

# 利用 COM 组件实现虚拟仪器驱动程序

樊 炜 潘 宏 浙江大学机械电子控制工程研究所流体传动及控制国家重点实验室 (310027)

## Abstract

An idea and its engineering meaning to realize virtual instrument driver through using COM module technology are introduced by an example of a digital signal analytic instrument (ActiveX COM Driver) in this paper. Furthermore, the hierarchy and developmental mode of COM driver are illustrated by using in Visual C++ IDE.

Keywords: COM module, virtual instrument, driver

## 摘 要

以一个数字信号分析仪驱动程序(ActiveX COM Driver)为例,本文介绍了利用 COM 组件技术实现虚拟仪器驱动程序的思路及工程意义,针对 Visual C++ 6.0 集成开发环境,说明了 COM 驱动程序的层次结构和开发模式。

关键词:COM 组件, 虚拟仪器, 驱动程序

## 1 前言

传统仪器是由厂家设计并定义好功能的一个封闭结构,它有固定的输入输出接口和仪器操作面板,每种仪器实现一类特定的测量功能,并以确定的方式提供给用户。虚拟仪器指的是在必要的数据采集硬件的支持下,在通用计算机平台上通过软件设计仪器的全部功能,实现传统仪器的分析处理、人机交互和显示等几部分功能模块。相对传统仪器而言,虚拟仪器具有测量成本低、功能多、灵活性强、人机见面友好等优点。

目前使用较多的虚拟仪器开发平台有如 NI 公司的 LabVIEW, Agilent (原 HP) 公司的 VEE 等,这些应用软件都需要相应的驱动程序驱动来访问具体的数据硬件设备。一般说来,即使同样的硬件设备,不同开发平台的驱动程序也是不能互换的。考虑到现有开发平台对 COM 组件的支持,可利用 COM 组件技术实现虚拟仪器驱动程序,从而做到开发一次,各平台通用,大大减少了工作量,增加了通用性、替换性。

## 2 COM 驱动程序的层次结构和开发模式

COM 组件有以下几种类型:自动化服务器(ActiveX Server),自动化控制器,控件(ActiveX Controller, OCX),文档(Document),容器(Container)。

关于 COM 的详细介绍,请参见文献 [1][2] 各种虚拟仪器开发平台(如 NI 公司的 LabVIEW, HP 公司的 VEE 等)都支持以上几种 COM 组件类型。所

以我们能够利用 COM 组件实现虚拟仪器驱动程序。

在本文的应用示例中,我们选择使用自动化服务器来实现虚拟仪器驱动程序。

COM 驱动程序的层次结构如图 1 所示。用 COM 组件实现的虚拟仪器驱动程序(ActiveX COM Driver)用于客户应用程序(Client Application)和 Windows 驱动程序(Windows Driver)的连接。Windows Driver 是基于 VxD (Virtual Device Driver)或 WDM (Win32 Driver Model)的,Windows95 和 Windows98 设备驱动程序的机理不尽相同,Windows98 不仅支持与 Windows 2000 兼容的 WDM 模式驱动程序,而且还支持与 Windows 95 兼容的虚拟仪器驱动程序 VxD。

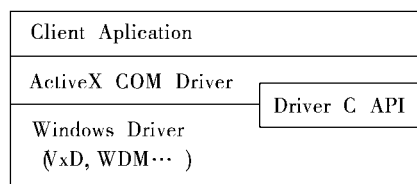


图 1 COM 驱动程序的层次结构

由于 Windows 对系统底层操作采取了屏蔽的策略,加上数据硬件设备不完全是标准外设,所以由客户应用程序发出的命令,必须经过 ActiveX COM Driver 传递给 Windows Driver,再由 Windows Driver 驱动接口送至作为外设的数据硬件设备。同样,数据硬件设备采集的数据必须经过 Windows Driver 取得,再由 ActiveX COM Driver 往上传给客

户应用程序。我们从中可以看出, ActiveX COM Driver 在层次结构中的作用不是取代 Windows Driver, 而是将 Windows Driver 封装起来, 提供给客户应用程序一个接口。

根据图 1 所示 COM 驱动程序层次结构, 我们可以使用 Microsoft VC++ 开发生成 OCX 或 ActiveX Dll(从实际上讲, OCX 就是 Dll, 只不过 OCX 是从早期控件开发中沿用下来的扩展名)。当我们需要用户界面 (UI) 时, 比较合适的开发工具是 Microsoft 基本类库 (MFC), 这是因为 MFC 非常可靠, 且能满足我们大部分的要求。首先, MFC 为创建 ActiveX 控件 (OCX) 提供了一个稳固的框架, 再者, Microsoft 在设计 VC++ IDE (集成开发环境) 充分考虑了 MFC, 并提供了很多用于应用程序开发的有用工具和 ClassWizard, 这使我们可以忽略控件实现的许多细节而专注于控件特有的功能, 从而减少了很多工作量。当我们不需要用户界面 (UI) 时, 使用 ATL (ActiveX Template Library) 作为开发工具较为方便。自从 Visual C++ 5.0 推出后, Active 模板库 (ATL) 已经成为一个完整的 ActiveX 开发框架, 成为 MFC 的替代产品。ATL 主要是针对 COM 应用开发的, 它内部的模板类直接实现了 COM 的一些基本特征, 比如一些基本 COM 接口 IUnknown、IClassFactory、IDispatch 等, 也支持 COM 的一些高级特征, 如双接口、连接点、ActiveX 控制等, 因此, ATL 特别适合建立小巧、快速的 COM 组件程序。由于大部分 COM Driver 不需要用户界面, 所以使用 ATL 作为开发工具是非常合适的, 在这里让我们看一下在 Visual C++ 6.0 集成开发环境中, 使用 ATL 开发 COM 组件的基本步骤:

(1) 运行 Visual C++ 开发环境, 从 File 菜单中选择 New, 然后从 New 对话框中选择 Project 标签, 在 Project 标签下面的类型中选择 ATL COM AppWizard, 然后填上项目名等, 单击 OK 键。

(2) 下一步是在 AppWizard 中定义 ATL 项目的基本结构。在 Server Type 选项中选择 Dynamic Link Library (DLL), 这是因为我们要创立的是 COM 组件。单击 Finish 按键继续操作。

(3) 在 New Project Information 对话框中单击 OK 键确认新项目设置。

(4) 接着为项目添加控件实现代码。从 Insert 菜单中选择 New ATL Object 菜单项。然后选择 Objects 中的 SimpleObject 图标。

(5) 下一个对话框是 ATL Object Wizard 属性

对话框, 用于定义加到项目中的新对象的属性。选择其中的 Attributes 标签, 我们可以定义此控件项目的各种特性。单击 OK 键, 就生成了新的 ATL COM 对象类。

(6) 根据需要, 在 ATL COM 对象类中加入内部使用的成员变量或成员函数。

### 3 应用示例

美国 Dactron 公司生产了一种名为 Spectra-Book 的数字信号分析仪 (8-Channel, 96Ksa/sec), 它的虚拟仪器驱动程序 (ActiveX Driver) 采用 COM 组件技术编写, 只需开发一次, 各数据硬件设备和开发平台通用。我们只需在各开发平台中设置 ActiveX Driver 的属性并调用其方法就可以方便地使用 SpectraBook。在下面的图例中, 分别演示了如何在不同的开发平台中调用 SpectraBook ActiveX Driver 的方法 (如图 2) 其中 MeasEngine 为其 COM 组件的名称:

Visual Basic:

```
set h = new MeasEngine ' Create a MeasEngine
COM object.
h.Open ' Make connection to hardware.
h.Start ' Start measurement.
n = 0;
While n < 100
    If h.DataAvailable Then ' Check if new data is
available.
        y = h.yData (1) ' Read the channel 1 data.
        :
        h.ReadDone ' Notify that I am done with
this data.
        n = n + 1
    End if
End
h.Stop '
Stop measurement.
h.Close ' Disconnect the hardware .
```

MatLab:

```
h = actxserver (' Me4x.MeasEngine '); % Create a
MeasEngine COM object.
invoke (h, 'Open'); % Make connection to VXI hardware.
invoke (h, 'Start'); % Start measurement.
n = 0;
while n < 100
    if get (h, 'DataAvailable'); % Check if new data
is available.
        y = get (h, 'yData', 1); % Read the channel 1
```

```

data.
    plot (f);
    invoke (h, 'ReadDone'); % Notify that I am
done with this data.
    n = n + 1;
end
end
invoke (h, 'Stop'); % Stop measurement.
invoke (h, 'Close'); % Disconnect the VXI hardware.
release (h); % Cleanup COM objects

```

### 4 结论

在虚拟仪器系统中,仪器驱动程序是连接仪器与用户界面的桥梁,是系统设计的一个关键。利用 COM 组件技术实现虚拟仪器驱动程序,具有极强的可交互性,只需开发一次,就适用于不同的数据硬件设备和开发平台,既方便开发,又方便维护,具有很好的工程意义。

### 参考文献

1. Dale Rogerson, COM 技术内幕,清华大学出版社
2. 潘爱民,COM 原理及应用,清华大学出版社
3. Jerry Anderson. Visual C++ 5 ActiveX 编程指南,清华大学出版社
4. LabVIEW User Manual for Windows. National Instruments Corp.,1994
5. 李扬,郑莹娜,朱铮涛,图形化编程语言 LabVIEW 环境及其开放性,计算机工程,1999.4

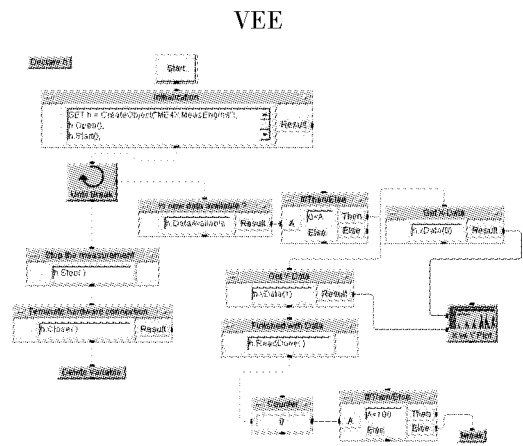
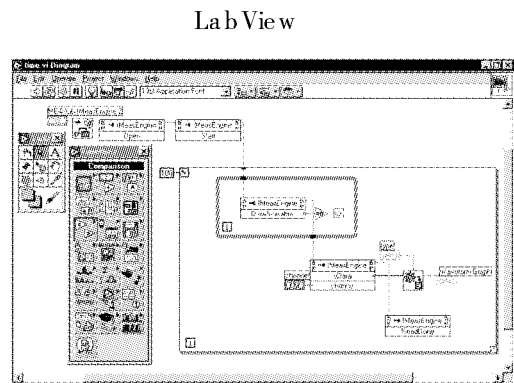
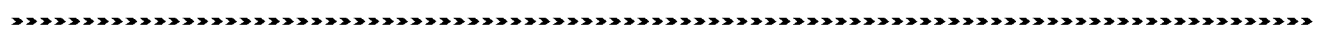


图 2 两种不同开发平台中调用 SpectraBook Active Driver 的方法



(上接第 51 页)

点数,从而启动 CGI 程序完成数据采集,并在用户浏览器屏幕上成功地看到了采集数据。

设定采用频率为 1K,取不同的采样点数并测量发出采集指令到看到采集数据的时间,结果如表 1 所示。可见,随着数据采集时间的延长,传输时间所占的比重逐渐减小,而且,在局域网范围内,由网络传输引起的延时可忽略不计。

表 1