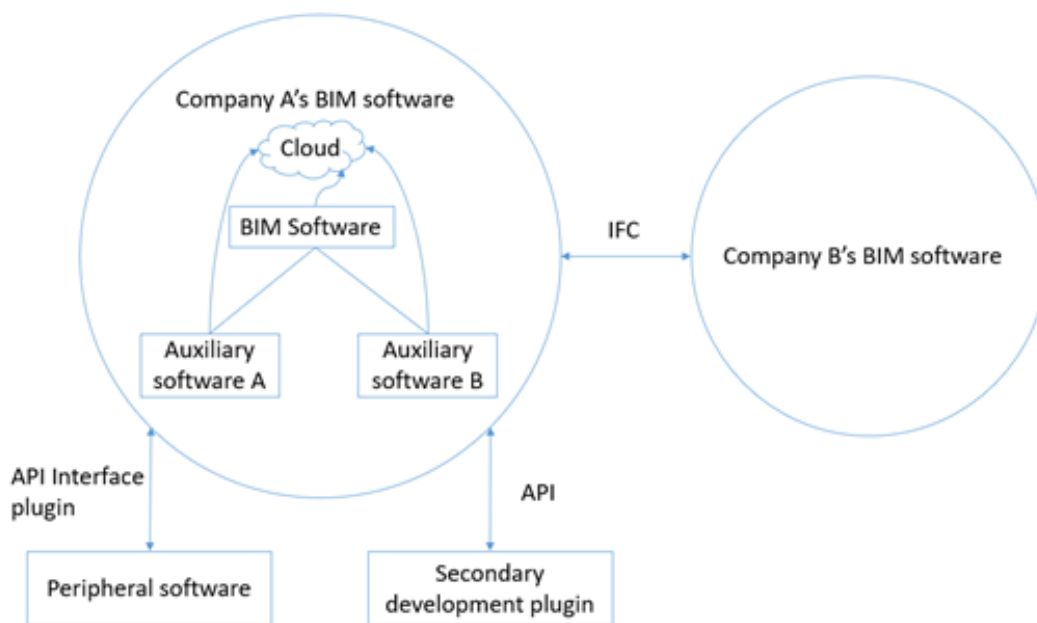


Figure 1. The collaboration relationship of BIM software.

The BIM software developed by IFC standard has good compatibility, and the data transmission between different companies are mainly relies on the IFC standard (Jeong et al., 2009; Lipman, 2009). Figure 1 shows the collaboration relationship of BIM software. The secondary development plug-in is compatible through the Application Programming Interface (API) interface, and peripheral software are compatible through the API interface plug-in. Usually, the building model and main building information are built by the core modeling software, but aiming at a certain specialty, the auxiliary software is needed to improve and transform the model.

Collect prefabricated component data

The component database is the core of the entire BIM modeling process and the entire lifecycle management of the building. Throughout a building's life-cycle this procedural transition is further expedited by BIM technology (Abanda et al., 2015). All project construction is realized by the combination of components. A prefabricated building is made up of many precast components, and these components are typically produced in a factory and transported to construction site for assembly [Kim et al., 2012; Kim et al., 2015; Hong et al., 2014]. The parameterized components help data management and modification become easier, and data analysis is also derived from the support of the component database, which can be said to be the most basic building unit of the structure model.

Based on the previous collection of local component data, the local component database can be established in two ways: one is to use the API interface secondary development plugin to achieve localization; the other is directly using the component template file that contains in BIM software, customize the parameterized component database that conforms to the local situation. Both have their own focus, the former can achieve more complex

and advanced functions, but it takes a lot of time and effort, while the latter is relatively simple and easy to use, and it is easy to quickly expand the component database. The process is conducive to promote the standardization, mechanization and informatization of construction and improve the environment of construction (Pons, Wadel, 2011).

The establishment of the local component database is based on the full investigation of the local products. the created precast components should also be saved to the standard parametric precast component library as this will simplify subsequent design works of prefabricated buildings (Arayici et al., 2011; Wang et al., 2016). In the investigation of the local prefabricated component products, various data information of the components needs to be collected, such as the information listed in table 1.

Table 1. Component information.

Geometric information	Other information
Component size (Length, width and height, etc.)	Material, Price
Spatial location (Prefabricated component placement, installation, etc.)	Insulation performance
Connection location (Connected, parallel and vertical, etc.)	Volumetric weight
.....	Application
	Identification and data

Secondary development by API

API means Application Programming Interface, which is an interface reserved for secondary development by the application, allows secondary developers to easily access the internal functions of the original program without knowing the details of its source code or its internal working system. In order to meet the different needs of

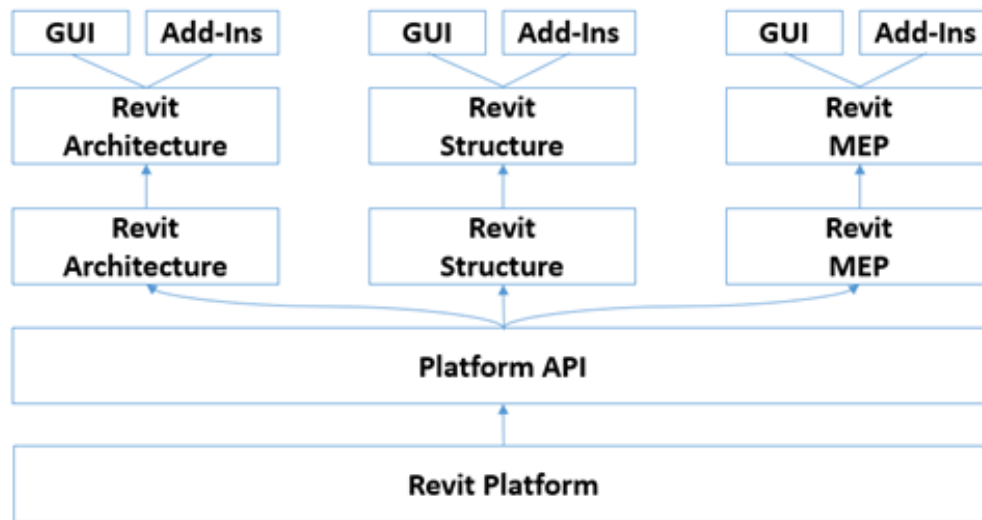


Figure 2. Schema-level diagram of BIM API theory.

various customers, the current mainstream BIM software has reserved rich and powerful API. We can integrate other applications through API or using API to redevelop our own applications. For example, the schema-level diagram of Revit is shown in Figure 2.

The bottom layer in Figure 2 is the development platform of the program. The code and functions internal the application cannot be accessed by external programs. It is the most basic part of the program. It mainly provides internal data access and graphic display functions. The upper layer is the core interface. Based on this, the Revit series of applications (Revit Architecture, etc.) are developed; on each application, the interface for the program is pre-reserved for the user, and through these interfaces, we can do further improve of the application that suits our needs.

Traditionally, split design of a prefabricated building is made by prefabricated-component factory after construction drawings are finished by architectural design firm (Fan, Li, 2014). By using the API, we can improve the efficiency of the operation of the BIM model: First, we can access the BIM model more precise, and the target object can be quickly found through the combination of the object filtering class function, Filtered Element Collector, and the filter condition. Secondly, the speed of modifying BIM model can be optimized by batch processing the model. By running all the commands in the API plugin and then performing a unified model update, the time lost by the original model real-time update is reduced. Finally, by using the transaction mode to ensure that the API program commits or discovers problems without error. Guarantee access performance and functionality better. At present, the application direction of API secondary development mainly focuses on the following aspects (Autodesk Asia Pte Ltd, 2012):

1. Replace some mechanical operations in the original program;
2. Develop interface for compatibility with other software;

3. Extract model data for data optimization analysis or export data reporting;

4. Link the external information of the converter to complete the drawing of the primitive or the change of the parameters;

5. Accomplish the automatic error checking of model file based on design specifications.

Secondary development of the Revit API requires the .NET development language, such as Visual Basic.

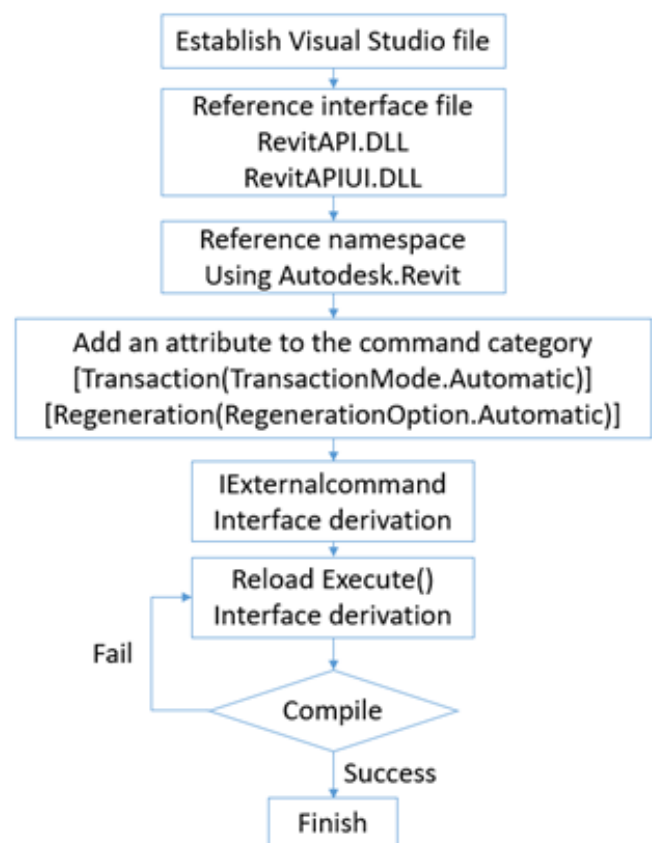


Figure 3. Basic flow of API secondary development.

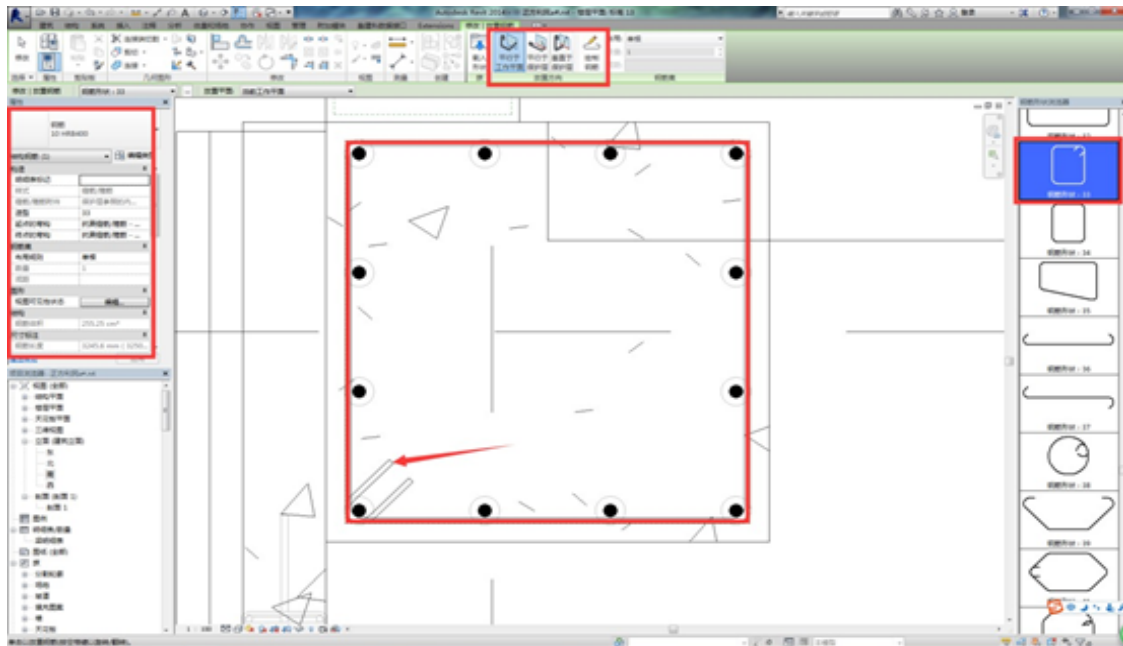


Figure 4. Arrangement method of reinforcement in the original program.



Figure 5. Operator interface of parametric arrangement of reinforcement.

NET, C# or C++/CLI. The secondary development plug-in can be divided into two types: external command plug-in (IExternalCommand interface), this plug-in loaded into the Revit program when Revit starts, created a Revit command by manual call; external program plug-in (IExternalApplication interface), which is automatically called by the corresponding Revit startup and shutdown events, used to add menus and toolbar or other initialization commands. In most cases, an external command plugin is used. The basic flow of API secondary development is shown in Figure 3.

At present, the improvement of the component database by API secondary development is mainly reflected in the rapid modeling based on the component database. The following part shows how to use the API plug-in to improve the modeling efficiency.

Taking Revit structure rebar modeling as an example, the layout of the rebar model in the original software is mainly manually selected on the 3D solid model section, with many control options and complicated operation (Figure 4), and the large amount of steel bars and the complexity of modeling leads to many problems in the applicability of this method: the modeling operation is complicated and inefficient, and the accuracy is low, etc. In response to solve the above problems, a parametric reinforcement interface plug-in has been developed based on the API.

The plug-in can avoid the problem of precise low result caused by manual selection on the original three-dimensional model, and at the same time, the various options are concentrated on one interface, and the operation is relatively simple and convenient. Improve the

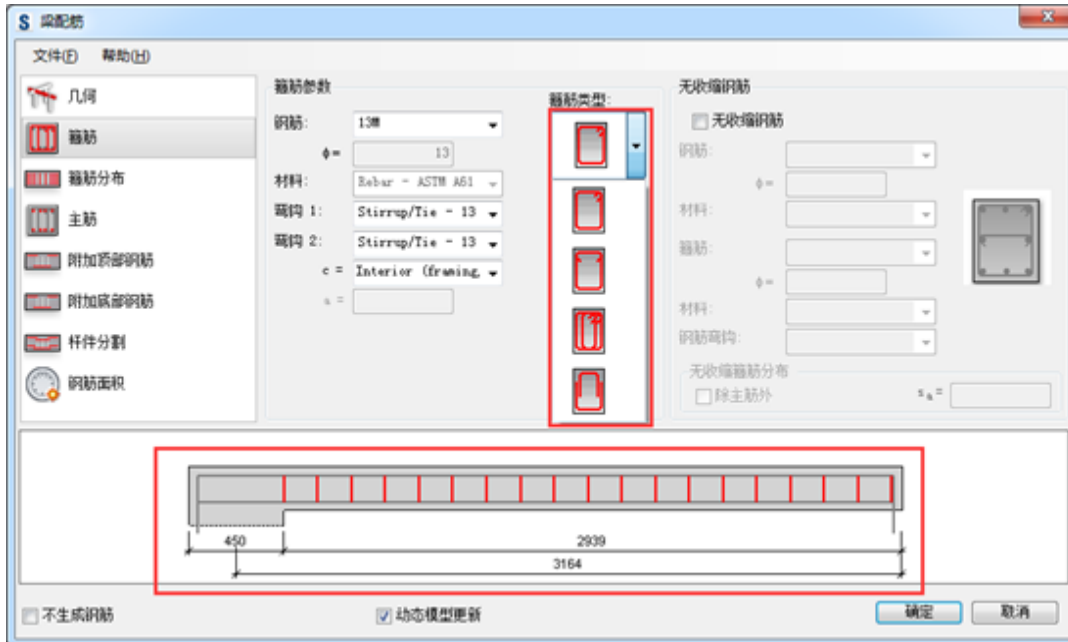


Figure 6. Parameter setup of hoops.

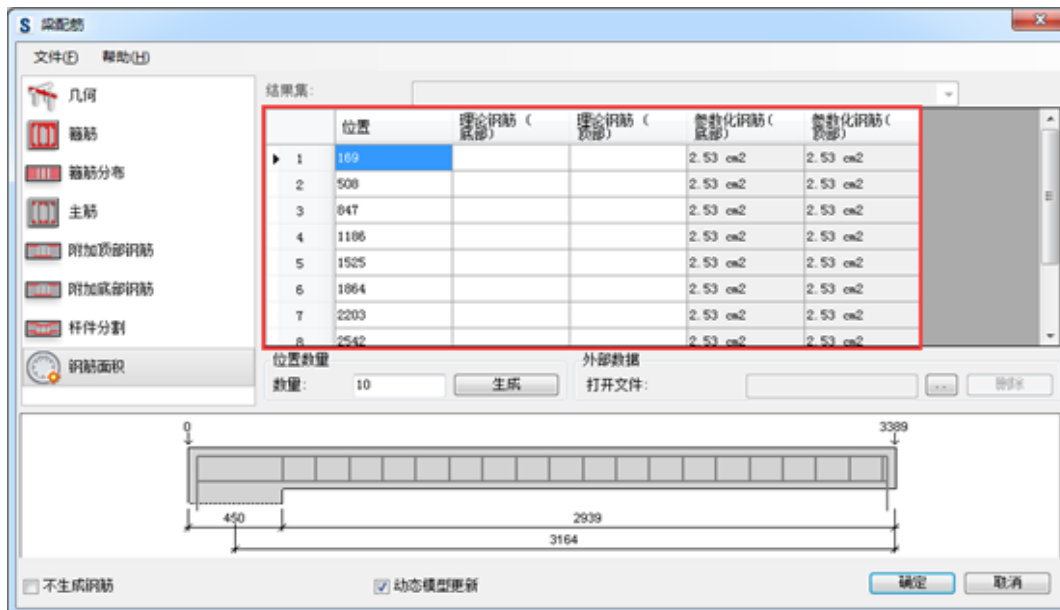


Figure 7. Bar area options.

efficiency of steel modeling (Figure 5). Any modifications in building design will be automatically reflected in other related parts (Park, 2011).

Taking the beam reinforcement as an example, the various options required for the reinforcement of the beam are listed on the left side of the interface. The first item is the geometric information, and the program automatically extracts the relevant geometric information of the member to be reinforced, including the family name, type, geometry parameters and component parameters, etc., the second to seventh items are the parameter options of the steel bar arrangement, rebars of various diameters, types, positions, configurations can be selected directly by adjusting parameters, and there are schematic diagrams on the lower side of the interface to assist us. Perform the

reinforcement operation (Figure 6). The last item is the statistics of the area and type of the reinforcement, which is convenient for later calculations (Figure 7).

Custom parameterized component database based on template file

Template file, which is the template that comes with the original program to create parameterized components.

The file basically covers the various settings and related graphical representation styles of the parameterized components, thus avoiding the complicated programming process of the secondary development of Application Programming Interface and facilitating the user to quickly establish corresponding components adapted to their own needs. Because the establishment of local parameterized

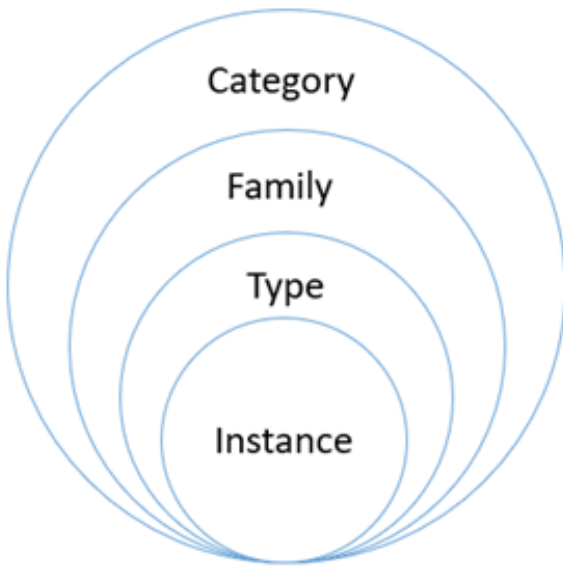


Figure 8. Hierarchy diagram of graphic element.

component database has a large workload, and the template file of the BIM modeling software can basically cover the required basic parameters, this paper mainly uses the method of this section to establish the local component database.

The following part is a demonstration of establishing a component database using Revit as an example.

Organization of primitives

The primitives, which is the various display units that exist in three or two dimensions in the BIM model, such as a column or an axis, can be referred to as a primitive. In Revit, we call the component "family", and the component database is called "family database". Autodesk's official interpretation of the "family" is: "family" is a set of primitives that contain a common set of parameters and corresponding graphical representations.

Some or all of the parameters of different primitives in the same group may have different values, but the set of parameters (its names and meanings) are same, and these variants in the family are called family types or types (Qiang, 2011; Huang, 2013).

In the real engineering project, the number of primitives in the project is extremely large. If the user does not make the corresponding rules, it will lead to confusion in data file management. Therefore, in Revit, all the primitives are logical according to a certain level. Organized together, the classification hierarchy is category, family, type and instance.

Categories are the highest level when categorizing primitives. Categories represent different sub-items of the building model. For example, foundations, columns, and beams in buildings are represented in Revit as three different categories: structural foundation, structural column, and structure framework, the category is preset in the program, and cannot be changed.

Categories are classified into different families. For example, "structural beam category" includes steel beam

family, cast-in-situ concrete beam family and precast concrete beam family, among which "precast concrete structure beam" can be classified into different families such as shapes, sizes, uses, etc. A family is a grouping of primitives based on the same use and similar graphical representation of the parameter set (Liu et al., 2009). Some attributes of different primitives in a family may have different values, but the settings of the attributes (the names and meanings) are the same (Qiang, 2011; Huang, 2013).

Family can be classified into different types. "Type" is a hierarchy that classified elements with different parameter values within a family. For example, different precast concrete beams may have several different sizes and shapes, so that in this family of precast concrete beams, there are several different types, such as 300mm * 600mm and 300mm * 700mm, etc. These specific types are different types of precast concrete beam families.

Finally, each "type" can be placed multiple times in the project file according to different settings, such that an item (single primitive) placed in the project becomes an instance of the type, the same one structural column can be placed in one layer or in three layers, which are two different examples.

The relationship of primitives is shown in Figure 8.

Family parameters

The family parameter is a variable that controls the common attribute of the component size, shape and material, etc. In the project, the attribute of the family can be adjusted or modified without using the family, which ensures the applicability of the parameterized component and the convenience of later data extraction.

When creating a local family database, selecting different family templates will have different family parameter settings. The setting of the family parameters reflects the user's intention to create the family's expected performance in the project. In Revit, all family changes are driven by family parameters.

Table 2. Family parameters.

Parameter Type	Description
Text	Enter characters to define literal parameters
Integer	A value expressed as an integer
Numerical value	Used for various digital data, is real number
Length	Create the length of the primitive or sub-component
Area	Create the area of the primitive or sub-component
Volume	Create the volume of the primitive or subcomponent
Angle	Create the angle of the primitive or subcomponent

Table continuation 2

Slope	Define the slope of the parameters
Currency	Define the currency of the parameters
URL	Provide a web link to a user-defined URL
Material	Specify parameters for a specific material
Yes/No	Use "yes" or "no" to define parameters that can be used in conjunction with conditional judgments.
Family type	Nested components, different family types can match different nested families

Test of the family

The main purposes of testing the component library are as follows:

First, make sure that the family file has good parametric performance. Testing the parameters and variables of the family file ensures that the family file has stable parametric performance in the actual project.

Secondly, ensure that it meets the domestic architectural design and the company's internal drawing specifications. With reference to the Chinese architectural design codes and atlases, as well as the company's internal drawing specifications for line types and legends, the display of family files under different views and precisions can be checked to ensure the final quality of the project documents.

Thirdly, achieve the unity of family file. The first two tests are the key to ensuring the quality of the family files, and they are indispensable. However, the test of the uniformity of family files does not directly affect the quality itself, but if you pay attention to the uniformity setting when creating family files, it will be very helpful for the management of family files. Moreover, after the family file is loaded into the project file, it will also bring some convenience to the creation of the project file:

On the one hand, ensure the unity of family documents and project templates. After the family file is loaded in the project file, the information that comes with the family file, such as "Material", "Fill Style" and "Line Style Pattern" will be automatically loaded into the project file. If the project file already contains information with the same name, the information in the family file will be overwritten by the project file.

Therefore, when creating a family file, we should try to refer to the information already in the project file. If there is a need to create a new file, the naming and settings should be consistent with the project file to avoid redundant information.

On the other hand, ensure the unity of the family files themselves. Some settings of the specification family file, such as insertion point, saved thumbnail, material, parameter naming, etc., will facilitate the management of

the family database and loading the project file to make the information contained in it uniform.

The testing process of the family can be summarized in one sentence: according to the requirements of the test document, the family files are tested one by one in the test project file environment, the family editor environment and the file browser environment, then a test report is generated. The main content of the test document includes four aspects: test purpose, test method, test standard and test report.

The main test process is as follows:

1. Formulate test documents: there are different test methods of different types family files, we can first classify the family files in 2D and 3D. Because the 3D family file contains many different family categories. The creation process of the partial family category, the family template function and the modeling method are highly similar.

For the 2D family file, the creation process and the family template function of the detailed component are typical, so based on this category, two-dimensional universal test document is formulated. "Title bar", "comment" and "contour" have certain specialties, we can add or delete some specific test content and make related documents based on two-dimensional universal test document.

For the 3D structure family, in addition to the parametric test and the uniformity test, some special settings in the structure family should be checked, because these settings are related to whether the structure family behaves correctly in the project. For example, check that whether the end points of the concrete structural beam are locked with the two reference planes "component left" and "component right" in the template, and so on.

Then, various types of structure families can be loaded into the project to check whether the family behavior is correct, for example, the connection of structural column and beam, which built in same material or different materials; analysis the model, check whether the steel protection layer is correct, etc.

2. Create test project files: For different categories of family files, we need to create corresponding project files during the test, simulate the call process of the family files in the actual project, and discover possible problems.

3. Test in the test project environment: Load the project file to check the display and performance of the family files in different views. Change the family file parameters and system parameter settings to check the parametric performance of the family file.

4. Test in the family editor environment (Liu et al., 2009): Open the family file in the family editor, check the unity between the family file and the project template, such as "material", "fill style" and "linear pattern". And check the unity between family files, such as insertion points, materials, parameter naming, etc.

5. Test in the file browser: In the file browser, observe the display of thumbnail and check whether the file size is within the normal range according to the file properties.

6. Complete the test report: Refer to the test criteria in the test document, mark the wrong items one by one,

and complete the test report to facilitate the next file modification.

Conclusions and future work

BIM technology and prefabricated structures are two of the most popular technology branches of the recent construction industry, injecting vitality into the construction industry. BIM technology is very suitable for the concept of detailing and industrialization of the prefabricated structure. This paper shows the establishment of local component database of the prefabricated structure based on Revit. As the basis of BIM technology, the establishment of local component database is a focus of localization of BIM technology.

This paper introduces the method of building Revit parameterized component database and studies the naming rules and testing of component library. Secondly,

through API, the form of secondary development plug-in, avoids the cumbersome modeling process of Revit's original steel modeling tool, and realizes the integration and parameterization of steel reinforcement modeling.

The combination of BIM technology and prefabricated structure is a complex and huge system engineering, focusing on the application of the whole life cycle of the building. This paper has done in-depth research and discussion on the basic problems in the system engineering, but further research is needed in the follow-up of the overall workflow.

Acknowledgement

This research is sponsored by BUCEA Urban Rural Construction and Management Industry Research Development Collaboration Post Graduate Training Centre.

References

- Abanda, F. H., Vidalakis, C., Oti, A. H. Tah, J. H. M. (2015). A critical analysis of building information modelling systems used in construction projects. *Adv. Eng. Softw.*, 90, pp. 183–201. DOI: 10.1016/j.advengsoft.2015.08.009.
- Arayici, Y., Coates, P., Koskela, L., Kagioglou, M., Usher, C., O'Reilly, K. (2011). Technology adoption in the BIM implementation for lean architectural practice. *Autom. Constr.*, 20 (2), pp. 189–195. DOI: 10.1016/j.autcon.2010.09.016.
- Autodesk Asia Pte Ltd (2012). *Autodesk Revit Structure*. Shanghai: Tongji University Press.
- Chen, K., Lu, W. S., Peng, Y., Rowlinson, S., Huang, G. G. Q. (2015). Bridging BIM and building: from a literature review to an integrated conceptual framework. *Int. J. Proj. Manag.*, 33 (6), pp. 1405–1416. DOI: 10.1016/j.ijproman.2015.03.006.
- Fan, Z., Li, X. (2014). BIM proposal of precast building design. *Archit. Tech.*, 6, pp. 68–76.
- Feng, Y. (2013). The Roadmap of Hongye BIM Software. In: *Proceedings of the 4th Computing in Construction Summit*, pp. 12–17.
- Hong, W. K., Lee, G., Lee, S., Kim, S. (2014). Algorithms for in-situ production layout of composite precast concrete members. *Autom. Constr.*, 41, pp. 50–59. DOI: 10.1016/j.autcon.2014.02.005.
- Huang, Y. (2013). *Preliminary Study on Green Building Information Integration Based on Revit*. Chongqing: Chongqing University.
- Jeong Y. S., Eastman, C. M., Sacks, R., Kaner, I. (2009). Benchmark tests for BIM data exchange of precast concrete. *Automation in Construction*, 18 (4), pp. 469–484. DOI: 10.1016/j.autcon.2008.11.001.
- Jung, Y. Joo, M. (2011). Building information modeling (BIM) framework for practical implementation. *Automation in construction*, 20 (2), pp. 126–133. DOI: 10.1016/j.autcon.2010.09.010.
- Kim, M. K., Cheng, J. C. P., Sohn, H., Chang, C. C. (2015). A framework for dimensional and surface quality assessment of precast concrete elements using BIM and 3D laser scanning. *Autom. Constr.*, 49, pp. 225–238. DOI: 10.1016/j.autcon.2014.07.010.
- Kim, S., Nussbaum, M. A., Jia, B. (2012). The benefits of an additional worker are task-dependent: assessing low-back injury risks during prefabricated (panelized) wall construction. *Applied Ergonomics*, 43 (5), pp. 843–849. DOI: 10.1016/j.apergo.2011.12.005.
- Leminen, V., Eskelinen, H. Matthews, S., Varis, J. (2013). Development and utilization of a DFMA-evaluation matrix for comparing the level of modularity and standardization in clamping systems. *Mechanika*, 6, pp. 711–715.
- Lipman, R. R. (2009). Details of the mapping between the CIS/2 and IFC product data models for structural steel. *Journal of Information Technology in Construction*, 14, pp. 1–13.
- Liu, T., Yang, Y., Guo, T. (2009). Method of creating parametric family in 3D design of cement factory. *Cement engineering*, 5, pp. 8–11.
- Park, J. (2011). BIM-based parametric design methodology for modernized Korean traditional buildings. *J. Asian Archit. Build. Eng.*, 10 (2), pp. 327–334. DOI: 10.3130/jaabe.10.327.
- Pei, G. (2010). BIM and BIM related softwares. *Journal of Civil Engineering Information Technology*, 4, pp. 110–117.
- Pons, O., Wadel, G. (2011). Environmental Impacts of Prefabricated School Buildings in Catalonia. *Habitat International*, 35 (4), pp. 553–563. DOI: 10.1016/j.habitatint.2011.03.005.
- Qiang, Yu. (2011). Core Design of CAD Data Platform Based on BIM. *New technology*, 9, pp. 90–93.
- Richard, A., Jon, W. (2010). Integrating acoustical data into the Building Information Modeling using Industry Foundation Classes. In: *Proceedings of 159th Meeting Acoustical Society of America/NOISE-CON 2010*, pp. 141–150.
- Singh, V., Gu, N., Wang, X. (2011). A theoretical framework of a BIM-based multi-disciplinary collaboration platform. *Automation in construction*, 20 (2), pp. 134–144. DOI: 10.1016/j.autcon.2010.09.011.
- Wang, J., Zhao, J., Zongyu, H. (2016). Review and thinking on development of building industrialization in China. *China Civ. Eng. J.*, 49, pp. 1–8.
- Wang, X., Yang, H., Zhang, Q. (2015). Research of the IFC-based transformation methods of geometry information for structural elements. *J. Intell. Robot. Syst.*, 79, pp. 465–473. DOI: DOI 10.1007/s10846-014-0111-0.