

浅析基于 CAA 的 CATIA 二次开发^{*}

Brief Analysis on Secondary Development of CATIA Based on CAA

中航工业北京航空制造工程研究所 梁岱春 张为民 隋立江

[摘要] 概括介绍了 CATIA 的客户化定制工具、CAA C++ 二次开发及其开发环境 RADE, 运用 CAA 的 CATIA 二次开发技术实现了 CATIA 产品结构树的遍历, 零部件父子装配关系的提取及零部件属性信息的提取。

关键词: CATIA COM 二次开发

[ABSTRACT] Tools customized for CATIA, secondary development of CATIA based on CAA and RADE is briefly introduced. By secondary development of CATIA based on CAA, traverse of product structure tree, father-child assembly relationship and extraction of attribute information are realized.

Keywords: CATIA COM Secondary development

CATIA 是法国达索公司与 IBM 公司开发的 CAD/CAE/CAM 一体化软件, 广泛应用于航空航天、汽车、船舶及电子工业, 尤其在航空航天业, CATIA 基本上占据了垄断地位^[1]。CATIA 如此备受人们的青睐, 这主要是因为它具有超强的自由曲面造型及逆向功能、人性化的界面、智能型的功能操作以及全面组合分析功能。

随着 CATIA 软件应用领域的不断扩大和深入, 为满足用户对软件个性化服务的要求, CATIA V5 开放了大部分接口。CATIA 二次开发是对其软件进行个性化和专业化的有效手段, 可以使软件更好地为用户服务, 对提高工作效率和产品质量, 节约研发成本和缩短开发周期有着重要的作用。国内采用组件应用架构(Component Application Architecture, CAA)进行 CATIA 的二次开发已有几年, 但相关的资料相对较少, 通过对 CATIA 二次开发技术进行一些探讨, 以期为其他相关研究提供一些借鉴。

1 CATIA 客户化功能定制工具

CATIA 客户化功能定制工具主要包括 4 种: 用户定义特征、知识工程、自动化应用接口、组件应用架构。前

2 种为 CATIA 系统本身提供的定制工具, 应用过程中局限性比较大; 后 2 种是二次开发工具, Automation API 入门容易, 但功能限制大, CAA C++ 入门困难, 但提供的接口最全面, 实现的功能最强大, 开发的程序效率高, 能够满足用户深层次专业化的要求。

(1) 交互式的用户定义特征(Interactive User Defined Feature, IUDF)。

IUDF 是一种编制式的定制开发, 通过聚合现存的特征来交互定义新的“IUDF”(用户定义特征)。“IUDF”可以通过引用一个目录文件(Catalog)保存在 .CATPart 文档中, 它可以交互地被客户使用。

(2) 知识工程(Knowledge Ware)。

CATIA 系统的知识工程是一种基于规则、面向目标的客户化方式, 它能够提供最完整的预定义应用服务, 并通过函数、规范、组件和系统等方式实现知识的管理和重用。各类知识可以自行定制, 并能够与外部代码集成。知识工程主要用于 3 个方面: 知识顾问、知识专家和产品设计工程优化。

(3) 自动化应用接口(Automation API)。

Automation API 提供了交互的方式, 可通过录制宏(自动生成 VBScript 等)或 Visual Basic 平台等方式实现用户所需功能。该应用接口的模型设计功能比较全面, 并具备了与任何 OLE(对象连接与嵌入技术)所兼容的平台进行通信的能力。

(4) 组件应用架构(CAA C++)。

CAA C++ 是基于组件的定制开发, 是对其组件对象的组合和扩展, 采用了标准接口(COM)技术和对象连接与嵌入(OLE)技术。标准接口技术作为一种软件架构具备了更好的模块独立性和可扩展性, 使 CAA 的程序设计更加容易且趋于标准化。

2 CAA C++ 二次开发及其开发环境 RADE

CAA C++ 这种组件式的解决方案采用了开放式、可扩展的模块化开发架构, 使得全球诸多开发商可以参与 Dassault Systemes 产品的研发。对客户而言, CAA 可以进行从简单到复杂的二次开发工作; 可以做精致的交互命令, 这些命令可以分组归入工作台(workbench)并展

^{*} 大型复合材料壁板轮廓加工工艺及加工单元结构研究项目(20101625008)资助。

示精美的面板;用户可以根据需要定义并产生与其他 Dassault Systemes 特征平等的新特征,从而丰富 CATIA V5 的数据模型。

(1) CAA 的架构与 CAA C++ 二次开发。

CAA 的组件架构如图 1 所示,CAA 由一系列模块组成,在 CAA 架构的支撑下,点击相应的模块,即可进入各个模块的文档。构成 CAA 组件的模块文档数量是很多的,而且各个模块文档内容之间相互渗透、关联。CAA 采用了组件对象模型(COM)和对对象连接与嵌入(OLE)技术;CAA C++ 是 CATIA 的一套 C++ 函数库,该函数库在 CATIA 运行时加载,可利用其开发的应用程序与 CATIA 通信。

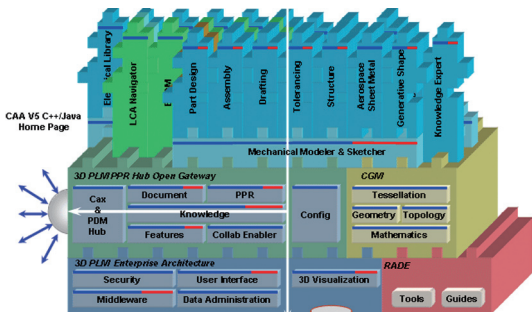


图1 CAA V5组件架构

Fig.1 CAA V5 component architecture

CATIA V5 的体系结构是一种面向行业应用的垂直框架体系结构,组件是 CAD/CAM 系统的各种几何特征和管理、分析单元。框架是一些具体的应用对象,包括 2D/3D 建模、分析、混合建模、制图、数控加工等,CATIA V5 也称为领域(domain)或应用(application)。系统通过 3D PLM PPR (Products Process Resource) HUB 将产品总线连接起来。CAA C++ 部分框架及 ProductStructure 框架下的接口和类如图 2、图 3 所示。

CATBaseUnknown 继承自 IUnknown 接口,是所

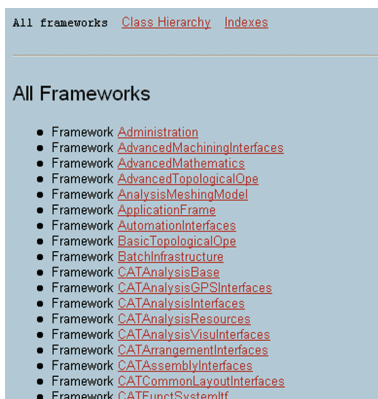


图2 CAA C++部分框架

Fig.2 Part framework of CAA C++

ProductStructure Framework

Interface Index

- [CATIAsmNavigateStruct](#)
- [CATIBlockMovable](#)
- [CATIContextualProduct](#)
- [CATICtxMenuProvider](#)
- [CATICustomName](#)
- [CATIMovable](#)
- [CATIPrdObjectPublisher](#)
- [CATIPrdProperties](#)
- [CATIProduct](#)
- [CATIProductFileSelection](#)
- [CATIProductInSession](#)
- [CATIProductSelection](#)

Class Index

- [CATPrdCommonFileSelection](#)

图3 ProductStructure 框架下的接口和类

Fig.3 Interface and class of ProductStructure

有 CAA 组件和接口的基类,它包含了对 CAA 内部接口和组件进行管理的方法。此外,它还为 CAA 中所有应用于组件的类提供了 IUnknown 的基本方法:接口查询(QueryInterface)和对象生存期管理(AddRef 和 Release)。QueryInterface 用于查询同一对象的其他接口;AddRef 与 Release 用于控制引用计数(Reference Counting)。引用计数用于控制对象的生命周期,它是一个整数,当对象被创建时,引用记数从 0 开始增加,当调用 AddRef 时,对象的引用记数加 1;当调用 Release 时,对象的引用记数减 1;当对象的引用记数为 0 时,对象被自动释放^[2]。

(2) 开发环境 RADE。

CAA 的实现,是通过提供的快速应用开发环境 RADE (Rapid Application Development Environment) 和不同的 API 接口程序来完成的。快速应用开发环境 RADE 是一个可视化的集成开发环境,它提供了完整的编程工具组。实际上 RADE 以 Microsoft Visual Studio C++.NET 为载体,在 .NET 环境中增加了 CAA 的开发工具。API 提供了操作各种对象的方法和接口。可以说 CATIA CAA RADE 是当前所有高端 CAD 软件中开发环境最复杂、最强大的。RADE 在 Microsoft Visual Studio C++.NET 2003 中添加的部分菜单如图 4 所示。

3 CAA C++ 二次开发过程

下面用实例说明如何使用 CAA C++ 二次开发来建立一个 CATIA 内部程序。实现的功能是新建一个独立的工作台(workbench),并在其下面实现添加自定义菜单、并生成对菜单的响应。菜单实现了 CATIA 产品结构树的遍历及属性信息获取。

(1) 新建工作空间。

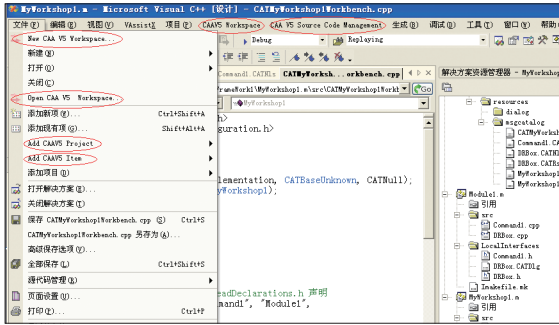


图4 RADE 在Microsoft Visual Studio C++.NET 2003中添加的部分菜单

Fig.4 Some menus inserted into Microsoft Visual Studio C++.NET 2003 by RADE

工作空间将二次开发形成的一系列命令集中在一个 workbench 中,便于以后操作。新建工作空间如图 5 所示。

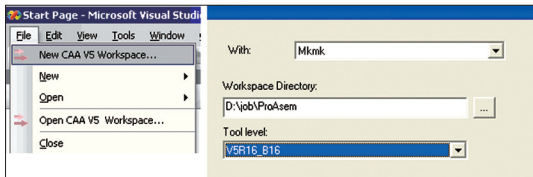


图5 新建工作空间
Fig.5 New workspace

(2) 载入 API。

确定开发所需要的 API 资源有 3 种方式:

- 链接安装路径下的 API;
- 拷贝安装路径下的 API 到工程里,链接工程里的 API;
- 拷贝安装路径下的 API 到指定路径下,链接该路径下的 API。

确定完方式后,由 RADE 调用相关资源。载入 API 如图 6 所示。

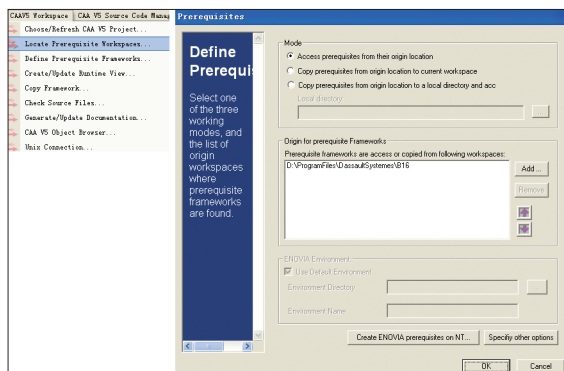


图6 载入 API
Fig.6 Load of API

(3) 创建 module。

工作空间完成后添加 module,主要开发工作在 module 里面进行。大型的开发为了程序的可读性与可维护性,工作空间下会包含很多 module,通常将实现不同功能的程序放在不同的 module 中。创建 module 如图 7 所示。

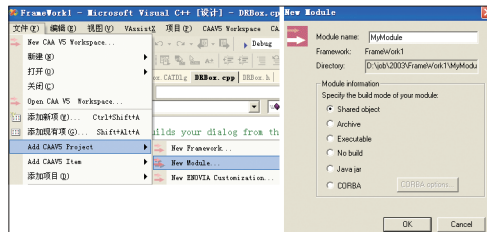


图7 创建 module
Fig.7 New module

(4) 创建菜单。

当前工作空间的类为 MyWorkshop, MyWorkshop 继承于 CATBaseUnknown,可自动建立 CreateWorkbench() 和 CreateCommands()2 个函数。函数 CreateWorkbench() 中实现插入菜单,函数 CreateCommands() 中实现菜单和响应函数的关联。最后生成的菜单如图 8 所示。

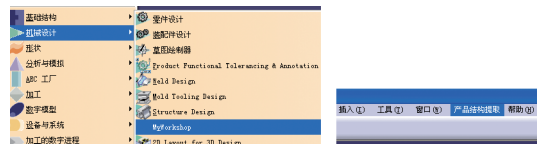


图8 生成的菜单
Fig.8 Menu created

(5) 程序实例。

无论在 CATIA 还是 PDM (Product Data Management) 中,产品结构树都是 BOM (Bill of Material) 的关键部分,与菜单关联的响应函数运用 CAA C++ 实现了 CATIA 产品结构树的遍历,零部件父子装配关系的获取及零部件属性信息的获取,主要用到了 ProductStructure 框架下的 CATIPProduct、CATIPrdProperties 等接口。CATIA 产品中零部件父子装配关系及零部件属性信息的获取流程如图 9 所示。

该流程的部分代码如下:

```
// 获取当前文档
CATFrmEditor * Product =CATFrmEditor::GetCurrent
Editor();
CATDocument *pDoc=Product->GetDocument();
// 获取文档的根节点
CATIDocRoots* piDocRootsOnDoc = NULL;
HRESULT rc = pDoc->QueryInterface ( IID_
```

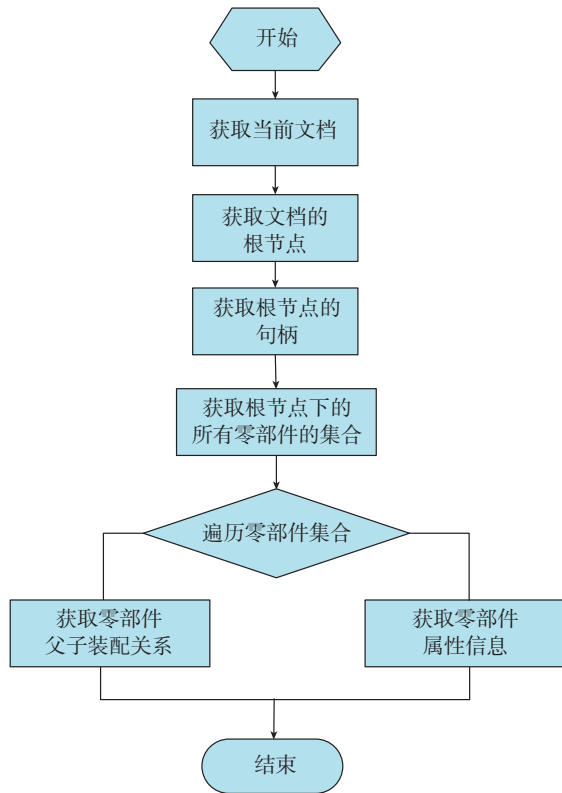


图9 零部件父子装配关系及零部件属性信息获取流程
Fig.9 Flowchart of father-child assembly relationship and attribute extraction

```

CATIDocRoots, ( void** )&piDocRootsOnDoc );
    CATListValCATBaseUnknown_var* pRootProducts =
piDocRootsOnDoc->GiveDocRoots();
    CATIPProduct_var spRootProduct = NULL_var;
    spRootProduct = ( *pRootProducts )[1];
    // 获取根节点的句柄
    CATIPProduct *piProductOnRoot = NULL;
    rc = spRootProduct->QueryInterface ( IID_
CATIPProduct, ( void** )&piProductOnRoot );
    // 获取根节点下的所有零部件的集合
    CATListValCATBaseUnknown_var* childrenList =
piProductOnRoot->GetAllChildren();
    // 遍历零部件集合, 获取父子装配关系及零部件的
属性信息
    CATUnicodeString TextspChild="";
    CATUnicodeString PartNumber="";
    CATUnicodeString FatherNumber="";
    CATIPProduct_var spChild = NULL_var;
    CATIPProduct_var spFather = NULL_var;
    int childrenCount = childrenList->Size();
    for ( int i = 1; i <= childrenCount; i++ )
  
```

```

{
    // 父子装配关系的获取
    spChild = ( *childrenList )[i]; // 当前节点
    spFather = spChild->GetFatherProduct(); // 当前节点
的父节点
    PartNumber=spChild->GetPartNumber(); // 当前节
点零部件号
    FatherNumber=spFather->GetPartNumber(); // 当前节
点的父节点零部件号
    // 属性信息的获取
    CATIPProduct *piChildProduct = NULL;
    spChild->QueryInterface ( IID_ CATIPProduct,
( void** )&piChildProduct );
    CATIPrdProperties_var spPrdProps
( piChildProduct );
    if ( NULL_var != spPrdProps )
    {
        spPrdProps->GetNomenclature ( TextspChild );
        // 获得术语信息
        spPrdProps->GetRevision ( TextspChild ); // 获
得版次信息
        spPrdProps->GetDefinition ( TextspChild ); // 获
得定义信息
        spPrdProps->GetDescriptionRef ( TextspChild );
        // 获得描述信息
    }
    piChildProduct->Release();
    piChildProduct = NULL;
}
  
```

4 结束语

利用 CAA C++ 开发的功能与原系统的结合非常紧密,如果没有特别的说明,无法把客户所研发的功能从原系统中区分出来,这非常有利于用户的使用和集成,具有很好的应用价值。但是,由于 Dassault Systemes 应用系统本身的复杂性和 CAA 所涉及的深层次内容,利用 CAA 进行二次开发必然具有一定的复杂性和难度,用户在熟练掌握 Dassault Systemes 应用系统的同时,还必须具备扎实的软件开发知识和能力^[3]。

参考文献

- [1] 尤春风 .CATIA V5 机械设计 .北京:清华大学出版社,2002.
- [2] 潘爱民 .COM 原理与应用 .北京:清华大学出版社,1999.
- [3] 刘俊堂 .CAA 及其应用 . [2004-08-01]. <http://blog.csdn.net/littlechen/articles/57782.aspx>.

(责编 深蓝)