

# 基于COM的ActiveX控件 在网络中的应用

匡满 程瀚依 李光明 李茜 (西北轻工业学院自控系 712081)

**摘要:** 通过对软件设计规范COM的介绍,引出COM的一个具体的实现--ActiveX控件,并简要介绍动态字控件SwayWord的制作方法及其在网络中的应用。

**关键词:** COM ActiveX 控件 网络 VC++

## 1 引言

快速开发强壮的开放式应用程序是现代科技迅猛发展的要求,如何更好的实现应用程序间的交互与协作以及计算机间的网络通信,如何使所开发的软件程序具有良好的鲁棒性、可扩展性和可维护性,如何缩短开发周期是软件界关心的重要问题。

COM正是解决以上问题的一种强有力的集成技术。它作为建立在面向对象技术和组件化程序设计思想基础上的程序设计新标准,不仅给出了程序之间的通信规范,也提供了实现组件程序的基础结构。它可方便的将多个应用程序联合在一起,共同来完成一项工作。

ActiveX控件是建立在COM基础之上的原OLE控件在网络上的扩展,它可以作为为网页提供交互内容的工具,并具有适于网上工作的轻便性。现今,它已成为一种通用控件广泛应用在软件设计中。下面结合COM对ActiveX控件加以介绍。

## 2 COM及其接口

COM是Microsoft公司开发的描述软件体系结构的一种模型。它的基本思想是将复杂的应用程序分散设计成一些小的、独立的功能模块,然后通过彼此间的接口进行通信,实现无缝结合,来组建开放的应用程序。COM组件作为一个二进制对象具有以下特征:

- (1) 良好的可移植性、易于扩展;
- (2) 独立于编程语言、支持多种平台;
- (3) 采用单一的编程模式进行通信、具有对象间共享存储器管理;

(4) 对象能够动态的装卸;

(5) 提供丰富的错误和状态报告等功能。

其中,每个功能模块(COM组件)都是独立于任何应用程序的DLL或EXE。它是预先构造好且测试无故障的部件,程序开发者可以放心地选用所需的组件进行新程序的组装。从而有效地缩短了开发周期,增强了程序的鲁棒性和可扩展性。

COM接口是COM技术的关键所在。接口是C++结构,它定义了一组纯虚函数,代表着COM对象可以被客户访问到的功能。使用COM组件的客户只能通过获取指向接口的指针来访问对象。这表明客户只知道组件可以完成什么样的工作,而不了解它内部的具体实现。从而实现了较安全的调用对象,并且与它所在的操作系统、硬件、程序设计语言无关。

COM的所有接口都基于IUnknown接口派生而来,通过对它的实现可以获得各个需要的接口指针。一个组件允许有多个接口,接口一旦建好便不可更改,需要提供新功能时通过增加新接口来实现。从而保证旧的客户仍可使用旧接口,新客户则使用新接口的工作方式。程序在运行时可以确认调用的是哪一个接口。

## 3 ActiveX 技术

ActiveX是新版的OLE技术,它不但能够实现原来的功能,并且将这种技术扩展到网上。提供开发和实现网上应用程序的技术和部件。它具有事件触发、持久性(能够保持其属性和方法的状态)、可移植性(适用于任何OLE容器)、OLE自动化(即用户可以用来修改控件属性的属

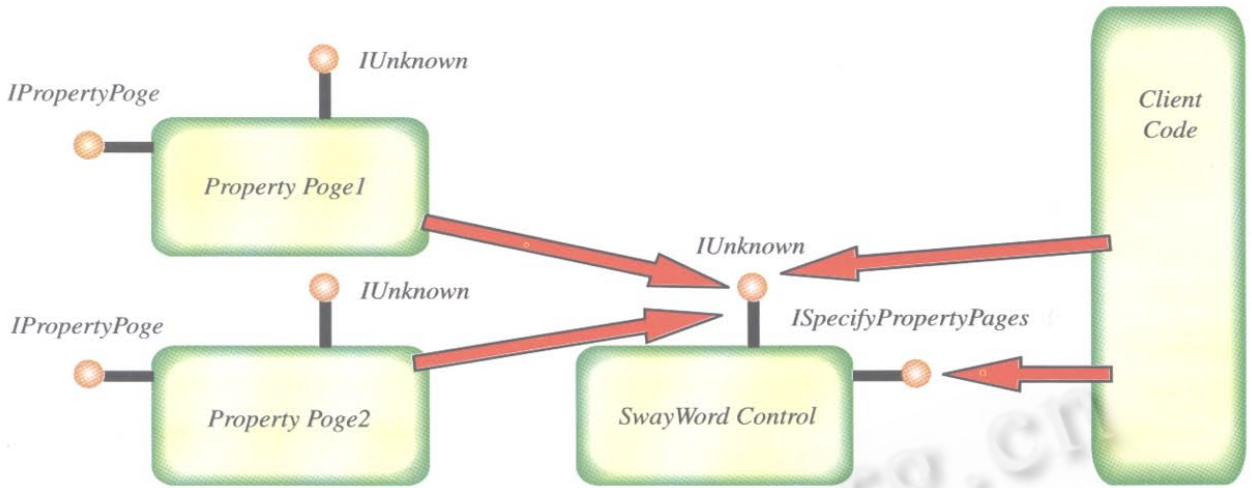


图 1 控件、属性页及客户代码的通信方式

性页)等特性。目前提供的以OLE技术为核心的功能项有ActiveX文档、异步存储、ActiveX控件、Internet标记等。在此我们主要讨论一下ActiveX控件。它采用了新的标准-简化的COM,也就是说ActiveX控件首先是一个COM对象,它必须支持IUnknown接口和自登记。同时考虑到网上的需要,控件还应该尽可能地轻便,它不必像OLE控件那样包含全部COM接口,而只支持它需要的接口。

#### 4 ActiveX 控件的制作方法

制作该控件主要有三种方法:

4.1 利用原始C++代码编辑,一切工作都由手工完成,程序员将承受巨大的高强度劳动;

4.2 利用MFC创建,这种方法适于创建重量级的或包含复杂图形的本地服务器,它的缺点主要是受到的限制较多,程序必须在包含MFC DLLs的系统中才能运行,故使控件变得庞大;

4.3 利用ATL创建,它是一种轻便的快速开发方法,创建的组件不需要完全包含MFC自动提供的内置功能。并且ATL做了大量的工作来进行内存处理:初始化类厂和类标识符、实现定义类型库的代码和控件版本信息、定义控件的接口Ids、说明控件类型信息、在系统注册表中添加或删除控件等。

#### 5 一个ActiveX 控件实例

我们利用VC++6.0中的ATL COM对象向导制作了一个动态字控件,并把它加到网页上以激活相应资源,该控件名为SwayWord。它是一串可以随机晃动的字符,点

击它可以激活子网页。同时在网页设计阶段可以通过它的属性页修改相应属性。下面对一些关键之处加以介绍。

(1) 添加属性和属性页。增加Short类型的Interval属性和CfontHolder类型的m\_fontDefault属性。Interval是控件内部时钟的时间间隔,m\_fontDefault是控件中摆动文本的字体(文字内容采用库存属性),ATL自动为属性添加处理函数框架,在其中我们编写需要的代码。然后为控件添加属性页。

由于ActiveX控件和属性页是分开的COM对象,它们需要通过接口进行通信。图1是属性页和控件及客户代码之间的通信图。

(2) 实现文字的摆动。为了使文字不停的摆动,在CSwayWord类中增加WM\_TIMER消息,给该消息的处理函数OnTimer添加一些代码使控件重绘。在OnDraw中调用我们编写的绘图函数Sway(ATL\_DRAWINFO& di),ATL\_DRAWINFO是一个包含设备环境的结构。在该函数中给每个字符的X、Y方向加一个随机的偏移量,通过不断的重绘使文字达到摇摆的目的。下面是具体代码。

```
void CSwayWord::Sway(ATL_DRAWINFO& di)
{ HDC MemDC; //创建内存设备环境,以备加载内存位图
  MemDC=CreateCompatibleDC(di.hdcDraw);
  //获得当前文字的长度
  int nLength=m_bstrCaption.Length();
  //用属性中的字体更新内存设备环境并保留原
  //来的字体
  HFONT pOldFont=
```

```
(HFONT)SelectObject(MemDC,&m_pFont);
// 设置背景模式并保留原来的模式
int nOldMode=
SetBkMode(MemDC,TRANSPARENT);
// 设置字体颜色并保留原来的颜色
int crOldTextColor=SetTextColor(MemDC,
m_clrForeColor);
TEXTMETRIC tmetric; // 获得当前字体的大小
// 并保存
GetTextMetrics(MemDC,&tmetric);
int nTextHeight=tmetric.tmHeight+
tmetric.tmExternalLeading*2;
int nAveCharWidth=
(int)(tmetric.tmAveCharWidth*1.5);
// 获得当前字符串边界矩形的宽和高
int nCtrlWidth=nAveCharWidth+nLength+
(nAveCharWidth/2);
int nCtrlHeight=nTextHeight+(nTextHeight/4);
// 创建当前字符串的边界矩形
RECT OldRect;
OldRect.left=0;
OldRect.top=0;
OldRect.right=nCtrlWidth;
OldRect.bottom=nCtrlHeight;
HBITMAP bmp;// 创建提供了宽和高的内存位图
bmp=CreateCompatibleBitmap(di.hdcDraw,
nCtrlWidth,nCtrlHeight);
// 更新内存设备环境中的位图并保留原位图
HBITMAP pOldbmp=
(HBITMAP)SelectObject(MemDC,bmp);
//用当前背景颜色填充内存设备环境中的字符串边界矩形
RECT rect;
rect.left=0;
rect.top=0;
rect.right=nCtrlWidth;
rect.bottom=nCtrlHeight;
LOGBRUSH bs;
bs.lbColor=RGB(0,0,255);
HBRUSH brBack;
brBack=CreateBrushIndirect(&bs);
FillRect(MemDC,&rect,brBack);
```

```
// 获得一个随机数
srand((unsigned)time(NULL));
// 下列循环用来实现字符的摆动
for (int i=0;i<nLength;i++)
{ RECT rc; // 创建单个字符的边界矩形
// 得到水平和垂直方向的随机偏移量
UINT cx=(rand()%(nAveCharWidth))/2;
UINT cy=(rand()%(nTextHeight))/4;
// 更新单个字符的边界矩形
rc.left = (nAveCharWidth*i) + cx; rc.right = rc.left
+nAveCharWidth + cx;
rc.top=cy;
rc.bottom=rc.top+nTextHeight+cy;
// 向内存设备环境出字符
DrawText(MemDC,(LPCTSTR)
m_bstrCaption [i] ,1,&rc,DT_CENTER); }
// 把内存位图粘贴到控件用户区 BitBlt(di.hdcDraw,
di.prcBounds->left+1,
di.prcBounds->top+1, nCtrlWidth,nCtrlHeight,MemDC,
0,0,SRCCOPY);
// 下面代码恢复原内存设备环境并清除位图
// 对象
SelectObject(MemDC,pOldbmp);
SelectObject(MemDC,pOldFont);
SetTextColor(MemDC,crOldTextColor);
SetBkMode(MemDC,nOldMode);
DeleteObject(MemDC);
}
```

在该程序段中我们采用函数 BitBlt()将内存位图粘贴到控件用户区,而没有采用在控件中直接绘图的方法,这样可以避免位图闪烁及跳跃现象,使其摆动得更加平滑。

## 6 使用 ActiveX 控件

制作一个HTML网页,然后利用ActiveX Control Pad工具将SwayWord控件插入设计网页。该工具打开一个类似于VB的编程窗口,通过属性页可以设置库存属性。为了在网页中得到可编程的控件,使网页首次被加入浏览器时我们可以改变控件的定制属性和库存属性,还需要做下列工作:选择Control Pad中的菜单命令,在对话框中作适当设置就可以自动生成需要的脚本代码。首先在左侧窗(下转第63页)

(上接第 56 页)

口中选定 windows 对象的 OnLoad 事件,在右侧窗口中选择 Interval 属性按下 Insert ActiveX 按钮,在弹出的对话框中输入数字 50。这样便将 SwayWord 控件的属性在页面被加载的时候设置为 50。运行时 SwayWord 控件的文字不停地在网页上摆动。另外在为控件增加事件时可以实现我们需要的功能,如单击控件以激活子网页等。此处不再赘述。

## 7 总 结

本例只是 ActiveX 控件应用的一个方面,作为可重用代码组件,ActiveX 控件有着更加深广的意义和应用前

景。此外,ActiveX 还扩充了一些独立的新技术。这些技术主要用于 web 网上交互式程序的开发,它们使 web 网页变得更加生动。这些新技术包括:超级连接、ActiveX 会议、服务器扩展、代码签名、活动电影等。■

### 参考文献

- 1 [美] Peter Norton, Rob McGregor 著,孙凤英,魏军,徐京等译,《MFC Windows 95/NT 4》,清华大学出版社,1998.4
- 2 [美] Scot Wingo 著,希望图书创作室译,《Visual C++6.0 技术内幕》(第五版),北京希望电子出版社,1999.5
- 3 [美] Corry, Mayfield, Cadman 著,刘云,孔雷译,《COM/DCOM 编程指南》,清华大学出版社,2000.1