

三维数字化设计软件二次开发平台技术的应用

潘为民, 雷毅, 闫光荣

(北京航空航天大学机械工程及自动化学院, 北京 100083)

摘要: 二次开发平台是三维数字化设计软件的重要组成部分, 该文介绍了三维数字化设计软件二次开发平台技术的发展状况, 研究了国产三维数字化设计软件“CAXA 实体设计”二次开发平台的关键技术, 给出了一个在 CAXA 实体设计上所做的复杂二次开发实例: 基于加工过程的弧齿锥齿轮三维数字化造型。

关键词: 三维数字化设计; 二次开发平台; COM 技术; 弧齿锥齿轮

Application of Customized Development Platform of 3D Digital Design

PAN Weimin, LEI Yi, YAN Guangrong

(School of Mechanical Engineering & Automation, Beihang University, Beijing 100083)

【Abstract】 The importance and development situations of the customized development platform of 3D digital design software are discussed. The key techniques of CAXASolid API including objects diagrams, naming convention, callback functions, App wizard and extension data object are researched. A complex application of CAXASolid API is also given, which is the 3D modeling of the bevel spiral gear based on the manufacturing process.

【Key words】 3D digital design; Customized development platform; COM; Bevel spiral gear

1 概述

CAD 技术从诞生至今已有 30 多年的历史, 历经二维绘图、线框造型、表面造型、实体造型、特征造型等重要发展阶段, 其间还伴随着参数化、变量化、尺寸驱动等技术的融入。通过 30 多年的努力, CAD 技术在基础理论及软件产业方面日趋成熟, 推出了许多优秀的 CAD 系统, 在航空、航天、机械、模具、夹具、家电等制造业得到了广泛应用, 大大提高了制造业产品设计水平。

本身的特点和设计过程。不同的用户往往需要二次开发自己的专用系统, 这就要求 CAD 软件要具有功能强大、使用方便的二次开发平台^[1]。

基于组件技术的二次开发平台应用在以下几个方面:

(1) 用户化定制 CAD 环境: 提供用户化 CAD 规范; 提供用户化标准件库; 定制用户化 CAD 界面等。

(2) 开发用户的专用软件: 主要是指与行业密切相关的专用设计软件、CAPP 软件等。

(3) 开发与其它 CAD 软件的接口: 目前商品化的软件很多, 使用这些软件所建的 3D 模型和 2D 图纸都使用各自的数据结构, 有时需要把它们的数据进行相互交换, 就有必要开发它们之间的接口。

2 “CAXA 实体设计”二次开发平台的关键技术

CAXA 实体设计是北京北航海尔软件有限公司出品的国产三维数字化设计软件, 解决了三维数字化设计软件二次开发平台的多项关键技术^[2], 具有一定的代表性。IC API 是 CAXA 实体设计软件的二次开发平台, 作为 CAXA 实体设计与外部应用程序之间的接口, IC API 是一系列函数的集合。开发者可以通过高级语言编程来调用 IC API 函数实现如下功能:

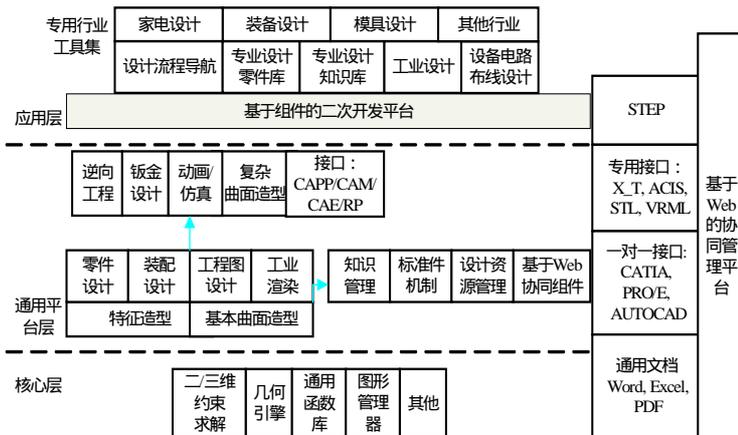


图 1 三维数字化设计系统的体系结构

三维数字化设计系统的体系结构 (如图 1 所示) 有 3 个层次: 核心层, 通用平台层和应用层。目前, 三维数字化技术已成为企业设计运用的热点。但是, 长期以来出于商业上的考虑, CAD 软件过分强调系统的通用性, 在核心层、通用平台层方面的研究和开发做了大量工作, 而忽略了设计对象

基金项目: 国家“863”计划基金资助项目“面向行业典型产品的三维数字化设计系统框架及通用平台”(2002AA411010)

作者简介: 潘为民(1968—), 男, 博士生, 主研方向: 计算机图形学, CAD/CAM, CMM 测量; 雷毅, 教授、博士; 闫光荣, 副教授、博士

收稿日期: 2005-11-14 **E-mail:** wmpool1@163.com

- (1)对 CAXA 实体设计模型文件及相应模型进行操作,包括建立模型、查询模型对象、遍历装配体、创建工程图等。
- (2)在 CAXA 实体设计主界面中创建交互式程序界面。
- (3)创建并管理用户定义对象等。

2.1 CAXA 实体设计二次开发平台的对象库

CAXA 实体设计二次开发接口的外部对象是标准的 COM 对象。每一个对象都支持一个或者多个接口。每一个接口都为 C++ COM 和 VB Automation(IDispatch)提供双重支持。VB Automation 能够自动匹配数据类型,如 VARIANT、BSTR 等。C++ COM 机制则允许直接使用基本的对象或数组,更适合于较为复杂的应用。

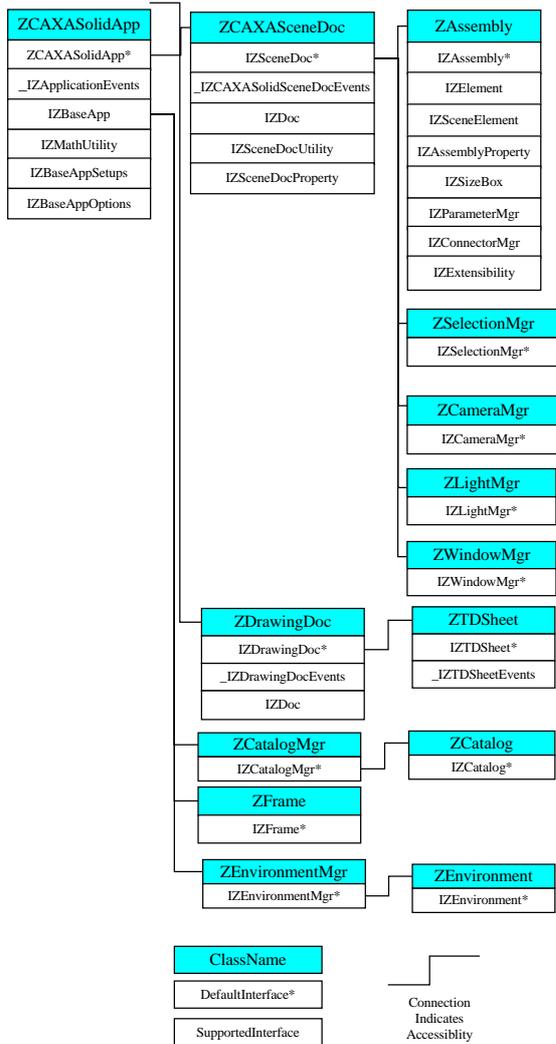


图 2 CAXA 实体设计二次开发平台的对象库框架 1

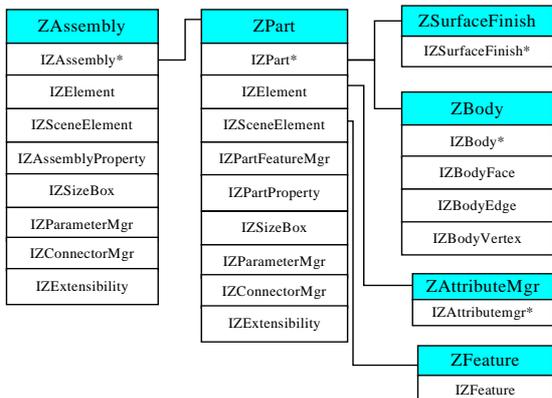


图 3 CAXA 实体设计二次开发平台的对象库框架 2

二次开发平台为用户提供了丰富的对象库,通过 IC API 的编程,用户几乎可以实现所有的 CAXA 实体设计界面交互执行的所有功能。除此之外,一个二次开发用户还能定制和增强许多附加功能,例如:使用 EDO(Extensible Data Object)属性,回调机制来定制用户交互行为,增加用户数据等。使用用户充分利用三维数字化软件的已有功能来开发自己所需的新功能,CAXA 实体设计二次开发平台的主要对象库框架图如图 2 和图 3 所示。

在 CAXA 实体设计中,创建实体采用自顶向下的设计方法。例如:在一个设计环境中创建一个块特征的零件,设计步骤是首先调用 ZSceneDoc::CreatePart 创建零件,再用 ZPart::CreateBlock 创建特征。这项技术防止了创建出一个没有父对象的、与其它对象关系摇摆不定的对象。

2.2 命名规范

由于二次开发平台的对象库极为庞大,为二次开发平台建立一个科学的命名规范是非常重要的。IC API 在函数名称、参数名称、数据类型、CAXA 实体对象类型等多方面采用了自己的独特规范,命名规范采用自解释机制,并且保持一致性。遵循了以下原则:

- (1)对象名字直观,最终用户在交互时会根据名字或多或少地了解它的功能。
- (2)每个对象都有一个默认的接口,它的名字一般是“IZObjectName”。例如,对象 Part 有一个默认的接口 IZPart。命名规范中之所以包含字母“Z”,是为了防止名字冲突所采取的简单措施。
- (3)一个对象所支持的接口的名字通常采用“IZObjectName****”格式。
- (4)有些通用接口会被多个对象所支持,它的名字可以不包括对象的名字。例如,IZElement 是一个通用接口,Zpart、Zassembly、ZextrudeFeature 等对象都支持它。
- (5)一个对象所支持的事件接口的名字一般为“_IZObjectNameEvents”,如果它是一个由多个对象支持的通用事件接口,那么名字一般为“_Z****Events”。
- (6)查询函数(或方法)的名字是“Get****”或“Is****”格式。
- (7)执行特定操作的函数(或方法)的名字通常以这个操作所关联的一个动词作为开头。例如“Create****”、“Delete****”、“Move****”等。

(8)为了方便采用 C++ COM 技术的用户,专门设计了一系列的方法和工具。这些方法和工具的名字都以 COM 作为结尾。例如:通常情况下名字为 CreateLine 和 Data 的函数,所对应的专门用于 C++ COM 技术的函数名字就是 CreateLineCOM 和 DataCOM。

(9)其它规则:获得个数的函数的名字采用如 GetChildren ElementsCount 的格式,“个数”在名字中对应 Count,“条目”在名字中通常叫 Item。

2.3 二次开发向导

二次开发向导可以使用户快速创建一个应用的 Add-in,大大提高用户的工作效率,CAXA 实体设计定制了一个二次开发向导 ICAddinAppWizard.awx,并可按照下面的设置创建一个 ATL 工程。

- (1)微软基类库(MFC)的支持;
- (2)应用程序事件接收器(Application Event Sink Implementation);
- (3)用户可选择的事件处理信息;
- (4)文档事件(Document Events);
- (5)交互和拾取事件(Interactor and Selection Events);
- (6)将来用户可能添加的事件;
- (7)带有二个按钮的工具条。

在<add-in project>\Registration 下生成两个相应的 CAXA

实体设计的 add-in 注册文件，分别为 debug 版和 release 版。

2.4 回调函数

为了简化事件处理的回调过程，IC API 使用了 4 个类：CZAppEventsSink, CZDocEventsSink, CZInteractorEventsSink 和 CZSelectEventsSink，分别应用于应用事件(Application Events)，文档事件(Documentation Events)，交互事件(Interactor Events)和拾取事件处理 (Selection Events)。

这 4 个类包含了事件执行过程的全部细节，用户只需要从这 4 个类中的一个类派生出一个新的类，重载基类中的虚函数，分别使用 Advise 和 Unadvise 来注册事件和取消事件的注册。

2.5 扩充数据对象 (Extensible Data Object)

扩充数据对象 EDO 使二次开发用户具有扩充 CAXA 实体设计功能的强大能力。一个 EDO 对象就是任何使用 IZExtensionDataObject 接口的对象。一旦一个对象使用这个简单的接口，它就成为 CAXA 实体设计应用的一个完整部分。

一个二次开发用户通过创建一个 EDO 对象来定制或扩充 CAXA 实体设计功能，用户开发的 EDO 对象可以简单、也可以复杂，具有强大的灵活性。一个简单的 EDO 对象可以是仅仅为一个设计元素存储一些扩充数据，一个较复杂的 EDO 对象可以是将一些基于扩充数据的应用实例具体数据存入文件，一个更复杂的 EDO 对象可能完全控制一个设计元素行为、持久性以及与用户交互等方面。

一个 EDO 对象是通过任何设计元素的 IZExtensibility 接口与 CAXA 实体系统联系起来，成为 CAXA 实体层次结构中一个有机组成部分。一个 IZExtensibility 接口允许二次开发用户对一个设计元素对象增加、获取、去除 EDO 对象，每一个 EDO 对象通过类标识符 (CLSID) 进行识别。一个设计元素可以增加任意个不同类标识符 EDO。如果一个二次开发者想在一个设计元素中增加一个具有相同类标识符的多个 EDO 对象，那么他可以创建一个 EDO 组，并将这个 EDO 组增加到该设计元素上。

一个 EDO 的生命周期是由 CAXA 实体设计管理的，当一个关联 EDO 的设计元素被加载，CAXA 实体设计即创建 EDO。当一个关联 EDO 的设计元素被删除，那么与之相关联 EDO 即被删除。当系统已执行增加一个 EDO 请求的话，CAXA 实体将返回一个运行的 EDO 实例。当二次开发用户在 CAXA 实体设计中增加一个 EDO 对象，用户可以让一个设计元素的所有实例都共享这个 EDO，也可只关联到一个具体实例。

一旦 EDO 对象加入 CAXA 实体设计系统，该 EDO 对象将改变甚至替代所关联设计元素的操作行为，这是通过一系列回调接口来完成。EDO 只执行所感兴趣的回调操作，当改变和重载这些回调接口函数，其相应操作就发生变化。目前开发的回调接口有：

IZExtensionDataObject——鉴别 EDO 的接口，并提供总体信息

IPersistStream ——为 EDO 存储和读取数据的操作

IZEDOInteractionCallback—— 改变用户与一个被选中的设计元素的交互的操作

IZEDOMouseCallback—— 改变一个设计元素在执行鼠标交互时的操作

IZEDORenderCallback—— 改变一个设计元素被画入一个文档的操作

IZEDOHittestCallback—— 改变一个系统确定的一个设计元素被选中的操作

IZEDOSelectionCallback—— 当一个元素被选中或取消选中的时候，获取相关 EDO 的操作

IZEDODragAndDropCallback——当一个元素被拖动或释放的时候，获取相关 EDO 的操作

IZEDOUUpdateCallback——改变一个元素更新的操作

2.6 内存管理

CAXA 实体设计二次开发接口按照 COM 的标准方法来管理内存。当多种非标准类型参数（例如对象、字符串）作为某方法的传入或传出参数，以及某些 COM 对象的属性时，制定了以下几条规则：

(1)对于传入参数，调用者负责分配内存，同时在调用返回后负责释放。

(2)对于传出参数，方法或函数负责分配内存，调用者负责释放。

(3)对于既传入又传出的参数，调用者负责分配内存，方法和函数可以随意地释放和重新分配，最后由调用者统一释放内存。

2.7 出错处理

所有的 IC API 都有一个 HRESULT 类型的返回值来指示函数调用是否成功。但是返回值并不总意味着函数实现了它的功能。例如：函数 IZElement::GetFirstChild 只是判断这个 Element 对象是有子项的话就返回 S_OK，IZElement::GetFirstChild 可以不返回任何子项 Element。因此除了检查返回值之外，还总是检查返回的接口指针是否为 NULL。如果函数调用出错了，ICAPI 通过标准的 COM 错误对象提供了丰富的错误信息，开发人员可以获得详细的调用失败信息。

2.8 Undo 处理

Undo 处理为 CAD 用户提供了很大的方便。IC API 通过 IZDoc 接口提供了 3 个方法来处理撤销和恢复，这 3 个方法是：StartUndoTransaction，EndUndoTransaction 和 AbortUndoTransaction。StartUndoTransaction 使用在撤销处理开始的位置。EndUndoTransaction 被用来完成或者说是提交最后一次 StartUndoTransaction 调用后所作的的所有操作。点击 CAXA 实体设计上的撤销按钮就可以回到 StartUndoTransaction 调用之前的状态。AbortUndoTransaction 放弃最后一次 StartUndoTransaction 调用后所作的的所有操作，即用户操作可以方便地返回到 StartUndoTransaction 调用之前的状态。

3 二次开发实例——弧齿锥齿轮三维数字化造型

目前，多个行业的用户在 CAXA 实体设计二次开发平台上开发了专用系统，“基于加工过程的弧齿锥齿轮三维数字化造型系统”是北京北海海尔软件有限公司在 CAXA 实体设计二次开发平台上完成的。

弧齿锥齿轮在许多机械设备中是不可取代的关键动力传动部件，被用于传递相交轴的轴运动。具有重迭系数大、承载能力高、传动平稳及对安装误差敏感度小等优点。弧齿锥齿轮设计原理复杂，而且与加工参数（机床调整参数、刀具参数等）密切相关。设计与制造的计算过程涉及 500 多个参数，不同的加工参数可以制造出符合同一张工程图纸要求、但传动性能（啮合性能、强度性能、动态性能等）完全不同的齿轮产品^[3]。

至今为止，弧齿锥齿轮的三维数字化造型的研究存在着两个问题：一是没有考虑具体的加工方法及机床、刀具和轮坯的参数，使得构造的弧齿锥齿轮三维模型只是理论上的近似，而与实际加工出来的弧齿锥齿轮有着较大的出入，这样的三维模型是“似是而非”的，只能用来做示意性的演示，而无法在工程中真正应用，如弧齿锥齿轮的有限元分析、数

(下转第 8 页)