

ActiveX Automation 及其在 CAD 系统开发中的应用

徐宝霖¹, 黄君洲², 陈金水²

(1. 福州晋安建筑工程公司, 福建 福州 350000; 2. 河海大学计算机及信息工程学院, 江苏 南京 210098)

摘要 介绍了 ActiveX Automation 的原理及其开发的方法, 讨论了以 Visual BASIC 为控制器, 通过 ActiveX Automation 实现与 Auto CAD 集成的方法和技术. 文中列举了该方法链接、访问 Auto CAD 对象(命令、文件等)以及创建、修改和查询图形对象的具体步骤, 并且结合实例给出了 Visual BASIC 程序. 该方法可实现 Windows 环境下一体化 CAD 系统的开发.

关键词 计算机辅助设计; 系统开发 接口

中图分类号 TP391.72 **文献标识码** A **文章编号** 1000-1980(2000)05-0114-05

ActiveX 是一组软件技术和软件方法的总称, 其目标是实现不同语言开发的软件组件在单机或网络环境下能够相互操作, 从而使得应用程序的开发过程成为组件架构的过程, 如同搭积木一般. 组件(Component)是在计算机上运行的实实在在的软件, 一个组件类似于一个微型应用程序, 已编译、链接好, 并可以使用. 每个组件都实现一定的功能, 组件的实现细节对使用组件的客户隐藏, 它展示给客户的仅仅是设计好的输入与输出. ActiveX 是对 OLE 的扩充, 较之以前的 OLE, ActiveX 更符合 C/S 体系, 也更能有效地开发基于网络的分布式 CAD, 乃至各种应用系统.

1 ActiveX 及其实现的基础 COM

ActiveX 基于 COM. 所谓 COM(Component Object Model)^[1], 即组件对象模型, 是关于如何建立组件以及如何通过组件架构应用程序的一个规范. 按照 COM 规范建立的组件称为 COM 组件. COM 客户通过接口(Interface)访问 COM 组件. 对于 COM 来说, 接口是一个指向虚拟函数表(vtbl)地址的指针, 虚拟函数表每一项包含的是一个由组件所实现的函数的地址. 图 1 是它的接口示意图.

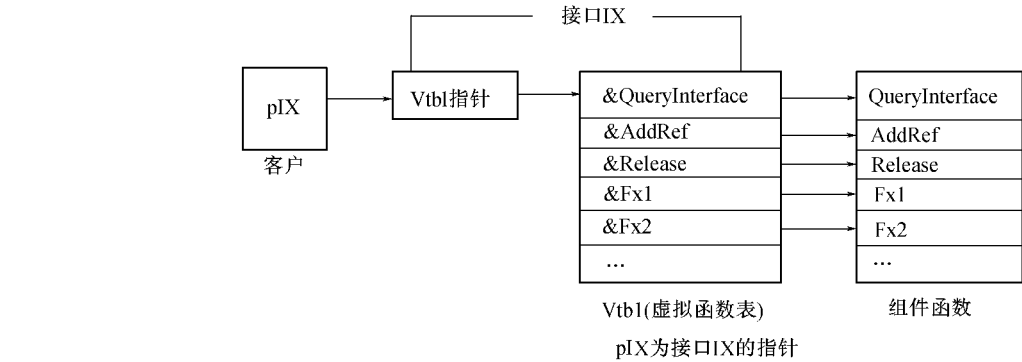


图 1 接口 IX 示意图

Fig.1 Interface IX

所有的接口都继承一个名为 IUnkown 的接口, 所以所有接口的 vtbl 的前三项都为 IUnkown 接口的成员函数 QueryInterface, Addref 和 Release 的地址. QueryInterface 函数用来查询组件接口, 获取指定接口的指针. Addref 函数和 Release 函数主要管理组件的生存周期.

ActiveX 采用 C/S 模式并通过 COM 接口来实现它的性能.ActiveX 提供的主要性能有 :ActiveX Automation (ActiveX 自动化)、ActiveX 控件、ActiveX 文档.图 2 给出了 ActiveX 的 C/S 实现模型.

2 ActiveX Automation(ActiveX 自动化)

ActiveX Automation(以前称为 OLE Automation)是 ActiveX 为 Windows 用户和程序员提供的一项重要技术. ActiveX Automation (ActiveX 自动化),或简称 Automation(自动化),是一种跨应用的组件编程模式(paradigm,或方法),它使得一个自动化服务器(EXE 或 DLL)能把自动化对象展示给一个自动化控制器,自动化控制器可以通过编程来驱动控制这个 EXE 或 DLL.图 3 是 ActiveX Automation 的 C/S 实现模型.

- 下面给出几个主要术语的定义：
- a. 自动化服务器(Automation Server) 是可以由其他应用程序编程驱动的一个 EXE 或 DLL,一个自动化服务器包容一个或多个自动化对象.
 - b. 自动化对象(Automation Object) 是可编程对象,它包括一组属性和方法.它们为自动化控制器提供一种特定的功能.
 - c. 自动化控制器(Automation Controller) 或称客户,是使用或操纵自动化服务器的应用程序.

ActiveX Automation 的核心部分依赖 IDispatch 接口,它是一个通用功能的接口.一个自动化服务器实际上就是一个实现了 IDispatch 接口的 COM 组件,而一个自动化控制器则是一个通过 IDispatch 接口同自动化服务器通信的 COM 客户.与 COM 模型不同,IDispatch 接口可以接收一个函数名称并执行它,而在 COM 模型中,客户需要获取函数在 vtbl 中的索引.因此,在 IDispatch 接口和 COM 组件中间需要提供一种机制来实现利用函数名对函数的间接调用.这个任务由调度接口来完成.调度接口的实现依赖于 IDispatch 的两个成员函数 GetIDsOfNames 和 Invoke. IDispatch 接口及调度接口如图 4 所示.

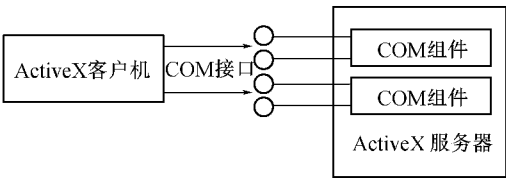


图 2 ActiveX 的 C/S 实现模型
Fig.2 C/S implementation model of ActiveX

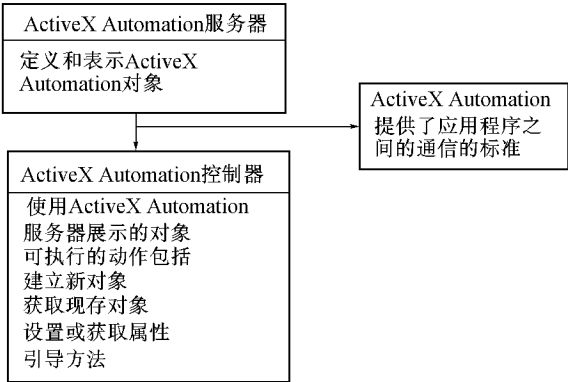


图 3 ActiveX Automation 的 C/S 实现模型
Fig.3 C/S implementation model of ActiveX Automation

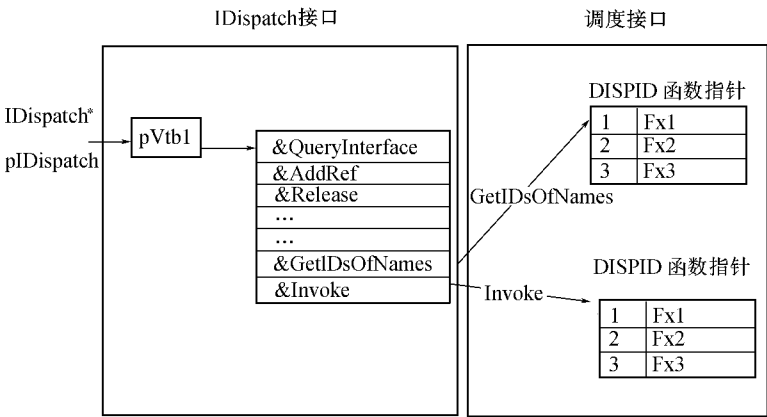


图 4 IDispatch 接口与调度接口
Fig.4 IDispatch interface and attemper interface

客户访问自动化组件的过程如下 :当客户访问自动化组件时,自动化服务器首先要创建一个称为类工厂(Class Factory)的组件,这个组件的唯一功能就是创建其他组件.类工厂创建后,会返回一个指向一个称为 IClass Factory 的接口的指针,客户程序可以通过接口 IClassFactory 访问类工厂,并创建要访问的服务器组件的一个实例.服务器组件创建后,返回一个 IDispatch 接口指针.当客户调用某个函数时,将函数名传给 IDispatch

接口, IDispatch 接口调用成员函数 GetIDsOfNames 返回函数的调度 ID, 即 :DISPID, DISPID 唯一标识一个函数. 随后 IDispatch 接口调用其成员函数 Invoke, Invoke 函数将 DISPID 作为函数指针数组的索引, 按照 DISPID 调用相应的函数完成任务.

IDispatch 与一类型库或 typelib(.tlb 文件) 相关, 此类型库主要是公布(publishes)有关接口的信息. 它可以使这些信息适用于比如 VB 或 VBA 等编程语言. 这些信息包括 : (a) 方法名字、返回类型和参数. (b) 别名、枚举和结构. (c) 对象的描述. 一个对象可以是一个模块、一个接口或一个组件对象类. (d) 引用来自其他类型库的类型描述.

类型库中也可以包含组件、接口以及函数的帮助串. 使用对象浏览器(如 :VB 的 Object Browser), 程序员将可以方便地获取任意属性或方法的说明.

3 在基于 AutoCAD 的 CAD 系统开发中的应用

AutoCAD R14 支持 ActiveX Automation, 为其他应用程序同 AutoCAD 的集成提供了极大的方便^[2]. 通过 ActiveX Automation, 其他应用程序可以方便地利用它向应用系统开发者提供集成支持的对象, 并且控制、操作这些对象. 这里, AutoCAD 作自动化服务器(AutoCAD 的自动化对象层次结构见 AutoCAD 相关文档).

下面以 Microsoft Visual Basic (VB)^[3] 作为自动化控制器, 结合实例, 详细介绍 VB 通过 ActiveX Automation 与 AutoCAD 集成的主要过程.

VB 是第一个使用 ActiveX Automation 的商业化软件, 它向用户隐藏了 ActiveX Automation 的实现细节, 为程序员访问自动化对象提供了极大的方便. VB 使用 ActiveX Automation 与 AutoCAD 集成的几个主要步骤为 :

a. 设置主要的对象变量, 并实现与 AutoCAD 的链结(Link)

假设程序中包含模块, 在模块中定义以下对象变量 :

```
public acadApp As Object           'AutoCAD Application 对象
public acadDoc As Object           'AutoCAD Document (drawing) 对象
public moSpace As Object           'model space 集合对象
public paSpace As Object           'paper space 集合对象
```

注意 :AutoCAD 不支持事先捆绑(early binding), 所以不能进行如下定义 :

```
Dim moSpace As IacadModelSpace
```

当你在 Visual BASIC 的工程项目中引用(Reference)了 AutoCAD R14 Object Library 时, IacadModelSpace 成为可用类型出现在 VB 的变量类型中.

与 AutoCAD 链结(Link) 引用 AutoCAD 的 Application 对象.

```
On Error Resume Next
```

```
Set acadApp = GetObject(, "AutoCAD.Application")
```

如果 AutoCAD 已经在运行, 用函数 GetObject 获取 AutoCAD 的 Application 对象

```
If Err Then
```

```
Set acadApp = CreateObject("AutoCAD.Application")
```

如果 AutoCAD 没有运行, 用函数 CreateObject 创建 AutoCAD Application 对象

```
If Err Then
```

```
MsgBox Err.Description
```

```
Exit Sub
```

```
End If
```

```
End If
```

可以利用 Application 对象的属性定置 AutoCAD. 例如, 设置 AutoCAD 为可见.

```
acadApp.Visible = True
```

获取 Application 对象后, 就可以利用它去访问其它对象.

b. 利用 Document Object 访问 AutoCAD 中的绘图文件(Drawing)

打开一个 .dwg 文件 :

```
Set acadDoc = acadApp.ActiveDocument
Dim dwgName As String
dwgName = "C:\acadr14\sample\campus.dwg"
If Dir(dwgName) <> "" Then
    acadDoc.Open dwgName
Else
    MsgBox "File "& dwgName & "does not exist."
End If
```

也可以利用 New 和 Save 方法去创建和保存一个 .dwg 文件.

c. 在 AutoCAD 中画图：

画一条曲线(polyline)：

```
Set moSpace = acadDoc.ModelSpace
Dim lwpolyObj As Object
Dim ptArray(0 To 5) As Double
ptArray(0) = 2                第一个点的坐标(下标 0 和 1)
ptArray(1) = 4
ptArray(2) = 4                第二个点的坐标(下标 2 和 3)
ptArray(3) = 2
ptArray(4) = 6                第三个点的坐标(下标 4 和 5)
ptArray(5) = 4
Set lwpolyObj = moSpace.AddLightWeightPolyline(ptArray) 画曲线
```

d. 查询和修改图形对象(Graphical Object)

返回当前模型空间和图样空间中的图形对象总数(Count)：

```
Dim noMSObj,noPSObj As String
noMSObj = Str(moSpace.Count)
noPSObj = Str(paSpace.Count)
MsgBox "Objects Mspace = "& noMSObj & "Pspace = "& noPSObj
```

利用 Item 方法,可以获取实体集合中特定实体的一些信息.

```
Dim ent As Object                定义对象变量
Dim msgStr,NL As String
NL = Chr(13) & Chr(10)          创建换行符
Set ent = moSpace.Item(0)        设置对象为第一个实体
msgStr = "Information about the first model space entity. "& NL & NL
msgStr = msgStr & "Entity name :"& ent.Entity Name & NL
msgStr = msgStr & "Layer :"& ent.Layer & NL
msgStr = msgStr & "Color :"& Str(ent.Color)& NL
msgStr = msgStr & "Linetype :"& ent.Linetype & NL
msgStr = msgStr & "Linetype scale :"& Str(ent.LinetypeScale)& NL
msgStr = msgStr & "Handle :"& ent.Handle & NL
msgStr = msgStr & "Visibility :"& ent.Visible
```

以上获取了一条直线的属性

```
MsgBox msgStr
```

修改实体的属性：

```
Dim ent As Object
Set ent = moSpace.Item(0)
```

```
ent.Color = acRed
ent.Update
Dim p1(0 To 2) As Double
Dim p2(0 To 2) As Double
p1(0) = 12
p1(1) = 5
p2(0) = 12
p2(1) = 7
If ent.EntityName = "AcDbLine" Then
    ent.StartPoint = p1
    ent.EndPoint = p2
    ent.Update
Else
    MsgBox "The first object is not a line!"
End If
```

4 结 束 语

ActiveX Automation 极大地改善了 AutoCAD 同其他软件的通信手段,它使得其他软件对 AutoCAD 的操作更加方便.本文对 ActiveX 及 ActiveX Automation 的概念及原理做了简要的阐述,并且举例说明如何利用 ActiveX Automation 实现 VB 同 AutoCAD 的集成.给出的实例虽然简单,但它们是最基本的,读者只要掌握了这些方法,就不难开发出一个耦合紧密、效率较之以前的常规方法更高的 CAD 系统.

参考文献:

[1] 潘爱民. COM 原理及应用. 北京:清华大学出版社,1999. 376 ~ 396.
[2] 郭朝勇. AutoCAD R14 二次开发技术. 北京:清华大学出版社,1999. 383 ~ 397
[3] Evangelos Petroutsos 著. Visual BASIC 5 从入门到精通. 北京:电子工业出版社,1997. 62 ~ 107.

ActiveX Automation and Its Application in the Development of CAD System

XU Bao-lin¹, HUANG Jun-zhou², CHEN Jin-shui²

(1. Jinan Architecture Engineering Corporation, Fuzhou 350000, China;
2. College of Computer and Information Engineering, Hohai Univ., Nanjing 210098, China)

Abstract :ActiveX Automation is a new technology of computer software development based on software components. The theory and development method of ActiveX Automation are presented. The steps to use Visual BASIC as a controler to link, to visit objects of AutoCAD (command, file, etc), and to create, edit and find graphics by ActiveX Automation under Windows operating system are listed. The present method can be used as a reference in integrated CAD system development.

Key words :computer aided design ;system development ;interface