

基于 CATIA 的机床夹具元件快速造型方法

Rapid Prototyping of Machine Tool Fixture Element Based on CATIA

北京航空航天大学机械学院 李昕欣 郑国磊

[摘要] 介绍了基于 CATIA 的 CAA 架构对实体进行参数化建模的方法,以一个航空夹具设计为实例,开发了基于 CAA 及零件特征的参数化航空夹具设计模块,为基于 CATIA 的航空夹具元件快速造型做了技术方法的探索。

关键词: CATIA CAA 快速应用开发工具 二次开发

[ABSTRACT] The method to constitute parametric substantiality modeling based on CAA frame of CATIA is introduced. With taking aviation jig design as a case, the design module of parametric aviation jig is developed based on CAA and part features, and the technical method for rapid prototyping of aviation jig element based on CATIA is investigated.

Keywords: CATIA CAA RADE Secondary development

机床夹具在飞机制造工艺装备中占有较大比重,其研制也是进一步缩短产品生产准备周期的一个重要环节。为了适应现代飞机制造技术的发展和满足重点航空型号研制的现实要求,我校与沈阳飞机工业(集团)有限公司合作,提出基于 CATIA V5 系统的“机床夹具智能 CAD(JigCAD)系统”的研究和开发,“夹具元件的快速造型和建模方法”是本课题研究中拟解决的关键技术之一。

本文拟就基于 CATIA V5 系统的夹具专用零件的快速造型方法开展讨论:简述 CATIA V5 系统的专业化开发方法;着重叙述基于 CATIA V5 系统应用接口(CAA)的零件快速造型算法,并介绍该算法的实现形式及其应用结果,最后总结了夹具零件快速造型方法的特点。

1 基于 CATIA V5 系统平台的专业开发方法

1.1 开发方法

以 CATIA V5 系统为平台,进行专业化开发的常用方法有以下 2 种。

一是应用宏指令的方法。在零件的交互建模过程中,同时启用 CATIA 系统的宏代码录制功能。应用可视化程序开发工具——VBScript,对录制结果,包括对其中的各个接口对象及其方法和属性进行编辑,使之成为合乎 VB 语言规范、可重复进行调用和运行的应用程序。但是,这种方法存在应用过程繁琐、开发工作量大、开放性差等局限性,加载多个工程时还易造成内存不足等问题。因此,这种方法仅适合于一些结构和形状较为简单的实体和零件快速造型功能的开发。

二是基于 CAA 的方法。为了克服“宏指令录制”方式存在的局限性,CATIA V5 系统还同时提供了基于组件技术的应用接口(CAA)及其快速应用开发工具(RADE)。CAA 分为不同的应用框架。每个框架提供了相应的接口。相比宏指令,CAA 具有接口齐全、开发灵活和开放性良好等优点。应用不同框架和接口中提供的方法,可进行较为深入的专业功能开发。

1.2 基于 CAA 的专业开发

CAA 是 CATIA 在以前的 API 开发接口的基础上形成的一个开发函数库,使用 RADE 快速开发环境作为开发平台。其进行操作的数据组织结构与以往的 3D 造型二次开发有着根本的区别。通常的开发软件都采用高级开发与底层数据直接进行对话的方式,而 CAA 则利用其特有的“接口(interface)”功能将底层函数库同开发程序进行了很好的区分。开发者进行程序开发时调用 Interface 将操作传给接口,接口接受命令之后去调用底层函数并将指针传回供开发者使用。

CAA 开发具有其特有的开发逻辑:厂—对象—接口。一般的开发方法为:(1)提取“厂”;(2)利用“厂”中方法创建“对象”;(3)提取对象“接口”;(4)利用“接口”中的方法操作对象,如图 1 所示。

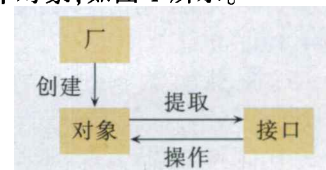


图 1 一般开发方法示意图

Fig.1 Diagram of general development method

2 夹具元件的快速造型

2.1 一般算法

进行零件快速造型的一般方法是在 PartDegin 模块下通过调用接口函数生成实体模型，其一般算法流程如图 2 所示。

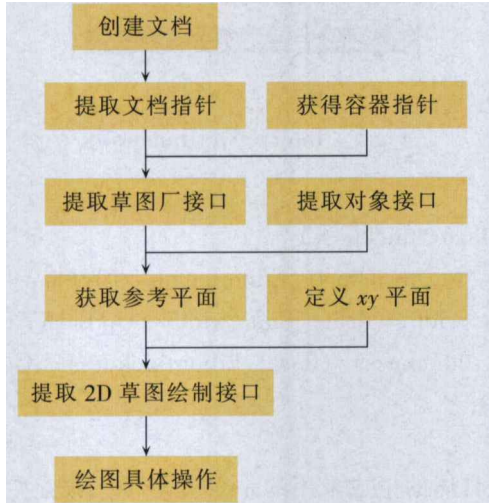


图 2 一般实体造型基本流程
Fig.2 Basic flowchart of general substantiality prototyping

(1)创建文档。方法如下：

```
CATDocument* pDoc = NULL;
rc = CATDocumentServices::New("Part", pDoc);
```

(2)提取文档指针并获得容器指针。如：

```
CATInit *piInitOnDoc = NULL;
rc = pDoc -> QueryInterface (IID_CATInit,
```

```
(void**) &piInitOnDoc);
```

(3)提取草图厂接口。通常在 Partdegin 模块下进行实体造型都是在草图上进行轮廓绘制之后再行其

他如拉伸等操作的。因此,提取的为草图厂接口,操作如下:

```
CATISketchFactory_var spSketchFactOnPrtCont
(piPrtCont);
```

(4)获取参考平面。定义结构如下:

```
CATIPrtPart_var spPart;
```

```
if(piPrtCont) spPart = piPrtCont->GetPart();
```

```
CATListValCATISpecObject_var spListRefPlanes =
spPart->GetReferencePlanes();
```

(5)定义 xy 平面。选取接口如下:

```
CATISpecObject_var spSketchPlane = spListRef-
Planes[1];
```

(6)提取 2D 草图绘制接口。首先利用草图接口创建草图,然后提取 2D 草图绘制接口:

```
CATISketch_var spSketch = spSketchFactOn-
PrtCont->CreateSketch(spSketchPlane);
```

```
CATI2DWFFactory_var spWF2DFactOnSketch
(spSketch);
```

(7)草图平面创建之后可以在上面进行草图绘制,如点、线等基本元素的绘制。然后调用方法进行具体实体生成操作,如拉伸等。

在实际造型中所遇到的零件通常都不是简单的结构,而会比较复杂。无论是零件还是复杂组件,最终均可分解为各种体素,并且可以提取各个体素之间的相互联结关系。可直接定义造型的体素目前已经做出 12 种,包括立方体、圆柱体、圆台、多棱柱、耳片、键体、圆环、球、槽钢、角钢、圆管和方管。目前用于体素间联结关系的主要是布尔运算:交、差、合、并(见图 3)。得到一个组件,首先应确定可将其分成几个零件及各零件之间的联结关系,然后针对各个零件确定其中涉及的体素以及各体素之间的联结关系,可列出 CSG 树表达式使这种联结关系一目了然。

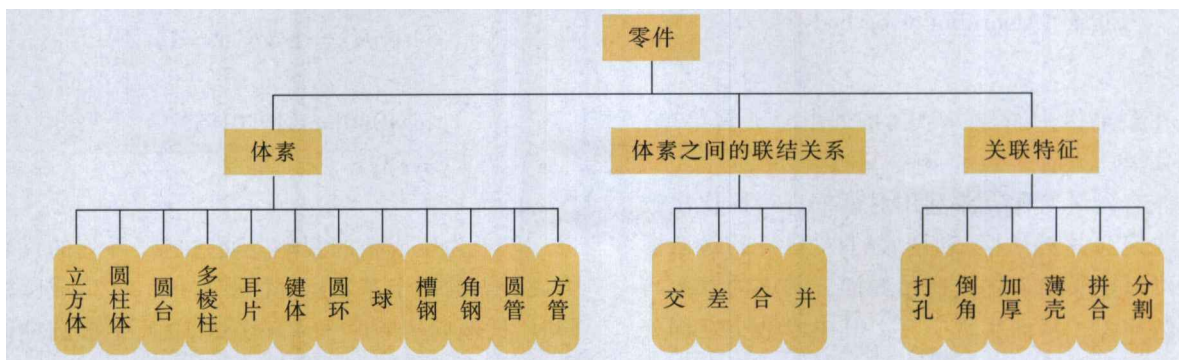


图 3 复杂零件的结构与联结模式
Fig.3 Structure and joining mode of complex part

2.2 夹具零件的快速造型

孔系列组合夹具结构中涉及到机床夹具中的基本元件,包括夹具体、支承件、定位件、夹紧件和导向件。其中夹具体是这个组合夹具的主体,其他元件与之相配合。下面以夹具体为例详细说明如何利用 CAA 进行夹具零件的快速造型。

(1) 夹具体元件的快速造型。

夹具体中的孔是标准的,共有 2 种。针对不同的材料可有不同的长宽尺寸,即当夹具体尺寸发生改变时,其上面的零件特征——孔的数目发生了改变,故将其长宽尺寸设置为参数,其余的辅助零件为标准件,有其自身的系列型号,此例中选择其中的一个规格作为示范。

依照参数体自身形状要求设置参数如下:夹具体的长 (length), 宽 (width), 厚 (thickness)。孔间距:以 K12 间距 40 为例。可以计算出各点起始坐标为: X 轴初始点: $t1 = \text{int}(\text{length}/52)$; $a = (\text{length} - 52 \times t1) / 2$ 。Y 轴初始点: $t2 = \text{int}(\text{width}/52)$; $b = (\text{width} - 52 \times t2)$ 。Z 轴厚度: thickness。

通过交互界面输入参数,生成夹具体。其算法如图 4 所示。



图 4 夹具体算法
Fig.4 Algorithm of jig body

首先在菜单栏中新建一个孔系列组合夹具菜单,设置下拉菜单。

夹具体有参量变化,故利用对话框进行操作比较方便。首先定义其对话框,利用 CATIA 自身的对话框定义功能可以定义出如图 5 所示界面,其中的参数与程序中参数相关,从而实现交互。其中参数交互部分为:

```
void LixxMmrTestDlg::GetSheet(float *length, float
```



图 5 对话框

Fig.5 Interactive framework

```
*width,float *thickness)
{
    *length=(float)_Editorlength->GetFloatValue();
    *width=(float)_Editorwidth->GetFloatValue();
    *thickness=(float)_Editorthickness->GetFloatValue();
}
}
```

夹具体的生成采用的是双重循环。随着夹具体长宽的变化,其上的孔数也是随之改变,故 2 个方向的变化都应考虑。

首先对其长度进行判断,在给定长度中应该有几个孔,两端的余量预知,故打孔的起始点为已知。根据同样的道理对宽度进行判断,由此循环的判断条件可确定。由于夹具体上的孔并不相同,而是每隔一定的距离由光孔变为沉头孔,并且按照一定的规律变化,所以亦应对此做出判断。判断过程为:

```
for(j=0;j<t2+1;j++)
{
    m=j%2;
    oxy[0][0]=x,oxy[0][1]=y;
    for(i=0;i<t1+1;i++)
    {
        k=i%2;
        if(k==0 && m==0) {... }
        else if(k==1 && m==1) {... }
        else {... }
        oxy[0][0]=oxy[0][0]+52;
    }
    y=y+52;
}
```

以上为生成孔的判断,通过这一系列的代码可以生成所需的用于布尔运算的圆柱体。在生成圆柱体的同时,把他们按顺序压入一个链表中,生成圆柱体代码为:

```
spSpecObj=LixxCreatePad(piPrtCont,pt,n,firstLim-
```

```
it,secondLimit,dirZ);
```

上述语句中引用的生成拉伸体函数代码为:

```
CATISpecObject_var LixxCreatePad(CATIPrtContainer
*piPrtCont,double pt[[2],int n,double firstlimit,double
secondlimit,CATMathDirection dir)
{ CATISketch_var spSketch=LixxCreateSketch(piPrt
Cont,pt,n);
//Retrieves the Part factory to create the pad
CATIPrtFactory_var spPrtFactOnPrtCont(piPrtCont);
piPrtCont->Release();
piPrtCont=NULL;
//Create the pad
CATISpecObject_var spSpecObj=spPrtFactOnPrt-
Cont->CreatePad(spSketch);
//Get the CATIPad interface
CATIPad_var spPadOnSpecObj=spSpecObj;
//Modify the parameters of the pad through the
CATIPad interface
spPadOnSpecObj->ModifyDirection(dir);
spPadOnSpecObj->ModifyEndType(catOffsetLimit);
spPadOnSpecObj->ModifyEndOffset(firstlimit);
spPadOnSpecObj->ModifyStartType(catOffsetLimit);
spPadOnSpecObj->ModifyStartOffset(secondlimit);
//Update the pad
spSpecObj->Update();
return spSpecObj;
}
```

在生成的底板中利用布尔运算依次减掉圆柱体,最终得到需要的夹具体,如图6所示。其中的布尔运算为:

```
spRemoveObj=LixxCreateRemove(piPrtCont,
spSpecObj,spObj);
```

(2)其他元件的快速造型。

限于篇幅,其他的元件不作详细说明,但是生成原

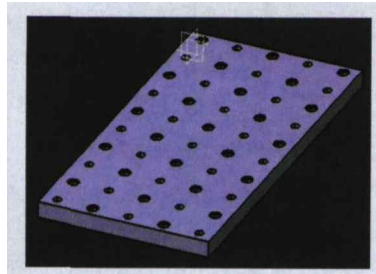


图6 夹具体

Fig.6 Jig body

理相似,也使用 PartDegin 模块,主要操作在草图绘制中进行。对实体进行整体体素划分,依次生成各部分体素,然后使用布尔运算进行加减运算得到最终结果,如图7所示。

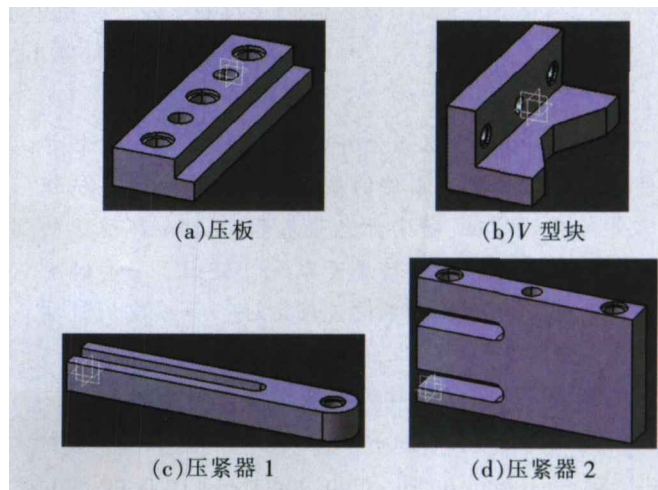


图7 其他元件

Fig.7 Other components

3 结束语

本文主要介绍了利用 CAA 进行 CATIA 航空夹具快速设计技术方法。首先对典型的航空夹具进行了特征分析,针对其特征运用 CAA 技术进行参数化特征建模,最终成功实现了建模的目的。通过利用这种技术可以将夹具设计中繁琐的零组件造型简单化,相似类型的零组件参数化,还可以快速实现其他方法难以简单实现的零组件建模。为快速装备设计提供了条件,为智能化夹具设计奠定了基础。目前 CAA 进行二次开发还属于起步阶段,很多问题尚在摸索中,但是由于 CATIA 自身的强大功能,可以预见其二次开发前景远大,还可以开展自动装配、自动布局、特征模型库管理等方面的研究。

参考文献

- [1] 何朝良,周安宁,刘毅. 基于 CAA 的 CATIA 二次开发的研究. 西宁:中国航空学会总体分会几何设计专业委员会第六次学术交流会议,2004.
- [2] 陈宁. 飞机发动机机匣零件虚拟制造系统[D]. 北京:北京航空航天大学,2004.
- [3] 董伊鑫. CATIA 二次开发研究及基于 CATIA 的五坐标数控机床运动仿真的研究与实现[D]. 北京:北京航空航天大学,2005.

(责编 咏智 七丁)

2008年第1期·航空制造技术 83