

基于 .NET 平台的 SolidWorks 二次开发研究与实践

阳 程,周临震,李 亮

(盐城工学院 优集学院,江苏 盐城 224003)

摘要:分析了 .NET 平台下非托管 Solidworks COM 组件访问的技术原理与实现方法,对该平台下 Solidworks COM 对象向 Solidworks IA 对象模型转变规则作了比较详细的归纳总结。并运用实例阐述在 .NET 环境下进行 Solidworks 二次开发的实现过程与步骤。

关键词:.NET;Solidworks 插件;组件对象模型;互调程序集

中图分类号:TP391.72 **文献标识码:**A **文章编号:**1671-5322(2008)04-0026-05

Solidworks 是基于特征的三维参数化造型软件,对其进行二次开发可以遵循参数化 CAD 的原则。与以往非参数化 CAD 系统相比,参数化技术很大程度上提高了二次开发的效率^[1]。并且 Solidworks 提供了完整的、免费的二次开发应用编程接口,用户可以使用众多的编程工具,开发出用户自己的应用系统,从而得到功能强大的企业级产品。

Microsoft .NET(简称 .NET)是微软公司推出的新的软件开发平台,综合体现了软件技术的最新发展。.NET 的集成开发环境 (IDE) Visual Studio,以及由公共语言运行环境和 .NET 框架基本类库构成 .NET 平台基础架构——.NET 框架,使 .NET 成为结合了 Java 和 Visual Basic 的优点,又能提供 C++ 的强大功能的优秀开发平台。

1 .NET 平台下 Solidworks 二次开发的基本原理

Solidworks 二次开发分为两种:一种是基于自动化技术的,生成 .EXE 的文件;另一种开发方式是基于 COM 的,开发 Solidworks Add-in,也就是 Solidworks 的插件,生成 *.dll 文件。COM 技术可以使用最多的 Solidworks API,并可控制 Solidworks 运行的方式^[2]。这种方式开发出的应用程序可直接嵌入到 Solidworks 内部,程序菜单直接出现在 Solidworks 的主菜单上,效率较高^[2]。

1.1 Interop Assembly

Solidworks Add-in 开发是基于 COM 技术的开发方式。.NET 托管环境下不支持对非托管 COM 组件的直接访问。在 .NET Framework 中,提供了一项专门用于在 .NET 代码中直接访问 COM 组件的 COM Interop 的技术。它的基本原理是:.NET Framework 能够自动针对某一个 COM 组件,帮助开发人员生成一个 Interop Assembly (互调程序集,后面简称 IA),IA 是一个完全的托管程序集。IA 中的名称空间、类、方法等,都是与那个 COM 组件对应的。在我们自己的应用程序中,只需要调用一个纯粹 .NET 的 IA 即可,这时 IA 就会帮我们再去访问真正的 COM 组件。

1.2 生成 Solidworks IA

Solidworks API 是以 COM 组件形式提供的,在 .NET 平台下要调用其 API 必须先将其转换成 .NET 平台下的 Interop Assembly (互调程序集)^[4-5]。

COM 类型库一般是独立的 TLB 文件、DLL 文件或 EXE 文件。当找到包含目标 COM 类型实现的类型库后我们使用类型库导入程序 (tlbimp.exe) 来生成 Interop 程序集。Tlbimp.exe 可以通过运行 Visual Studio 命令行工具,输入 "Tlbimp" 命令启动。Tlbimp 命令格式如下:

tlbimp COM 类型库文件 /keyfile:密钥文件路径 \filename.snk /out:目标文件.dll

使用.NET进行Solidworks二次开发时,我们首先要利用.NET的类型库导入程序对Solidworks的COM对象进行转换。Solidworks的COM类型库对象封装在其安装目录的3个TLB文件里,分别是:SldWorks.tlb、SwConst.tlb和SWPublished.tlb。另外,对于项目要用到的其他的COM类型库也需进行相同处理。将上述文件单独拷贝到新建的项目文件夹下面。然后运行Visual Studio命令行工具,启动DOS模式。按上述Tlbimp命令格式输入命令对每个COM类型库文件进行处理。如转换SldWorks.tlb文件输入命令为:tlbimp SldWorks.tlb /keyfile: keyPair.snk /out: SldWorks.dll。

命令执行的结果是生成Interop程序集,如上述命令会生成SldWorks.dll文件。这样,我们就可以在.NET中通过引用生成的DLL文件,对COM对象进行操作了。

Solidworks从2004版开始支持.NET,它会在我们的开发环境Microsoft Visual Studio.NET新建项目向导中添加一个Solidworks二次开发模板(Solidworks Addin)。进行二次开发时选择此模板,可以自动生成二次开发框架代码。此后,通过IDE在项目中添加对我们转换后的Interop程序集的引用即可调用Solidworks API进行编程开发^[6]。

2 深入浏览Solidworks IA

在Solidworks IA中,已经把Solidworks COM组件进行了封装,但是Solidworks的IA中的类、接口,并不是和Solidworks的COM组件中的类、接口一一对应的。出现这个情况的原因在于.NET的类型库导入程序TLBIMP帮我们根据COM组件生成Interop Assembly时,其实是不会一一对照COM组件来生成.NET类和接口的。它会根据一定的规则,来生成对应的.NET类和接口。下面,以Solidworks COM中的Sldworks这个接口为例,把它的转换规则简要的说明一下。在Visual Studio.NET的对象浏览器中,我们会发现它有7种不同的形式,所有的都声明为Public类型,7种形式如下:

```
Interfaces
ISldWorks
DSldWorksEvents
DSldWorksEvents_Event
```

```
SldWorks
Delegates
DSldWorksEvents _ * EventHandler ( 16 of them)
Classes
DSldWorksEvents_SinkHelper
SldWorksClass
```

理清这个复杂的转过程需要追溯到Solidworks的SldWorks对象的COM原始定义。SldWorks对象的COM coclass(简称组件对象类——component object class)如下所示:

```
coclass SldWorks {
    [ default ] interface ISldWorks;
    [ default, source ] dispinterface DSldWorksEvents;
};
```

它有2个接口:一个是叫ISldWorks的主接口,另一个是叫DsldWorksEvents的事件接口。

在生成IA时,TLBIMP会从COM coclass中直接导出ISldWorks和DsldWorksEvents接口,ISldWorks接口基本描述了Solidworks COM组件中的SldWorks类中的所有操作和属性,DSldWorksEvents接口基本描述了Solidworks COM组件中的SldWorks类中的所有事件。DsldWorksEvents事件接口实际编程中不会用到,它会被TLIMP进一步处理并生成以下3种类型:首先,它创建了DSldWorksEvents_Event。这种形式看起来跟DsldWorksEvents很相似,但是它以事件类型代替了DsldWorksEvents中的诸多方法。另外它创建了名为DSldWorksEvents_ * EventHandler(共16个,对应SldWorks对象的16个事件)的委托,这里,*代表相应的方法名,这些方法名都在原始的DsldWorksEvents接口中有对应。最后,它创建了DSldWorksEvents_SinkHelper,这个我们很少用到,可以忽略不管。

TLBIMP在处理SldWorks coclass时,会创建1个SldWorksClass的.NET类和1个SldWorks的.NET接口。Sldworks接口基本描述了Solidworks COM组件中的SldWorks类中的所有操作、属性、事件等等,因为它完全继承自ISldWorks和DSldWorksEvents接口。SldWorksClass是实现SldWorks接口的具体类

那么,在这7种形式中有哪几个是我们真正要用到的?在你的代码中如不需要处事Sldworks

对象的事件,只要用 ISldWorks 接口,如需捕获并处理事件,则要用到 SldWorks 接口(从 ISldWorks 和 DSldWorksEvents_Event 中派生)和 DSldWorksEvents_ * EventHandler 委托。在二次开发编程实践时,你可忽略其它的几种形式,当作不存在。

3 利用.NET Solidworks Addin 开发模板开发 Solidworks Add-in

3.1 利用 Solidworks Addin 开发模板生成与 Solidworks 接口的代码框架

在.NET 开发环境 Microsoft Visual Studio .NET 中新建项目时,选择 Solidworks Addin 开发模板会出现一个向导,向导允许我们指定生成的插件名称,选择要监听的事件,以及决定是否生成特性管理器页面等。最后向导会生成一些类文件。图 2 为 Solidworks 渐开线齿轮三维造型系统 Gear 项目的解决方案资源管理器截图。

向导为我们生成的代码提供了一个与 Solidworks 接口的代码框架。其中 Gear 项目同名文件 Gear.cs 文件中的 Gear 类可理解为一个与 Solidworks 接口的类,我们可以利用它来完成在 Solidworks 中添加菜单、工具栏及向特性管理器页面添加控件等基本的工作。

图 1 为折叠后以大纲方式显示的 Gear.cs 文件,其中 Local Variables 区定义了 Gear 类的成员变量,ISwAddin Implementation 区实现了 ISwAddin Implementation 接口的 ConnectToSW() 和 DisconnectFromSW() 方法,这两个方法分别为插件的入口点和出口点,加载和卸载插件时分别被执行。UI Methods 区主要是用来给 Solidworks 用户界面添加/删除菜单、工具条和特性管理器页面等。

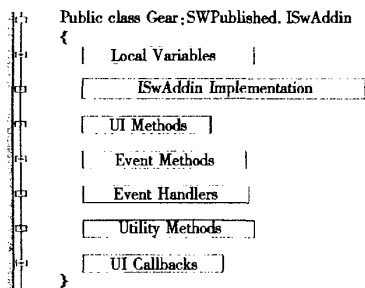


图 1 折叠后以大纲方式显示的 Gear.cs 文件

Fig. 1 The outline of the Gear.cs file

Event Methods 区为事件处理作准备,具体来说是将事件绑定到具体的对象及声明事件委托。Event Handlers 区包含对 SldWorks 对象几个典型事件万方数据

件的处理方法。Utility Methods 区只包括一个方法:public Object GetSldWorks(), 它用来得到 SldWorks 对象。UI Callbacks 区用来定义对在 UI Methods 中添加的菜单、工具栏等的响应函数。

3.2 为 Solidworks Add-in 添加交互对话框

在开发中,交互对话框是必不可少的。下面以 Gear 项目为例,说明添加窗体的一般方法和步骤。

3.2.1 添加一 windows 窗体,取名为 GearForm

该界面设置如图 2 所示。

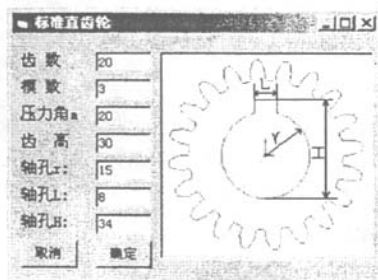


图 2 界面设置

Fig. 2 Interface design

在 GearForm.cs 中需做以下一般操作:

(1) 在 GearForm.cs 中添加 2 个引用:

```
using SldWorks;
```

```
using SWPublished;
```

(2) 在窗体类 GearForm 中添加 1 个成员变量:

```
private ISldWorks iSwApp;
```

(3) 为窗体类 GearForm 增加 1 个带参数的构造函数:

```
public GearForm(Gear theApp){
```

```
InitializeComponent();
```

```
iSwApp = (ISldWorks) theApp. GetSldWorks(); //iSwApp 指向获得的 ISldWorks 对象}
```

3.2.2 在 Solidworks 中添加菜单、子菜单及菜单响应函数

(1) 在 Gear.cs 文件的 Local Variables 区中添加一个窗体变量:

```
GearForm Dlg;
```

(2) 在 Gear.cs 文件的 UI Method 区 AddMenus() 方法中输入以下代码:

```
//添加一级菜单" Gear ";
```

```
iSwApp. AddMenu(docType, "Gear", pos);
```

```
//在 Gear 菜单中添加下级菜单,并指定菜单
```

响应函数为 ShowGearDlg ();

```
iSwApp.AddMenuItem2(docType,addinID,"  
渐开线齿轮三维造型系统@ Gear",-1,"  
ShowGearDlg","MenuEnable","齿轮参数化建  
模");
```

(3)在 Gear.cs 文件的 region Utility Methods 区中编写菜单单响应函数 ShowGearDlg():

```
public void ShowGearDlg(){  
    this.Dlg = new GearForm(this);  
    this.Dlg.Show();//显示窗体  
}
```

经过以上几步,我们就完全实现了 GearForm 类和 Solidworks 的接口,我们可以在窗体上进行代码编写完成相应的功能。本例为在“确定”按钮 btnApply 的 Click 事件处理函数中编写代码,代码如下:

```
private void btnApply_Click(object sender,  
System.EventArgs e){  
    this.part = (IModelDoc2) this.iSwApp.New-  
Part();//新建零件图  
    this.part = (IModelDoc2) this.iSwApp.ActiveDoc;//取得活动文档  
    this.GearCurve();//一个自定义方法,用来  
绘制渐开线齿廓  
    this.Drawing();//一个自定义方法,对渐开  
线进行拉伸操作,同时生成轴孔特征  
}
```

3.3 Add-in 编译与调试

插件不能象控制台应用程序和 Windows 应用程序一样可以在 Visual Studio.NET 中直接运行,在准备编译与调试时,需要对 add-in 项目的一些默认设置作以下修改:

1)在解决方法管理器中右击该 add-in 项目,选择“属性”。

2)在“属性”对话框的左侧,选择“通用属性”,接着选择它下面的“常规”,把输出类型设为“类库”。

3)在“属性”对话框的左侧,选择“配置属性”,接着选择其下面的“调试”。把调试模式设置为“程序”,把启动应用程序设置为 Solidworks 可执行文件的目录。

4)完成代码编写后,可以在 Visual Studio.NET 中点击“生成”菜单,选择“生成解决方案”,直接编译项目,会生成.dll 文件,本例为 Gear.dll。万方数据

5)使用.NET 的 regasm. 工具注册插件。成功编译生成.dll 文件后,对于其他的插件使用者(如将编译生成的*.dll 文件拷贝到其它机器上供用户使用),则需通过.NET 的命令工具 regasm.exe 将编译好的*.dll 文件注册后方可供 Solidworks 调用。

图3为 Gear.dll 插件在 Solidworks 下运行的结果。

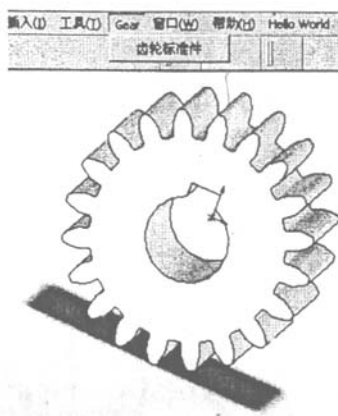


图3 程序运行结果

Fig.3 Program results

4 用.NET 开发 SolidWorks 的一般步骤

.NET 平台下进行 Solidworks 二次开发的一般步骤总结如下:

(1)安装 SolidWorks 软件、SolidWorks API SDK 和 Microsoft Visual Studio.NET。

(2)启动 Microsoft Visual Studio.NET,新建一个项目,项目类型选择 Visual C#项目,模板选择 SolidWorks C# Addin。选择开发模板的好处是可以调用模板自动生成一个接口类,该类实现了与 SolidWorks 的接口。

(3)把项目要用到的 COM 类型库文件转化为.NET Interop 程序集。并在项目中添加对该程序集的引用。

(4)在接口类中编写代码,通过调用 SldWorks 对象的 AddMenu() 函数和 AddMenuItem2() 函数添加菜单和主菜单。

(5)为每个子菜单写菜单命令处理函数。

(6)编译。编译后生成一个*.dll 文件。

(7)注册.NET 插件为 COM 组件。通过.NET 的命令工具 regasm 将编译好的*.dll 文件

注册为可供 Solidworks 调用的 COM 组件。

(8) 在 Solidworks 软件中加载插件。启动 SolidWorks, 选择打开文件的类型为 Add - Ins (* .dll), 打开编译好的 * .dll, Solidworks 软件即可加载插件。

5 结束语

本文分析了 .NET 平台下非托管 Solidworks

COM 组件访问的技术原理, 并对 .NET 平台下 Solidworks COM 对象向 Solidworks IA 对象模型转变规则作了比较详细的归纳总结。最后以一个实例特别阐述了在 .NET 环境下开发 Solidworks 的实现过程与步骤。

参考文献:

- [1] 李向阳. SolidWorks 二次开发的研究[J]. 精密制造与自动化, 2008(2): 56 - 58.
- [2] 赵韩, 吴振华, 陈兴玉, 等. 基于设计计算的齿轮三维设计系统的研究[J]. 机械传动, 2008, 32: 47 - 49.
- [3] 江洪, 魏峥. Solidworks 二次开发实例解析[M]. 北京: 机械工业出版社, 2004.
- [4] 庄成三. 微软. NET 平台软件技术[M]. 成都: 电子科技大学出版社, 2003.
- [5] 王宗彦, 吴淑芳, 秦慧斌, 等. Solidworks 机械产品高级开发技术[M]. 北京: 北京理工大学出版社, 2005.
- [6] 叶伟威, 余才佳. Solidworks 2006 实体建模与二次开发教程[M]. 北京: 国防工业出版社, 2006: 9.

The Development of Application System Based on .NET and SolidWorks

YANG Cheng, ZHOU Lin-zhen, LI Liang

(UGS College, Yancheng Institute of Technology, Jiangsu Yancheng 224003, China)

Abstract: This paper analyzes the technical principles and methods of accessing unmanaged COM object model of Solidworks in .NET environment, provides a summary on the transitional rules from Solidworks COM into Solidworks IA and offers an example to illustrate the processes of Solidworks development in .NET.

Keywords: .NET; Solidworks Add - in; COM; Interop Assembly

(责任编辑: 张英健; 校对: 范大和)