

多高层钢框架前后处理程序的研制与开发*

常时峰, 杜新喜, 张 慎

(武汉大学 土木建筑工程学院, 湖北武汉 430072)

摘要:多高层钢框架在进行有限元计算时,必须首先输入大量的信息数据,包括节点信息、单元信息、荷载信息等。因此开发出具有三维交互功能的多高层钢结构计算机辅助设计软件成为一种必要。采用面向对象的系统开发技术,完成了基于 ObjectARX 的多高层钢框架前后处理系统,扼要介绍了系统实现的关键技术,包括结构类的设计、可视化信息的选取、楼板的识别功能等,并给出了相应的代码。

关键词:多高层钢框架; 前后处理; ObjectARX

中图分类号: TU311.41 **文献标识码:** A **文章编号:** 1006-7329(2007)01-0056-05

Study on Pre - & Post - Processing Program for Multi - storey & Tall Steel Frame

CHANG Shi - feng, DU Xin - xi, ZHANG Shen

(School of Civil and Architectural Engineering, Wuhan University, Wuhan 430072, China)

Abstract: A vast amount of information data must be input before finite element analysis of steel frames, such as node information, element information and load information. It is necessary to develop a computer aided design system with 3D interactive function. A CAD system for steel frame based on object ARX is developed with AutoCAD API and the Object - oriented technique. In this paper, some key programming technologies are studied, including design of structures, identification of floor system, also, it has presented the relevant implementing code.

Keywords: multistorey & tall steel frame, pre & post processing, Object ARX

伴随着工程设计和计算机科学日益紧密的联系,计算机辅助设计(CAD)已成为一门新兴的研究领域。虽然国内几家科研院所对多高层钢结构 CAD 软件系统做了一些研究,并推出了各自的软件,但人机界面友好、图形编辑功能强大的多高层钢结构 CAD 软件并不多见。因此,对开发一个智能化与专业化的多高层钢结构 CAD 软件做一些探讨是非常有益的。

一个优秀的工程结构 CAD 软件必须有一个良好的图形支撑系统,而以现有的 CAD 平台作为支撑的软件开发模式,其优势是非常明显的。有鉴于此,本文采用 AutoCAD 2000 提供的二次开发系统 ObjectARX 及 Visual C++ 6.0,借助于 AutoCAD 2000 优秀的图形处理平台实现了一种集可视化模型设计、分析计算(本文略)和图纸输出于一体的多高层钢框架 CAD 系统。摒弃了许多泛泛而谈的软件功能的介绍,文中详细介

绍了系统实现的关键技术,对今后结构辅助计算程序的开发具有一定的参考价值。

1 系统设计

本软件系统由七个 AutoCAD 运行时扩展的动态库工程构成。功能介绍如下:

1) Criteria(DLL)。用于存放规范规定的构件验算的相关条文和截面及材料类。

2) ModeBase(DBX)。用于定义通用有限元分析所必须的类(如节点类、单元类、结构总体信息类等),类中通过继承和多态的机制封装了它们各自的属性和相应的操作。

3) SFDBX(DBX)。用于存放通过继承通用有限元类(ModeBase)生成的多高层框架特有的类(如板类),主要用于封装框架特有的属性和相应的操作。

* 收稿日期: 2006-11-05

基金项目: 建设部钢结构软件开发项目(03-5-019)

作者简介: 常时峰(1980-),男,江苏扬州人,博士,主要从事钢结构、组合结构、结构 CAD 等方面的研究。

4) AssitFun (ARX)。用于存放通用的全局辅助函数。

5) ComDlg (ARX)。用于存放通用对话框,便于程序的扩展。

6) SFU I(ARX)。用于前后处理中与用户进行交互,包括命令的定义、对话框的调用。

7) CaIProc (ARX)。用于结构内力的计算分析。

由于该软件系统的功能复杂、组织庞大,若采用传统的结构化程序的分析设计方法,数据和处理这些数据的操作间的分离将带来非常大的负面影响,程序在很多方面将出现重复投入,而采用面向对象的程序设计方法,不仅提高了软件水平和开发效率,也为以后的扩展提供了方便。

2 结构模型的抽象

整体结构由节点和单元组成,包含节点和单元的信息。可以将整个结构抽象为一个类,整体结构类反映模型的拓扑信息、约束信息、荷载情况等,封装了对整体结构数据的操作,如总体刚度矩阵的形成与分解、荷载列向量的形成等。

在进行有限元计算时,可以将结构的构件抽象为单元类。由于单元是一种可抽象为图形表示的实体,故从 AcDbEntity 派生出单元类。虽然单元形状和特性各异,但基本特征和功能相同,比如都有一定的几何形状,通过节点与整个结构相连,共同参与整个结构受力,在结构分析中均以刚度形式组合进整体刚度矩阵等。

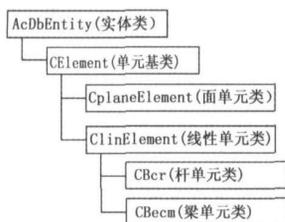


图 1 单元间的继承关系

根据单元的这些特性,基于“分工明确、各司其责”的原则,建立了如图 1 所示的继承关系。单元类包含单元的基本属性,如材料属性等;线性单元类则专门针对杆件设计,它包含了节点、截面等属性(通过将节点、截面的 ID 作为单元类的硬指针成员来实现);由于在形成单元刚度等方面的差异,又从线性单元类分别派生出梁单元类和杆单元类。图 2 概要的给出了系统设计的结构对象类的基本构成及其相互之间的关系。

值得注意的是,基类的设计不仅包含了单元的属性,还封装了相应的操作。比如节点类的设计既包含

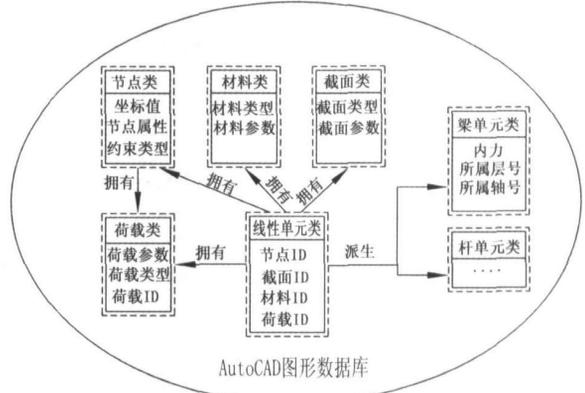


图 2 基类构成及其关联图

了节点坐标、自由度信息、节点荷载等属性,同时也封装了对节点的基本操作,如节点坐标的设置、节点荷载的绘制、差异化的图形显示等。区别于节点、单元这样的实体对象,对于抽象的工程数据,比如单元的截面、材料等。系统定义了派生自 AcDbObject 的类,用这些类实例化的对象将直接加入到相应的有名对象字典中。程序中我们在有名对象字典下分别建立了截面 (WD_SEC)、材料 (WD_MAT) 子字典,在对这些子字典进行反复的添加、删除操作时如何保持截面或材料对象编号的连续性以及如何处理拥有这些截面或材料的单元,以材料字典为例,我们编写了下列两个函数: wdAFAddMat() 和 wdAFDeMat(), 函数处理步骤如下:在删除某个材料时,首先遍历 CAD 模型空间,找出拥有这个对象的单元,将其材料 ID 置为空,紧接着以将要删除的材料对象为起点,将其后的材料对象的编号依次减一。

3 关键功能的实现技术

3.1 轴线功能的实现

为了方便结构构件的空间定位,建模过程中需输入结构平面的轴网。轴网包括矩形轴网和弧形轴网。基于信息保存以及在 CAD 中绘图的需要,将轴线的属性用一个类 (CAxis) 来描述。轴线类派生自 AcDbEntity,包含了轴号、轴线类型等基本信息,封装了轴线命名、轴号修改等一系列操作。系统还重载了相关的函数,比如: GetSnapPoints()、IntersectWith(), 实现了轴线的端点、中点以及轴线与其他 CAD 实体交点的捕捉功能,方便了构件布置时的定位。除了参数化建立标准轴网以外,系统还提供了将 CAD 的直线和弧线转化为轴线的功能,同时轴线的归并功能将使直线式方程相同的轴线自动合并,避免了模型中重复轴线的出现。下面的代码展示了程序如何实现轴线与直线的交点捕捉:

```
wdCSFAxis: IntSecWithLine (wdCAxis * eInt, AcGe-
```

```

LineSeg3d * lnsgArg,
    AcDb:: Intersect intType, AcGePlane * projPlane,
    AcGePoint3dArray &points)
{
    AcGeLineSeg3d lnsgThis; AcGeCircArc3d arcThis;
    AcGeTol tolerance; tolerance. setEqualPoint (0.
01); //设置误差为 0.01 mm
    switch (emt - > GetAxisType ())
    {
    case AT_L NE: //直线型轴线
        eml - > startPt (pt1); eml - > endPt (pt2);
        lnsgThis. set (pt1, pt2);
        lnThis. set (pt1, pt2);
        switch (intType) //相交模式
        {
            case AcDb: : kOnBothOperands: //两者均不
延长
                if (lnsgThis. intersectWith (* lnsgArg, pt1,
tolerance) == Adesk: : kTrue) points. append (pt1);
                break;
            case AcDb: : kExtendBoth: //两者均延长
            .....
            case AcDb: : kExtendThis: //延长轴线本身
            .....
            case ..... //其他相交模式
        }
        break;
    case AT_ ARC: //弧线型轴线
        .....
        break;
    }
}

```

3.2 图形数据的操作以及系统界面的组织

区别于以往将图形与结构信息分开存储的程序设计方法^[3],系统采用面向对象的程序设计方法,结合多高层框架结构设计的需要,定义了自己的结构类。由于这些类都直接或间接的从 AcDbObject 派生,因此对模型进行选取操作时可以直接利用基类的函数,如 desc()、isA()和 isKindOf()来判断对象的类型(节点、柱、梁等),避免了烦琐的图形与数据的对比操作,提高了程序执行效率。对于实体类型的单元类和节点类,系统设计了功能齐全的显示模式,以梁单元为例,对于单元本身,显示模式可以分为直线、实体或者隐藏,对于附加信息的显示,显示模式又可以分为显示单元号、材料号或截面号等。下面的代码展示了梁单元

的实体显示功能:

```

wdCBeam: : worldDraw (AcGWorldDraw * mode)
{
    AcGiFaceData faceData; //面信息
    AcGeVector3d m_dirc [3]; //局部坐标轴在整体
坐标下的单位方向向量
    AcGePoint3dArray pvAr; //端截面关键点相对坐
标数组
    AcGePoint3dArray pv [2]; //端截面关键点绝对
坐标数组
    AcGePoint3d showPos [2]; //单元两端截面形心
坐标
    switch (showmode) //showmode 单元显示模式
    {
        case SE_L NE: //绘制单线图
        case SE_ SOL D: //绘制三维实体图
            switch (SecTypeNo) //SecTypeNo: 单元截面类
型
            {
                //根据截面类型计算截面关键点绝对坐标并
绘制端截面
                case wdCSec: : ST_REC_USER: ..... //矩形
                case wdCSec: : ST_I_USER: ..... //工字形
(包括 H 形)
                    for (j=0; j<pvAr. logicalLength (); j++)
                    {
                        pv [0] [j] [0] = pvAr [j] [0] * m_dirc [0] [0] +
pvAr [j] [1] * m_dirc [1] [0] + pvAr [j] [2] * m_dirc
[2] [0] + showPos [0] [0];
                        pv [0] [j] [1] = pvAr [j] [0] * m_dirc [0] [1] +
pvAr [j] [1] * m_dirc [1] [1] + pvAr [j] [2] * m_dirc
[2] [1] + showPos [0] [1];
                        pv [0] [j] [2] = pvAr [j] [0] * m_dirc [0] [2] +
pvAr [j] [1] * m_dirc [1] [2] + pvAr [j] [2] * m_dirc
[2] [2] + showPos [0] [2];
                    }
                mode - > geometry (). polygon (pvAr. length (), pv
[0]. asArrayPtr ());
            }
            //确定两个端面之间各表面的各点坐标并画实体
表面 (示意)
            mode - > geometry (). mesh (2, 2, pFace, NULL,
&faceData);
    }
}

```

3.3 楼面荷载的自动导算

系统中,楼板被定义为一个由主梁或次梁围成的闭合多变形。房间识别功能是建模中楼面系统的核心函数,所识别出来的房间是楼面荷载布置的基本单元。具体来讲,即搜寻出所有的主梁及次梁围成的封闭区域,并将围成区域的节点与杆件按一定顺序存放,函数处理步骤如下:

1) 判断并排除无法构成封闭区域的“孤立”杆件。“平面内”的杆件中,与某节点所相连的杆件数目若小于 2,该节点为孤立节点,不应参加区域搜寻。

2) 形成封闭区域的网格线。遍历 AutoCAD 的模型空间,找到所有梁单元,添加到 D 数组,对数组中的元素进行循环找到它们两两的交点,并将这些交点放到点数组 (AcGePoint3dArray) 中,形成网点数组。再次遍历梁单元数组,找到每根梁上所有的网点 (包括梁的起点和终点) 并按照一定的顺序进行排序,将网点依次相连形成网格线。

3) 根据网格线寻找封闭区域。对得到的网格线数组进行循环,将网格线的起点作为板的起点,找到所有经过该点的网格线。通过找到的网格线 12 (见图 3),以 2 为圆点,12 边 (方位角为 $\pi/2$) 为起始边,逆时针旋转,得到角 123,再以 23 边为起始边,逆时针旋转得到角 231 和角 234,比较角的大小,小者即为下个网格线的一个顶点,如此便得到网格线段 31,从而得到一个封闭区域 123,依次进入下一次循环。对于已经参与循环的网格线,如果使用的方位角与上次相同,程序将会自动将其剔除,不再参与循环。提高了程序执行效率。按此规则进行遍历,即可得到所有的封闭区域,包括大板区域。

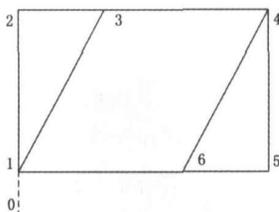


图 3 房间识别示意图

4) 设置封闭区域所包含的杆件数组和节点数组,也即形成板单元。

3.4 风荷载的导算

风荷载导算步骤如下:

1) 以各个楼层作为控制点求出该楼层处的风压值,楼层间的风压值近似为线性变化。

2) 将风压分配到各个楼层,形成各个楼层所受的线压值。3) 将各个楼层所受的线压值平均分配到处于迎风面和背风面的节点上。楼层的周边节点可以通

过在房间识别过程中形成的大板 (参考图 3 中的区域 123456) 获取。

楼层节点上的风荷载的计算方法如下:

整个楼层总的风荷载所形成的集中力为:

$$P_{wx} = (w_1 + w_2) \times Ly \times h$$

$$P_{wy} = (w_1 + w_2) \times Lx \times h$$

式中: w_1, w_2 分别为楼层的迎风面与背风面的标准风压值; Lx, Ly 分别为楼层沿 x 和 y 方向投影长度; h 为该楼层的计算高度。

楼层上每个节点上分配的风荷载为:

$$P_{xi} = \frac{P_{wx}}{2Lx} \times \sum_{i=1}^2 x_i/2 = \frac{(w_1 + w_2) \times h}{4} \times \sum_{i=1}^2 x_i$$

$$P_{yi} = \frac{P_{wy}}{2Ly} \times \sum_{i=1}^2 y_i/2 = \frac{(w_1 + w_2) \times h}{4} \times \sum_{i=1}^2 y_i$$

式中: P_{xi}, P_{yi} 分别为楼层 i 节点沿 x 和 y 方向的风荷载标准值; x_i, y_i 分别为与 i 节点与相邻的两节点的 x 和 y 坐标的差值。

4.5 单元内力及变形的图形显示功能

在单元应力及变形的后处理显示功能的开发过程中,参考通用有限元软件的做法,系统不仅提供了以云图或等值线 (图 4) 的形式表述的单元内力,还以查询对话框的形式直接给出了单元内力的具体数值 (图 5),方便用户检查或做进一步的验算。基于单元内力数据的封装要求,系统将内力的回显功能设计成单元的成员函数,函数名为 DrawInterForceGraph ()。在需要显示内力的时候,该函数读取实时更新的内力数组,包括沿单元长度各等分点的内力,单元端截面的内力,根据用户设定的信息 (工况组合、是否显示内力数值以及文字的高度等) 在单元的局部坐标系下进行图形的绘制。为了实现系统的模块化和可移植性,针对不同的内力 (轴力、剪力和弯矩等) 系统设计了不同的子函数。对于结构类型复杂,单元数目较多的情况,为了直观清晰的观察单元的内力情况,用户可以有条件的选出部分杆件进行观察,系统提供的筛选类别有楼层、轴号以及用户界面的直接选取,在这里,系统直接调用基类 AcDbEntity 的虚函数 SetVisibility 进行单元的隐藏或显示。

下面的代码概要的介绍了单元内力的图形显示功能的实现过程。

```

wdCBeam:: DrawInterForceGraph ( AcGWorldDraw *
mode)
{
    double txHt, scale; //文字高度和宽高比
    wdMBCurGlbData GetTextHeightAndScale ( txHt,
scale); //获得文字高度等总体信息
    
```

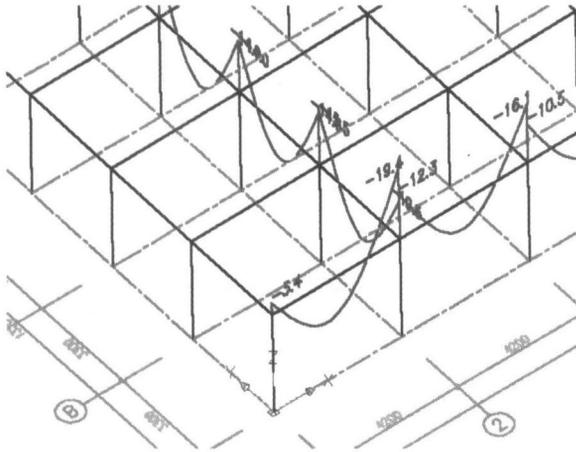


图 4 单元内力的图形显示(弯矩)



图 5 单元内力的查询

```

double m_interForc[8] //端部截面内力数组
double m_midForc[BEAM_D IV][6]; //单元等分
点内力数组, BEAM_D IV为单元等分数
switch (wdMBCurGlbData GetShow IntForcType ())
/获得当前需要显示的内力类型
{
case wdc: : FT_FX: //沿局部 1轴轴力 .....
case wdc: : FT_FZ: //沿局部 3轴剪力
{
const double stdFz = std IntForceVal[2]/3; //txHt
高度对应的剪力图线高
for(int i=0; i<(BEAM_D IV+2); i++)
{
/获得单元局部坐标系下的内力图形关键点

```

```

if(i==0)//单元始端
bcKp[i][2] = - txHt * (m_interForc[2]) /stdFz;
Fz;
else if(i==(BEAM_D IV+1))//单元终端
{
bcKp[i][0] = m_len;
bcKp[i][2] = txHt * (m_interForc[8]) /stdFz;
}
else //单元中间截面 {.....}
/坐标转换并进行应力图的绘制
wdMBCurGlbData GetGlbPosOfGlbPt ( bcKp [ i ], m _ showPos
[0],m_dir, tranVec, gbKp [ i ]);
mode - > geometry (). polyline (BEAM_D IV + 2,
gbKp);
}
} break;
}
}

```

4 结语

针对多高层钢框架设计的需求和特点,实现了一个人机界面友好,设计步骤较明确,功能较完善,计算精度较高(本文略)的多高层钢框架计算机辅助设计系统。系统充分利用 AutoCAD 的三维可视技术,提供了多种便捷的建模与图显功能,可以处理任意复杂的多高层建筑结构。系统设计过程中引入了面向对象的程序设计思想,利用其继承性、封装性、多态性的特点,为功能模块的集成化、可再用以及程序扩充的灵活性提供了有利条件。

参考文献:

- [1] 邵俊昌,李旭东. AutoCAD ObjectARX2000开发技术指南 [M] 北京:电子工业出版社,2000.
- [2] Stanley B. Lippman 深度探索 C++对象模型 [M] 武汉:华中科技大学出版社,2001.
- [3] 贺拥军. 高层与多层钢框架结构计算的前处理程序研制和开发 [J] 建筑科学,2004,(5):74-78.
- [4] 李国强. 多高层建筑钢结构结构设计(第一版) [M] 北京:中国建筑出版社,2004.