

# 基于操作的钢结构节点参数化建模系统研发

满延磊<sup>1</sup>, 吴 杰<sup>1</sup>, 张其林<sup>1</sup>, 朱燕新<sup>2</sup>

(1. 同济大学 土木工程学院, 上海 200092; 2. 上海同磊土木工程技术有限公司, 上海 200433)

**摘要:** 为改变当前主流的基于约束的参数化建模方式复杂、建模效率低下、用户很难掌握的情况,从操作记录出发设计了一套基于操作的钢结构参数化节点制作系统,讨论了基于操作的参数化建模与基于约束的参数化建模在建模方式、系统开发、底层逻辑等方面的区别,介绍了操作实现的原理,研究了参数化节点中逻辑信息的写入,实现了利用工业基础类文件来储存参数化信息的方式。基于操作的参数化建模与非参数化建模在建模方式上基本一致,与基于约束的参数化建模相比提升了参数化建模效率、实现了参数化信息的共享、减少了相似节点的重复制作工作量。最后用实例验证了基于操作的参数化建模方式的优点。

**关键词:** 钢结构; 节点; 参数化建模; 操作记录; 工业基础类(IFC); 系统研发

**中图分类号:** TP391.41

**文献标志码:** A

## Research and Development of Operation-Based Parametric Modeling System for Joints of Steel Structures

MAN Yanlei<sup>1</sup>, WU Jie<sup>1</sup>, ZHANG Qilin<sup>1</sup>, ZHU Yanxin<sup>2</sup>

(1. College of Civil Engineering, Tongji University, Shanghai 200092, China; 2. Shanghai Tonglei Civil Engineering Technology Co., Ltd., Shanghai 200433, China)

**Abstract:** To change the situation that the normal constraint-based parametric modeling method is complex, and the modeling efficiency is low, which is difficult for users to master, a parametric steel joint system based on operation was designed. The difference between operation-based parametric modeling and constraint-based parametric modeling in modeling method, system development and the underlying logic was discussed. The principle of operation was introduced. How to write the logical information into the parametric joint was studied and using industry foundation class(IFC) file to save the joint was realized. Operation-based parametric modeling and non-parametric modeling are

basically consistent in the modeling method. Compared with constraint-based parametric modeling, operation-based parametric modeling's efficiency is improved, the parametric information is able to be shared, and the redundant work of developing the similar joints is reduced. At last, the advantages of operation-based parametric system were introduced by examples.

**Key words:** steel structures; joint; parametric modeling; operation record; industry foundation classes (IFC); system development

对于钢结构计算机辅助设计(CAD)系统或者钢结构建筑信息模型(BIM)系统来说,节点模块的设计一直是系统的重点与难点<sup>[1]</sup>。钢结构节点种类繁多,任何一款钢结构软件都无法覆盖所有节点,因此一套成熟的钢结构节点系统应提供用户自行扩展节点库的功能。Tekla 软件中的参数化节点以及 Revit 软件中的族是目前市场上常见的可扩展的系统<sup>[2-4]</sup>。节点扩展系统的核心为参数化建模技术。通过参数化建模来制作新的节点,将参数化信息保存以便实现节点在系统中的重复使用。

参数化技术在 20 世纪 80 年代问世,经历将近 40 年的发展,参数化建模已经相对成熟,最为出名的有 Dassault System 推出的 SolidWorks<sup>[5]</sup>、PTC 推出的 Pro/ENGINEER<sup>[6]</sup>、Siemens PLM Software 推出的 Unigraphics NX(UG)<sup>[7-8]</sup>等软件。在以上软件中,参数化技术共分为以下 2 类。

(1) 与历史有关的参数化设计。此种方法主要用在几何造型领域,代表软件是 Pro/ENGINEER。几何约束系统的建模和求解严格依赖于约束系统的创建顺序,任意几何图元可由约束它的已知几何图元所确定<sup>[9]</sup>。

(2) 与历史无关的参数化设计(又叫变量化设

收稿日期: 2018-09-18

基金项目: 国家自然科学基金(51578404)

第一作者: 满延磊(1987—),男,博士生,主要研究方向为建筑信息模型、钢结构深化。E-mail: manyanlei@126.com

通信作者: 张其林(1962—),男,教授,博士生导师,工学博士,主要研究方向为土木工程信息技术、钢结构。E-mail: zhangqilin@tongji.edu.cn

计). 代表软件是 UG, 能够随设计意图更改几何约束间的依赖关系, 约束装配具有无序性, 它通常基于陈述式方法定义几何约束<sup>[10]</sup>.

以上 2 种参数化技术均以约束为核心, 即先添加图元, 然后通过定义图元之间的约束来形成一个固定的模型. 此种建模方式在机械领域取得了很大的成功, 但是土木行业的从业者并不熟悉此种基于约束的参数化建模方式. 究其原因, 参数化建模诞生于机械领域, 建筑相对于机械来说模型更加庞大、标准化程度更低、参数化描述更为复杂等. 目前, 大多数土木行业的从业者只熟悉 CAD 软件的非参数化建模方式. 以在钢结构深化领域处于垄断地位的 Tekla 软件为例, 其建模方式为非参数化设计<sup>[11]</sup>. Tekla 软件也推出了参数化节点制作功能, 此功能是基于约束的参数化建模. 经调研发现, 使用此功能的用户极少, 有 10 年以上 Tekla 软件使用经验的用户也反应此功能过于复杂, 正因如此, 在新版的 Tekla 软件中, 参数化节点的菜单入口已经取消. 除 Tekla 软件以外, 土木行业中的常用软件均为非参数化建模, 例如探索者、天正、PKPM 等. 另外, Revit 软件中通过族建模生成构件(比如梁、柱等), 此过程为参数化建模, 但在通过构件组装成整个模型时也为非参数化建模.

基于以上原因, 本文从操作记录的角度出发, 设计了一套与基于约束的参数化建模不同的参数化设计系统, 此系统的参数化建模方式与传统的非参数化三维建模方式一致, 以降低用户的学习成本. 同时没有复杂的约束求解, 更加适用于建筑的三维建模.

## 1 基于操作记录的参数化节点建模原理

### 1.1 基于操作记录的参数化建模概念

传统的 CAD 建模是一种面向过程的建模方式, 用户按照一定的逻辑顺序依次添加图元, 最终形成整个模型. 基于操作记录的参数化节点建模将整个节点的制作过程记录下来, 当对参数进行修改时, 将所记录的操作一步一步还原, 从而得到一个新的节点. 例如, 对于图 1 所示的钢结构节点来说, 典型的步骤如下: 切割杆件、增加板件及增加焊缝、增加螺栓, 对应于每一步, 参数化信息包括: 杆件切割的位置、板件的大小及位置、焊缝的形式、位置及高度、螺栓的位置及螺栓信息等. 将上述操作集合记录在参数化节点中, 当在做新的节点时, 依次根据记录的信

息执行杆件切割、增加板件及焊缝、增加螺栓等操作, 从而生成指定形式的节点.

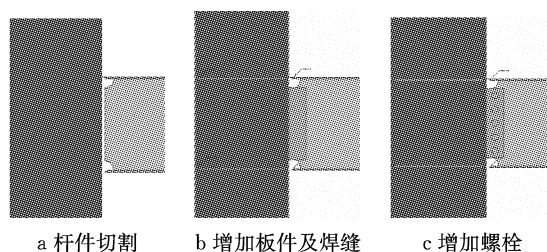


图 1 节点拼装步骤示例

Fig. 1 Instance of the process of joint design

### 1.2 约束与操作的对比

由操作的概念可以看出基于操作的参数化建模属于与历史有关的参数化建模方式. 但是操作与约束有很大的区别, 主要体现在以下 3 个方面.

(1) 建模过程. 以图 1 为例, 基于操作的参数化建模的过程依次为杆件切割、增加板件及焊缝、增加螺栓等, 建模过程与普通的 CAD 建模方式相同. 而基于约束的建模方式与 CAD 建模方式不同, 用户将板件、焊缝、螺栓等预先添加到模型中, 此时位置可以任意放置; 然后通过添加约束将图元放置在正确的位置, 可能的约束包括面与面重合、面与面平行、边线与面垂直、点到面的距离等. 由以上的对比可以看出, 在面向约束的建模中, 由约束来决定整个模型; 而面向操作的建模方式是面向过程的, 和传统建模方式一致, 需要一步一步地考虑整个模型的组装过程.

(2) 系统开发. 基于约束的参数化建模的重点和难点是约束的求解, 给定一组约束条件, 系统需要能够自动求解出满足此约束条件的结果; 基于操作的参数化建模的重点是参数化操作的开发, 没有约束求解这个过程.

(3) 底层逻辑. 约束本质是描述 2 个元素之间的关系, 因此约束是一对一的; 操作是基于已有的实体形成新的实体, 因此操作是有向关系, 同时操作可以基于多个已有实体的位置来形成新的实体, 因此操作是多对一的有向关系, 比如在新建螺栓的操作中, 需要基于已有的两块板的位置来对螺栓定位.

### 1.3 参数化节点的实现

#### 1.3.1 系统架构

系统的统一建模语言(UML)类图及 UML 时序图如图 2 和图 3 所示. 根据系统架构的单一功能原则, 功能划分得越细越有利于程序的维护, 因此将参数化节点划分为普通节点与参数化模型两大块. 普通节点是零件的集合, 零件包括节点板、螺栓、焊缝

等,模型中最终生成的为普通节点. 参数化模型储存操作集合. 普通节点通过生成节点调用参数化模型, 参数化模型通过更新模型依次调用操作集合, 操作

集合根据参数生成相应的零件, 最后将零件拷贝到节点中完成节点的制作.

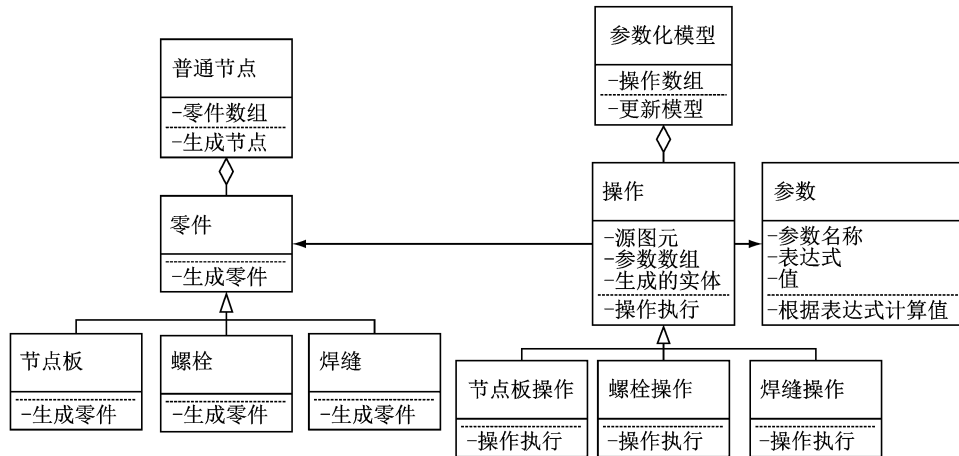


图2 统一建模语言类图

Fig.2 Class diagram in unified modeling language(UML)

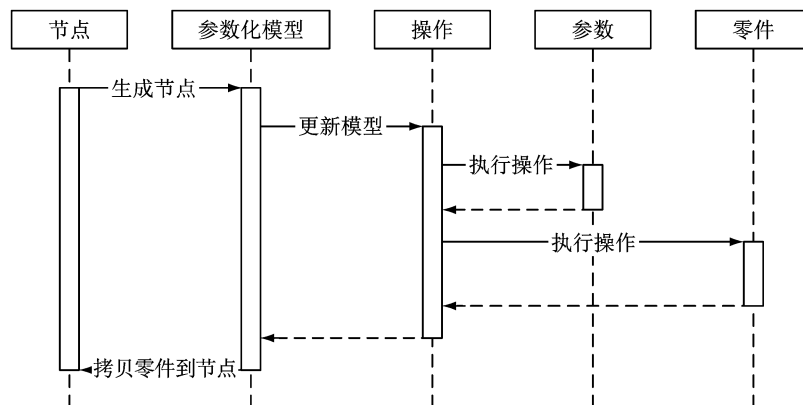


图3 统一建模语言时序图

Fig.3 Sequence diagram in unified modeling language(UML)

### 1.3.2 操作的表示

操作类的伪代码表示如下:

```
class operation
```

```
{
```

属性:

操作的源图元数组

操作的参数数组

生成的实体

方法:

执行操作

```
}
```

伪代码中的名词含意如下:

源图元: 为操作的执行提供几何参考定位.

操作参数: 操作执行时的几何参数或工程参数.

操作执行: 核心功能, 通过源图元的定位, 输入操作参数, 通过一系列的运算生成输出对象.

输出对象: 操作输出的实体, 为最终生成的实体.

在执行操作时, 通过源图元提供的参考定位, 读取参数数值, 然后根据一系列的运算生成输出实体, 如图4所示.

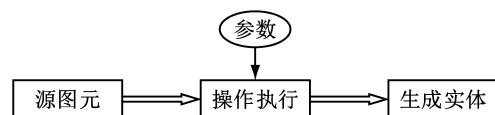


图4 操作基本原理

Fig.4 Principle of the operation

#### 1.3.2.1 源图元

在实际建模中, 实体的所有端点、边线、表面均可以提供几何参考定位, 因此实体的所有端点、边线、表面均可以充当源图元. 以图5为例, 在图5a中, 长方体有8个顶点、12条边、6个表面, 这些元素

均可以作为源图元,图 5b 中将长方体进行切割,新生成的实体有 10 个顶点、15 条边、7 个表面。图 5c 是图 5b 的一个变形,图 5c 并不属于长方体,但是在图 5c 中其图元的标识序号应该与图 5b 一样。如何对这些附属于实体的图元进行识别与标识是本系统的关键。

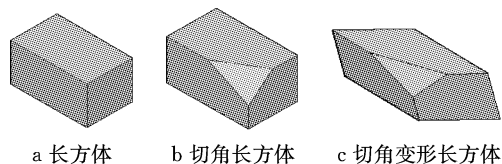


图 5 三维几何实体  
Fig.5 3D solid models

对于图元进行完整的识别涉及到模式识别领域,采用如下的简化方法进行:

(1) 确定实体的顶点、边线、面域。实体采用三角面片进行存储,相邻且共面的三角面片可以合并成更大的面域,最终形成图 5 长方体的外表面。面域的顶点、边线即为实体的顶点、边线。

(2) 顶点编号。在软件中,每个实体都有其生成方式,根据顶点生成的先后顺序进行编号。比如图 5b 和图 5c 的实体生成方式一致,共有以下 3 步:根据顶点生成面域、拉伸成实体、面切割。生成面域时,顶点编号根据新建顶点的顺序而定。拉伸实体时,拉伸实体的另外一面的编号根据基准拉伸面而定。面切割根据相交的边线的序号大小来确定相交点的序号。图 6 是顶点编号确定的过程。

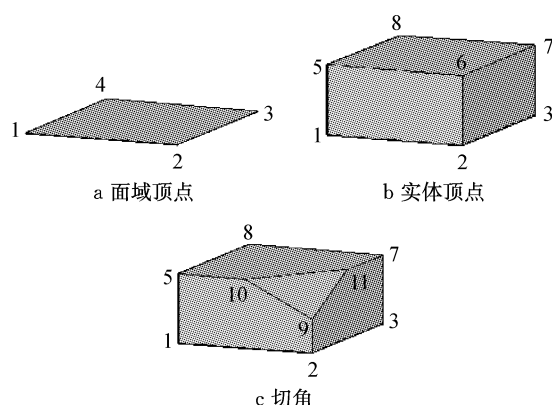


图 6 顶点编号的确定

Fig.6 Definition of the vertex's number

(3) 边线以及面域的编号。在图 6c 中,边线及面域可以用顶点编号进行如下表示。边线:1-2、1-4、1-5、2-3、2-9、3-4、3-7、4-8、5-8、5-10、7-8、7-11、9-10、9-11、10-11。面域:1-2-3-4、1-2-9-10-5、1-4-8-5、2-3-7-

11-9、9-10-11、5-10-11-7-8、3-4-8-7。将以上的所有图元的编号定义以下的偏序关系:①若图元  $\alpha$  中的顶点数目小于图元  $\beta$ ,则  $\alpha < \beta$ ;如果图元  $\alpha$  中的顶点数目大于图元  $\beta$ ,则  $\alpha > \beta$ 。因此由小到大排序为顶点、边线、面域;②当图元  $\alpha$  和图元  $\beta$  的顶点个数相同时,分别对图元  $\alpha$  和图元  $\beta$  内部的顶点序号从小到大排列,然后依次比较图元内部的顶点编号,若  $\alpha$  中第 1 个顶点编号小于  $\beta$  中第 1 个顶点编号,则  $\alpha < \beta$ ;若  $\alpha$  中第 1 个顶点编号大于  $\beta$  中第 1 个顶点编号,则  $\alpha > \beta$ ;若  $\alpha$  中第 1 个顶点编号等于  $\beta$  中第 1 个顶点编号,则比较第 2 个顶点,以此类推。由此可知,当比较面域 1-4-8-5 与面域 1-2-3-4 时,首先将面域内部的顶点编号排序,分别为 1-4-5-8 与 1-2-3-4;由于 1=1 且 4>2,因此面域 1-4-8-5 的偏序大于面域 1-2-3-4 的偏序。

当定义完此偏序关系之后,对所有图元从小到大进行排序,然后根据排序的位置分别赋予唯一的编号。上例的编号如下。① 顶点:顶点 1 至顶点 10 的编号分别为 1~10。② 边线:1-2 编号为 11,1-4 编号为 12,1-5 编号为 13, ..., 10-11 编号为 25。③ 面域:9-10-11 编号为 26,1-2-3-4 编号为 27,1-4-8-5 编号为 28,3-4-8-7 编号为 29,1-2-9-10-5 编号为 30,2-3-7-11-9 编号为 31,5-10-17-7-8 编号为 32。

在完成图元的编号之后,任意图元可以用“实体,图元在实体内部的编号”唯一标识,例如“实体 A,3”表示实体 A 内部的 3 号图元。定义源图元的标识符类的伪代码如下:

```
class OriginElement
{
    属性:
        实体指针
        图元在实体内部的编号
}
```

### 1.3.2.2 操作参数

操作参数将操作量化,操作参数是一系列的变量,通过更改参数的数值从而实现参数化。参数支持基本的数学运算。图 7 是一个参数列表图示。

### 1.3.2.3 操作执行

操作执行的核心是通过计算生成输出实体。相对于约束的求解,操作的计算一般都较为简单,均为普通的几何运算。例如图 8 所示的添加螺栓的操作,源图元包括:螺栓的上表面、下表面、定位点及  $y$  轴方向等;参数包括:螺栓的直径、行数、列数、边距、行间距和列间距等。操作计算只需要根据相应的条件

变量名	表达式	变量值	变量说明	值类型
jx	20	20	梁柱间隙	数值
t1	10	10	加劲板厚度	数值
a	45	45	切角半径	数值
bh	0	0	坡口钝边厚度	数值
ch	45	45	坡口角度	数值
jy	不加腋	0	是否加腋	枚举
h	350	350	加腋板高度	数值
l	500	500	加腋板长度	数值
t2	15	15	加腋板厚度	数值
t3	20	20	加腋底板厚度	数值
b	$lsbj*2+(lshs-1)*hjj$	200	板长	数值
c	$lsbj*2+(lsls-1)*hjj$	160	板宽	数值
t4	10	10	板厚	数值
ifdc	单剪	0	是否单剪	枚举

图 7 参数列表界面

Fig.7 Interface of the parameter list

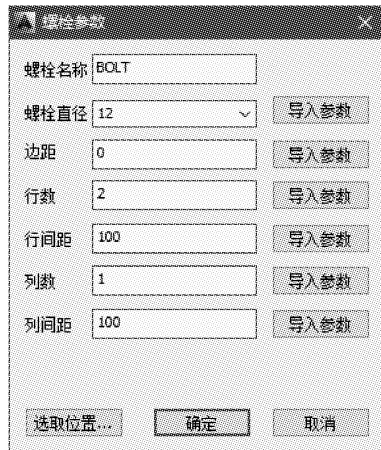
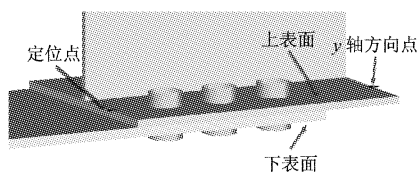


图 8 螺栓操作界面

Fig.8 Interface of the operation of bolt

计算出螺栓的坐标原点、螺栓的  $x$  轴、螺栓的  $y$  轴即可。用户在定义操作时,先选择上表面、下表面、定位点以及  $y$  轴上的点,然后在对话框中输入螺栓的参数。

表 1 是部分常用操作的源图元、参数情况。

表 1 常用操作的信息

Tab.1 Information of the common operations

类型	操作内容	源图元	参数
辅助点	延伸点	起点、终点	延伸距离
	线面交点	直线、相交面	
	相对点	基点	偏移坐标
辅助线	两点连线	起点、终点	
	平行线	基准线	偏移距离
三维实体	板件	板件端点	板厚
	螺栓	上表面、下表面、起始点、 $y$ 轴方向上的点	边距、行数、列数、行间距、列间距、直径
编辑	复制	源实体	偏移坐标
	旋转	源实体、旋转轴	旋转角度
	镜像	源实体、镜像平面	

### 1.3.2.4 生成实体

一般来说,每个操作都会生成一个实体,生成的实体可以是辅助点、辅助线或者是组成节点的实体比如节点板或者螺栓等。生成的实体可以为后面的操作提供定位。

综上所述,源图元是输入,操作执行和参数是联系纽带,生成的实体是输出,所有的操作通过彼此的逻辑关系相关联,最后形成了整个参数化模型。图 9 是一个参数化模型的简单示例:此模型中共有 4 个操作,操作 1 通过杆件 1 中的图元 3 定位生成实体 A;操作 2 通过杆件 1 中的图元 8 生成实体 B;操作 3 通过实体 A 中的图元 23 生成实体 C;操作 4 通过实体 A 中的图元 10、实体 B 中的图元 11 进行定位,生成实体 D。最终杆件和实体 A、实体 B、实体 C、实体 D 组成整个参数化模型。

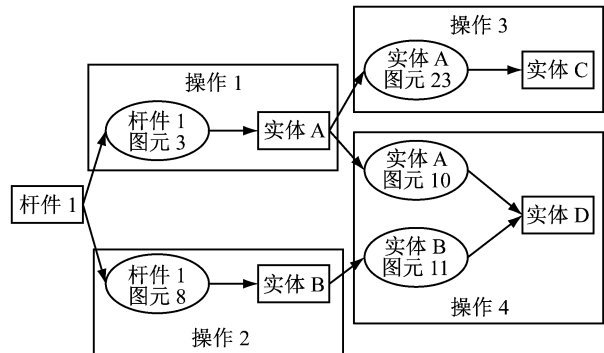


图 9 参数化模型示例

Fig.9 Instance of the parametric modeling

### 1.3.3 逻辑信息的添加

基于操作的参数化建模是与历史有关的建模方式,每一个操作具有前后的逻辑关系,而且每一个操作是多个具有内在联系的约束集合,因此每个操作都具有实际的意义。在建模过程中,可以根据相应的逻辑条件判断执行哪些操作,这样便可以实现将多个结构相似的模型用一个参数化模型表达,大大减少了重复节点制作的工作量。下面是一个简单的逻辑表达的伪代码示例,此逻辑判断可以将梁柱单剪节点与梁柱双剪节点融入到一个节点中。

if 单剪

增加连接板 1

以连接板 1 表面为螺栓上表面、以梁腹板表面为螺栓下表面增加螺栓

else if 双剪

增加连接板 1

镜像复制连接板 1 得到连接板 2

以连接板 1 表面为螺栓上表面、以连接板 2 表面为螺栓下表面增加螺栓

## 2 参数化节点的存储

完整的参数化节点设计系统包括操作设计以及节点存储. 参数化节点需要用单独的文件存储, 以便实现参数化节点的移植与分享, 即用户 A 制作的参数化节点能够分享给用户 B 直接使用. 工业基础类 (industry foundation class, IFC) 文件具有读写速度快、可读性强、可共享等特点<sup>[12]</sup>, 因此采用 IFC 文件存储参数化节点.

参数化节点的信息共分为 4 部分: 基本信息、实体、操作、变量列表.

### 2.1 基本信息的工业基础类存储

基本信息包括节点的名称、节点的类型名称、节点的匹配信息等. 在一个 IFC 文件中有且仅有一个参数化节点, 所有的其他信息均包含在参数化节点中, 因此将一个参数化节点看作一个 IfcBuilding, 参数化节点的信息作为属性附加在 IfcBuilding 上, 在 IFC 中, 通过 IfcRelDefinesByProperties 来附加属性. 基本信息的 IFC 表示方法如图 10 所示, 在图中实体箭头表示属性赋值, 箭头上的名称表示属性名称. IfcRelDefinesByProperties 类有 2 个属性, RelatedObjects 赋值为 IfcBuilding, RelatingPropertyDefinition 赋值为 IfcPropertySet, IfcPropertySet 的 HasProperties 属性赋值为 IfcProperty, IfcProperty 有 3 个属性, 分别为 name、

type 和 match, 这样最终的 3 个属性, 即 name、type 和 match 就通过 IfcRelDefinesByProperties 附加到 IfcBuilding 上.

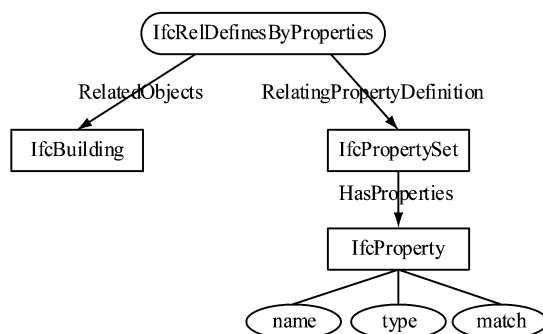


图 10 节点基本信息的工业基础类表示

Fig. 10 Basic information of the joint expressed in industry foundation class

### 2.2 实体的存储

基本实体包括杆件、板件、螺栓、焊缝等. 如图 11 所示, 空心箭头表示派生关系, 所有的实体均从 IfcProduct 派生, IfcRelContainedInSpatialStructure 表示 IfcProduct 隶属于 IfcBuilding. 在系统中, 梁用 IfcBeam 表示, 柱用 IfcColumn 表示, 板件用 IfcPlate 表示, 螺栓用 IfcMechanicalFastener 表示, IFC 中没有直接与焊缝对应的类, 在本系统中焊缝由 IfcBuildingElementProxy 表示, 由 IfcPolyLine 定义其显示方式, 通过 IfcProperty 定义附加属性, 比如焊缝高度、焊缝类型等<sup>[13]</sup>.

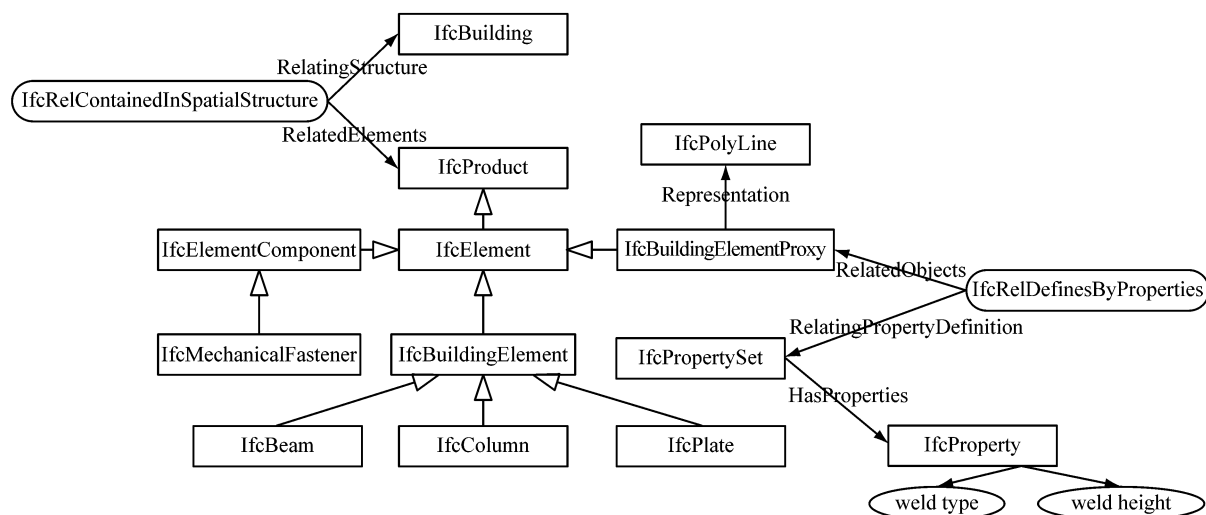


图 11 基本实体的工业基础类表示

Fig. 11 Basic entity expressed in industry foundation class

### 2.3 操作的表示

在 IFC 标准中并没有关于参数化的描述部分, 但是如果将节点拼装看作施工过程的话, 那么操作

可以转变为 4D 施工信息, 从而实现操作的 IFC 表示.

在 IFC 文件中, 4D 施工信息有几个关键性的类

别:工作计划(IfcWorkPlan)、工作安排(IfcWorkSchedule)、工作任务(IfcTask)<sup>[14-15]</sup>。

在一个典型4D施工信息中,一个工作计划(IfcWorkPlan)包含多个工作安排(IfcWorkSchedule),一个工作安排(IfcWorkSchedule)包含多个工作任务(IfcTask),一个工作任务(IfcTask)完成多个实体(IfcElement),

典型模型的IFC构架表示如图12所示。在图中,IfcRelAggregates表示集合关系,IfcRelAssignsToControl表示IfcTask隶属于IfcWorkSchedule,IfcRelAssignsToProcess表示IfcElement隶属于IfcTask,IfcRelSequence指定IfcTask的先后关系,IfcRelDefinesByProperties将属性附加到IfcTask上。

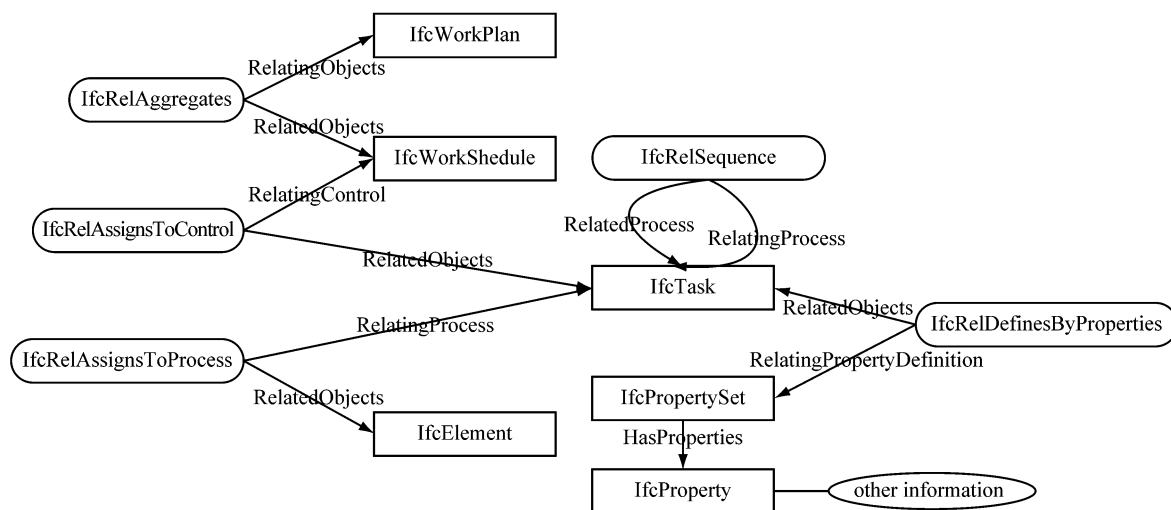


图12 4D施工信息的工业基础类构架

Fig.12 Framework of the 4D construction information expressed in industry foundation class

在参数化节点向4D施工信息转换时,一个参数化节点为一个IfcWorkPlan,由于参数化节点只有一个工作流程,因此一个IfcWorkPlan只包含一个IfcWorkSchedule。一个操作对应一个IfcTask,一个实体对应一个IfcElement。具体对应关系如图13所示,图中箭头上的1:n表示一对多的映射关系,比如一个参数化节点包含多个操作。

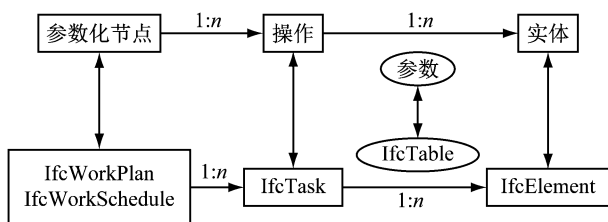


图13 参数化节点与工业基础类的对应关系

Fig.13 Relation between parametric joint and industry foundation class

## 2.4 变量列表的表示

操作在执行时需要变量,每一步操作由变量控制。变量列表的本质是建立一个数据表格,在IFC中,通过IfcTable来存储数据表格。变量列表的表示如图14所示。

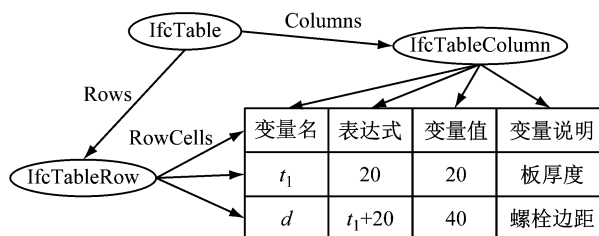


图14 变量列表的工业基础类构架表示

Fig.14 Variable list expressed in industry foundation class

## 3 实例展示

### 3.1 参数化节点实例

本系统隶属于上海同磊土木工程技术有限公司开发的3D3S钢结构软件,已经成功进行商业应用。此系统改变了3D3S软件无法添加节点的现状,同时带来了传统参数化节点没有的优势。

本系统基于AutoCAD ObjectArx环境开发<sup>[16]</sup>,采用IfcOpenShell作为IFC文件的生成工具。IfcOpenShell是一个对IFC文件进行解析的开源的工具包,具体介绍可以参见其官网<http://www.ifcopenshell.org/>。

系统中已经通过参数化建模成功制作了大量的节点. 图 15 是部分节点展示, 目前系统中提供了辅助点、辅助线、添加板件、螺栓、锚栓、栓钉、切割、镜像等参数化操作, 可以满足绝大多数节点的参数化制作.

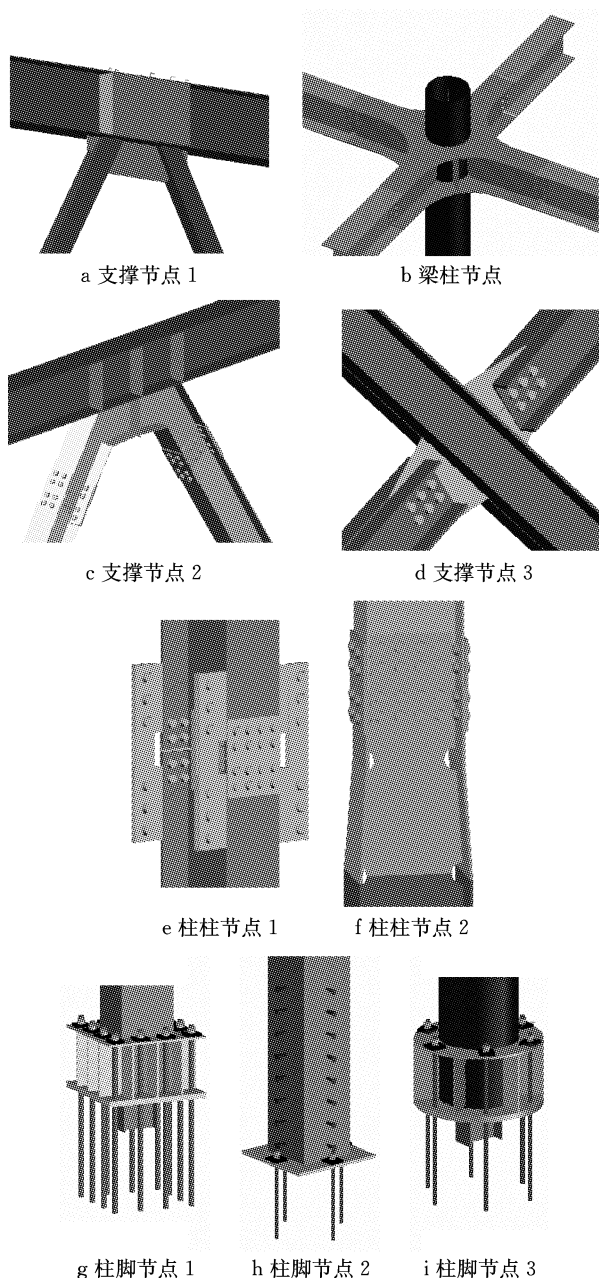


图 15 部分参数化节点展示

Fig.15 Part of the parametric joints

节点制作完成之后, 每一个参数化节点保存为一个 IFC 文件, 图 16 是已经制作完成的参数化节点文件列表. 将节点文件放在软件的指定目录下, 在软件启动时便能自动加载节点. 图 17 是部分已经加载的节点, 在图 17 的对话框中, 选择需要设计的节点, 弹出图 18 所示的节点设计界面, 输入合适的参数, 点击确定, 完成节点的设计.

- H Beam-H Column-Flange (Bolt and Weld).ifc
- Box Column-Box Support.ifc
- H Beam-Box Column-Box Support.ifc
- H Beam-Box Support.ifc
- H Beam-Box Support-Box Support (1).ifc
- H Beam-Box Support-Box Support.ifc
- H Beam-Double Angle Support-Double Angle Support.ifc
- H Beam-Double Channel Support-Double Channel Support.ifc
- H Beam-H Beam-All Weld.ifc
- H Beam-H Beam-Box Support (Horizontal).ifc
- H Beam-H Beam-Flange Double Plate-Web Double Plate.ifc
- H Beam-H Beam-Flange Single Plate-Web Double Plate.ifc
- H Beam-H Beam-Flange Weld-Web Double Plate.ifc
- H Beam-H Beam-Flange Weld-Web Single Plate.ifc
- H Beam-H Column-Flange (Angle Weld).ifc
- H Beam-H Column-Flange (Butt Weld).ifc
- H Beam-H Column-H Support (Bending Plate).ifc
- H Beam-H Support-H Support (Bar).ifc
- H Beam-H Support-H Support (Plate).ifc
- H Column-Box Support.ifc
- H Support-H Support-Cross (1).ifc
- H Support-H Support-Cross (2).ifc

图 16 制作的部分节点文件界面

Fig.16 Interface of the files of part of the joints made

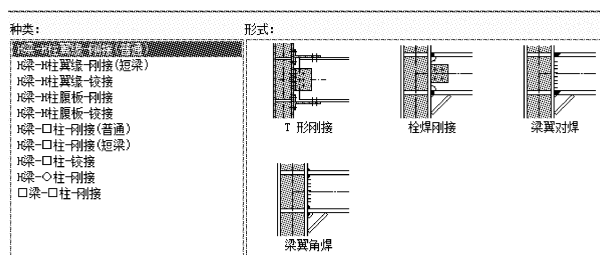


图 17 部分已经加载的节点界面

Fig.17 Interface of part of the joints loaded

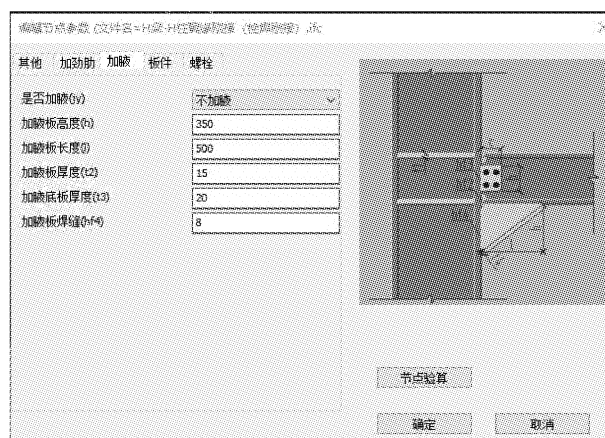


图 18 节点设计界面

Fig.18 Interface of joint design

### 3.2 系统优势验证

目前 Tekla 软件的自定义节点是基于约束的参数化节点的代表, 与 Tekla 软件的自定义节点相比,



基于操作的参数化节点具有以下优势:

(1) 建模方式容易被土木行业的用户所接受. 在 Tekla 软件中, 自定义节点的建模方式与传统的 CAD 软件建模方式不同, 通过添加约束来形成一个完整的模型, 这种方式并不被土木行业用户熟悉, 学习成本大. 据调查, 绝大多数 Tekla 软件用户不知道如何使用此功能. 而基于操作的参数化建模与普通的 CAD 建模方式一致, 经过测试, 土木行业的用户无需学习便能够上手使用基于操作的参数化节点来制作节点.

(2) 建模效率显著提升. 经实际测试, 挑选了 10 个没有参数化建模经验的用户来使用本系统, 这 10 个用户均能很快使用本系统制作出一个新的参数化节点. 最快的用户能够在 10 min 之内制作出一个梁柱节点. 而相对应的 Tekla 软件中, 用户首先要把节点中的零件放置到正确位置, 然后

在三维模型中增加约束, 由于在三维模型中增加约束情况比较复杂, 据实际调查, 一个梁柱节点平均需要 1h 的时间. 这个测试说明本系统适合土木从业者使用和制作节点, 参数化建模效率高.

(3) 实现了参数化节点的数据共享. 在 Tekla 软件中, 参数化节点文件仅能通过 Tekla 软件打开. 在本系统中, 通过操作向 4D 施工的转换实现了参数化信息向 IFC 标准的转换, 其他软件能够直接打开参数化节点文件, 可以查看节点的构造, 同时可以用施工模拟的方式查看节点的拼装过程. 第三方软件也可以直接编辑修改本系统的参数化节点的数据, 为软件直接交互共享参数信息提供了可能. 图 19 是本系统的参数化节点在 ifcwebserver 中打开的情况, 不仅可以查看节点的构造, 也可以查看节点的拼装过程.

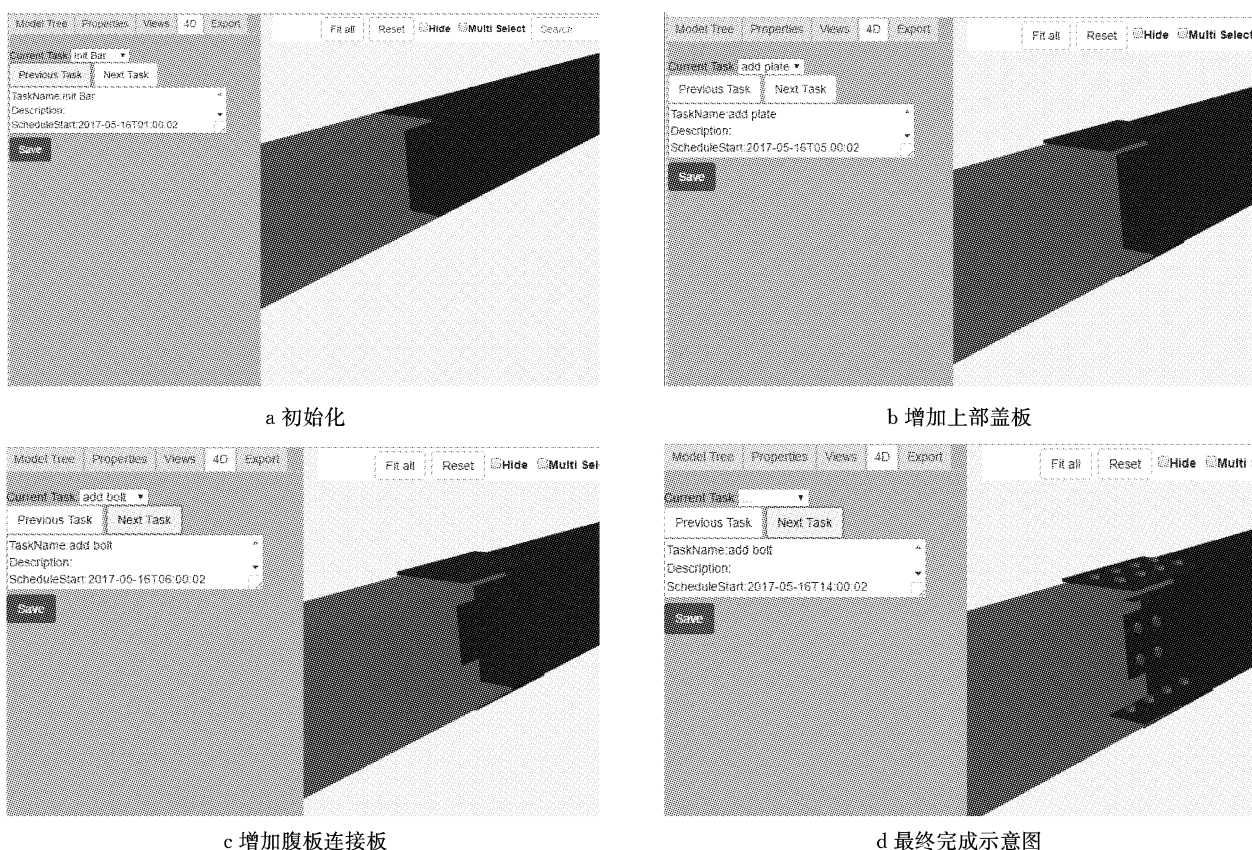


图 19 拼装过程在 ifcwebserver 中的显示

Fig.19 Design process displayed in ifcwebserver

(4) 可加入逻辑信息. 在 Tekla 软件中, 无论节点是否相似, 一个节点便是一个文件, 有多少个节点便需要建模多少次. 在基于操作的参数化节点中, 由于操作中可以增加逻辑信息, 这样可以将多个相似的节点融合在一个参数化节点之中, 减少了相似节点的重复制作工作量. 以梁柱节点为例, 梁柱节点分

为是否有加腋板以及与柱是单板连接还是双板连接, 因此 Tekla 软件中需制作 4 个节点: 加腋-单板连接、加腋-双板连接、不加腋-单板连接、不加腋-双板连接. 当有了逻辑信息之后, 此 4 个节点可以融合在一个节点中, 图 20 是节点设计时的界面, 通过下拉列表选择是否加腋以及是单剪还是双剪.



a 选择是否加腋

b 选择是否单剪

图 20 带逻辑信息的参数化节点示例界面

Fig. 20 Interface of the parametric joint with logical information

## 4 结论

从操作记录出发,从全新的角度设计了一套参数化节点制作系统,主要结论和成果如下:

(1) 实现了通过用户界面为软件添加钢结构节点的功能. 此套系统已经商用在 3D3S 软件中,用户反馈良好.

(2) 通过基于操作记录的方式建立参数化模型,此种建模方式与普通三维建模的方式基本一致,降低了用户的学习成本,更适合土木行业的从业者使用,同时不用考虑繁琐的约束添加,建模效率得到显著提升.

(3) 通过将操作转换为 4D 施工信息,实现了参数化节点的 IFC 文件存储,IFC 文件的可读性更强,读写效率高,同时也实现了参数化节点的信息共享.

(4) 基于操作记录的参数化建模可以增加逻辑信息,通过逻辑判断能够将多个相似的参数化节点融合于一个参数化节点之中,对于相似的节点无需重复建模,大大节省了节点制作的工作量.

## 参考文献:

- [1] ELLIOTT K S, JOLLY C K. Design of connections and joints [M]. Oxford: Blackwell Science, 2013.
- [2] Trimble Solutions Corporation. Tekla official website [EB/OL]. [2018-06-21]. <https://www.tekla.com/>.
- [3] 钱磊. Revit 族的创建技巧——以圆形风管变径弯头为例[J]. 暖通空调, 2017, 47(11): 83.  
QIAN Lei. Creating skills of Revit families: Taking reducing elbow of circular air duct for example[J]. Heating Ventilating & Air Conditioning, 2017, 47(11): 83.
- [4] WING Eric. Autodesk revit architecture 2012: No experience required[M]. New Jersey: John Wiley & Sons, 2011.
- [5] PLANCHARD David. Solidworks 2017 reference guide[M]. Mission: SDC Publications, 2017.
- [6] LIU Gang. Parameterized mockup design of PDC bits based on Pro/E [J]. Procedia Earth and Planetary Science, 2011 (3): 435.
- [7] 王乐文. UG 的参数化建模方法[J]. 中小企业管理与科技, 2016(4): 207.  
WANG Lewen. Parametric modeling based on UG [J]. Management & Technology of SME, 2016(4): 207.
- [8] LV Yanjie, ZHAO Gang. UG-based research and development of 3D pipe layout system of the aircraft engine[J]. Procedia Engineering, 2011(17): 660.
- [9] 孙明玉. 几何约束求解的关键技术研究[D]. 长春: 吉林大学, 2016.  
SUN Mingyu. Research on the key techniques of geometric constraint solving[D]. Changchun: Jilin University, 2016.
- [10] 王帅. 支持无历史建模的约束驱动技术研究[D]. 武汉: 华中科技大学, 2013.  
WANG Shuai. Research of constraint-driven technology for history-independent shape modeling [D]. Wuhan: Huazhong University of Science and Technology, 2013.
- [11] 陈振明, 张耀林, 黄冬平. Tekla Structure 软件在 CCTV 主楼钢结构深化设计中的应用[J]. 施工技术, 2008(8): 73.  
CHEN Zhenming, ZHANG Yaolin, HUANG Dongping. Application of Tekla Structure software on steel structure detail design in CCTV main building[J]. Construction Technology, 2008(8): 73.
- [12] OWOLABI A, ANUMBA C J, EL-HAMALAWI A, et al. Development of an industry foundation classes assembly viewer [J]. Journal of Computing in Civil Engineering, 2006, 20(2): 121.
- [13] Building SMART International, Ltd. IFC official website [EB/OL]. [2018-06-21]. <http://www.buildingsmart-tech.org/specifications>.
- [14] KASSEM M, DAWOOD N, CHAVADA R. Construction workspace management within an industry foundation class-compliant 4D tool [J]. Automation in Construction, 2015 (52): 42.
- [15] SEO J, KIM I. Industry foundation classes-based approach for managing and using the design model and planning information in the architectural design[J]. Journal of Asian Architecture and Building Engineering, 2009, 8(2): 431.
- [16] HOU X Y, LU X L. Automatic design of transmission lines using ObjectARX technology[M]. New York: IEEE, 2012.