

用 Delphi 集成 ActiveX 控件的方法

方红雨 (合肥轴承总厂微机房)
崔逊学 (中国科技大学自动化系)

摘要: 本文介绍了在 Delphi 中利用打包机制来实现应用系统集成和包容 ActiveX 控件的方法,并给出了实例。

关键词: Delphi 控件 集成

一、概述

控件编程是目前计算机发展网络化、程序设计模块化的需求。在 COM(控件对象模型)架构下,人们可开发出各种功能专一的软件积木块,然后按需要“搭”起来,构成复杂的应用系统。其主要作用是可将系统中的控件用新的控件替代,以随时进行系统的升级与定制,且可在多个应用系统中重复利用同一个控件。但一个关键技术是如何使自己开发的应用系统具备集成和包容控件的功能,笔者将从开发实践中总结出的方法与读者共享。

COM是控件对象之间相互接口的二进制规范,与源代码无关。COM对象即使由不同的编程语言创建,或运行在不同进程空间与操作系统平台,也能相互通信。本文的控件是指 ActiveX 控件,其基于 Windows 环境下的控件模型开发标准,可由多种软件开发工具制作。

COM 由一组规范和一个系统级的实现构成,包括 COM 核心、结构化存储、统一数据传输和智能命名。COM 还为控件提供了接口,使用者在实现组件功能时,需调用组件接口函数。COM 提供了两种不同类型的接口: IUnknown 和 IClassFactory。IUnknown 是默认祖先接口,提供了 COM 库的核心服务,包括引用计数和查询其他接口的功能。任何一个接口,除了它自身的成员外,还继承了它的祖先接口中的成员,包括 IUnknown 接口中的成员。IUnknown 接口中的方法包括:

- QueryInterface // 向对象查询由参数指定的接口
- AddRef // 增加一个对象的引用记数
- Release // 减少对象引用计数,计数值为 0 时则释放对象

在 Delphi 开发环境中的 System 单元(位于 /Borland/Delphi4/Source/Rtl/Sys 目录), IUnknown 接口声明如下:

```
IUnknown=interface
  ['{00000000-0000-0000-C000-000000000046}']
  function QueryInterface(const IID:TGUID;out Obj):
  HRESULT;stdcall;
  function AddRef:Integer;stdcall;
  function Release:Integer;stdcall;
end;
```

值得注意的是,在 COM 模式中一个接口尽管可从其祖先接口中继承方法,但并未继承实现祖先接口中的代码,即接口必须重新实现祖先接口中的方法。此问题的详述可参见文献 3。

另一类接口是 IClassFactory,当类工厂向系统注册时,它为指定的类创建对象。

对于被创建和调用的 COM 对象,其方法属性通常是一个可执行程序。如果使用某个 COM 对象,则须知道对象所支持的接口信息。对象在创建后,则可接受申请使用特定的接口,并通过接口来操作对象。

二、Delphi 环境下 ActiveX 控件集成的实现

1. 控件集成的意义和方法

控件以可视化的图标显示给用户,要求即插即用,嵌入到支撑环境(如窗体)的控件重用的是目标代码。这是一种黑箱重用,即控件的封装在物理上是隐蔽的。ActiveX 控件能集成到支持 ActiveX 的容器程序中运行,如 Delphi、VB 和 IE 等,但不支持 ActiveX 的平台则不能运行这种控件。

集成控件或使程序具备控件包容器功能的方法,一般有两种:

(1)当需要使用一个定制好的 ActiveX 控件时,使用 Windows提供的对象注册工具 regedit.exe 对该控件进行登录注册。使用该控件时,根据系统注册表定位控件,将 COM 的请求送至 OCX。但是随着 OCX 文件的增多,Windows系统的速度将越来越慢,注册表也变得很庞大。由于专业领域的控件只在使用特定应用软件时才会用到,平时一般并不使用,所以使用这种方法不仅占据了宝贵的内存资源,而且或许会产生许多错误。另外在控件版本升级时,以前的版本文件会占用硬盘空间,形成垃圾文件且不容易被清理。

(2)自己编写注册文件,使用 Delphi 提供的包(Package)方式集成控件或称作控件打包。因为自己规定控件注册机制,容易控制被包容的控件,如实现增加、删除、修改、替换和升级控件等。控件对象只有在使用时才调入内存,定位有关的控件地址和接口。控件文件存放在特定目录下,升级版本文件容易替换旧版本文件。

2. 集成控件的原理

通常应用系统的可视化功能可看作是由控件和容器两部分作用的结果。容器类似装配车间,是存放控件的“器皿”。有关消息传递和协调机制则由对象请求代理(ORB)来实现,只要每个控件在设计时都按照 ActiveX 接口规范声明自己的方法,就可将其登录注册加入控件系统。ORB负责实施控件之间的互操作和动态数据交换,允许 COM 对象跨越机器和操作系统发生作用。

集成控件实质是建造容器管理器,将应用程序的窗体作为 ActiveX 控件的容器,ActiveX 控件嵌入至窗体,即定制窗体。这样在平台上编辑控件时所见即所得,满足了可视化的要求。客户程序与控件之间的通信是通过服务控制机制来实现的。如果控件位于进程内,则可直接调用;如果控件位于进程外,则调用先到达服务控制机制提供的 Proxy,再由 Proxy 生成服务对象的远程过程调用来实现

3. 实现过程和典型例子

考查下面一个典型的控件集成的例子:ActiveMovie 控件是 Windows 操作系统中媒体播放器的核心部件,它有很多属性和事件,Windows98 已提供一个后缀名为 OCX 的文件 AMOVIE.OCX,我们不知道也无须了解其由何种编程语言制作,只知其中有一个注明属性和事件的控件对象(通常的 OCX 文件可以包括多个控件)。实现此控件集成的步骤如下:

(1)制作自己的后缀名为 DPK 的包(Package)文件。向

包文件添加控件并编译,可查看包中列表的控件图标和名称 TactiveMovie。据笔者经验,一般 Delphi 环境提供的控件(后缀名为 pas 的文件)可直接添加单元至包中;对于已生成 OCX 文件的控件须先在 Delphi 机制下进行注册、再添加,即先进行 Register OLE Control 操作,否则报错。生成的包文件中自动产生了有关的 .PAS 和 .DCR 文件(如 Amovie_TLB.dcr 和 Amovie_TLB.pas)。

(2)建立自己的注册机制并注册所有控件。主程序的 Pascal 代码关键片断如下:

```
Unit MainForm;
Interface
Uses Amovie_TLB,.....; // 引用包中 PAS 文件
Procedure TFMainForm.ReadClass;
Begin
    RegisterClasses([TactiveMovie,.....]);
    // 在单线程中注册控件的常规类名
End;
Procedure TFMainForm.LoadDefaultPalette();
Begin
    with Inifile do begin
        WriteString('Palette','控件工具箱','
TactiveMovie;.....;');
        //在初始化文件中写入控件名称及所处控件面板位置
    End;
End;
```

这里凡是应用到该控件的单元,须在 Uses 子句中引用包中其相关的执行单元文件名(如AMovie_TLB)。因为该 Pascal 文件包含了使用类型的声明、各控件的 GUID 和一些复杂的接口定义。

(3)选择和装配应用控件。选择控件就是使用鼠标“拖/放”实现对控件的操纵,在事先分门别类好的控件工具箱中点击需要的控件图标或名称,拖拽至自己窗体的适当处并放置。由于每一控件包含许多元素如属性、方法和事件,这里使用类似 Delphi 平台的 Object Inspector 的对象观察窗来显示控件的属性和事件,允许用户修改属性,将控件装配到自己的系统中,通过编译生成器形成用户最终的专家系统。

如何在应用程序的控件面板上动态产生所有的控件图标或名称,下面是实现类似 Delphi 开发环境中的控件面板的部分源代码:

```
procedure CreateButton(ButtonName:string);
```

```
var
  TabSheet:TTabSheet;
  Button:TSpeedButton;
begin
  Button:=TSpeedButton.Create(TabSheet);//产生加
速棒按钮
  with Button do begin
    Name:=UniqueName(Button);
    Font.Size:=8;
    Font.Style:= [fsUnderLine];
    Caption:=ButtonName;//加速棒按钮显示控件名称
    Flat:=True;
    GroupIndex:=1;
    Height:=25;
    Hint:=ButtonName;
    Left:=Index*110;
```

```
Parent:=TabSheet;
ShowHint:=True;
Top:=5;
Width:=105;
OnClick:=sbButtonClick;//单击事件调用另一
```

过程

```
end;
end; ■
```

参考文献

- 1 Brockschmidt Krig, *Inside OLE(Second Edition)*. Washington:Microsoft Press, 1995
- 2 杨美清, 软件复用及相关技术, 计算机科学, 1999, 26(5)
- 3 徐新华, COM CORBA和Internet编程技术, 北京:人民邮电出版社, 1998
- 4 Dale Rogerson 著, 杨秀章译, COM技术内幕, 北京:清华大学出版社, 1999