维普资讯 http://www.cqvip.com

ActiveX 控件的工作机理与实现研究*)

王映辉
(陕西师范大学计算机科学学院　西安710062)

Study about Theory and Realization of ActiveX Control

WANG Ying-Hui
(Computer Science School of Shanxi Normal University, Xi'an 710061)

Abstract　Look through theory of ActiveX control based on large scope software(based on Internet). Analyze the framework and micro-interface map of ActiveX control. Describe the process of development and customization technology of ActiveX control by Microsoft Visual BASIC6.0.

Keywords　ActiveX control, Control container, Control interface, Interface map

1　引言

ActiveX 是对象连接与嵌入技术 OLE(Object Linking and Embedding)同 Internet 技术的结晶，是 Microsoft 为适应互联网而发展的标准[1,2]。ActiveX 建立在 OLE 标准之上，是为扩展 Microsoft Web 浏览器 Explorer 功能而提供的公共框架。ActiveX 控件是用于完成具体任务和信息通讯的软件模块[3]。

网络尤其是 Internet 正改变着人们对信息的获取、传输、发布、共享、应用和可视化等过程和方式，它为人们提供了方便的信息发布与共享方式。目前几乎所有的管理信息系统无不建立在网络或 Web(或 Internet)技术之上，这些系统的客户端大都基本采用了浏览器作为用户界面，而浏览器所能解释的语言(如 HTML 语言等)的功能是有限的。为此，采用 ActiveX 控件对其功能进行扩展是目前在 Windows 操作系统之上开发软件普遍采用的技术，也是 Windows 操作系统之软件模块得以复用的核心规范[1,4]；加之基于 Web 应用的构件技术已成为目前企业信息系统和电子商务平台等大规模软件构建的主流技术[5]。因而，透视 ActiveX 控件在大规模软件构架中的工作机理和具体实现有其重要的意义。

2　ActiveX 控件的工作机理

ActiveX 控件是 ActiveX 部件的组成成员之一。在微软的 Windows 操作系统中，ActiveX 部件一般是以扩展名为 .ocx、.dll 和 .exe 的形式提供的。ActiveX 部件包括三类，即 ActiveX 文档、ActiveX 代码部件和 ActiveX 控件。ActiveX 为应用提供了符合 ActiveX 规范的对象模块单元，这种技术允许程序员将这些可复用的部件灵活方便地通过软件总线[7]组装到应用程序之中，特别是大规模软件构架的程序之中。

2.1　ActiveX 的分类描述

既然 ActiveX 控件是 ActiveX 部件的组成成员之一，为了更进一步地透视 ActiveX 控件的本质，首先有必要将之与其它的成员进行一下简单的对比。

ActiveX 文档是可以在 Internet 浏览器上显示的窗体。在微软公司提供的许多开发工具(如 Visual Studio6.0)中将复杂的 ActiveX 技术进行了隐藏，从而使开发人员对 ActiveX 文档的设计同窗体的设计一样简单。具体开发时，在 ActiveX 文档中可以内建滚动视口、超连接等，也可以包含各类可插入的对象和显示其它的窗体；同时利用开发工具可将 ActiveX 文档打包成进程内(In-process)部件和进程外部件(Out-press)。

ActiveX 代码部件中不包含用户的直接主界面，但它可以在包含一些模态或非模态的对话框，也就是说，在适当的时候，代码部件可以禁止所有用户的交互方式。代码部件为可复用代码提供了非常简单的方式。开发人员可以根据代码部件提供的类来创建各种对象，并调用该对象的属性、方法和响应该对象的事件。ActiveX 的代码部件在以前被称作 OLE 服务器，它同样也分为进程内和进程外两种。进程内实现对象的快速访问；进程外可实现不同的执行线程。

ActiveX 控件在以前的版本中被称为 OLE 控件。开发人员可以使用 ActiveX 控件(下简称控件)来实现自己的窗体和用户界面元素。在 Web 页面中可以使用 ActiveX 控件，从而大大增加了页面的功能。有了 ActiveX 控件，可使客户端的浏览器的功能更加强大。由此可见，进行 ActiveX 控件开发的意义不仅是用户可以定制自己的程序界面，而且可将这种投资转移到 Internet 之上，更为重要的是它构成了大规模软件构架的基石；同时为其它可使用 ActiveX 控件的工具提供可复用的模块，达到不同语言之间模块的相互调用。

ActiveX 控件不仅有代码部分，也有像窗体一样的可视部分。ActiveX 控件是 ActiveX 部件中应用最为广泛的一种，它也是本文讨论的重点。

2.2　基于 Web 的 ActiveX 控件的工作机理

ActiveX 是 Microsoft 为适应互联网而发展的标准。ActiveX 是建立在 OLE 标准之上，为扩展 Microsoft Web 浏览器 Internet Explorer 功能而提供的公共框架。ActiveX 控件是用于完成具体任务和信息通讯的软件模块。

ActiveX 控件是扩展 Web 浏览器的动态模块，ActiveX 能被支持 OLE 标准的任何程序语言或应用系统所使用。基于 ActiveX 控件的互联网应用系统是依赖 ActiveX 来完成对应用数据的处理的。ActiveX 控件与 Web 浏览器灵活无缝结合

*)基金项目：国家863重大专项基金(2003AA118105)资助。王映辉　博士，副教授，目前的主要研究方向为可视化技术与大规模软件工程。

维普资讯 http://www.cqvip.com

在一起。在通常情况下，ActiveX 控件包容在 HTML 代码中，并通过<OBJECT>参考标签来获取。当 Web 浏览器发出应用数据显示操作请求时，Web 服务器接受到用户的请求，进行处理，并将用户所要的数据和 ActiveX 控件传送给 Web 浏览器；之后，客户机端接受到 Web 服务器传来的数据和 ActiveX 控件，启动 ActiveX 控件，对相应数据进行处理，完成各种操作(图1)。

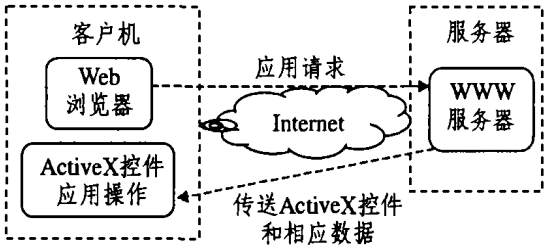

图1 基于 ActiveX 控件的 Web 应用体系结构

3 ActiveX 控件设计中的微观结构

控件包含三部分内容，即两个共有部分和一个私有部分。控件的外观和控件界面接口(属性、方法和事件)是共有的，这两部分用于用户交互和编程。控件工作的代码部分是私有的，其实现的效果可见，但代码隐藏。

许多的工具可以用来设计和实现 ActiveX 控件。由于用 VisualBASIC6.0(下简称 VB6.0)开发 ActiveX 控件具有简单、容易和快捷等优点，在此选取用之。通过 VB6.0 对 ActiveX 控件的设计和实现过程，来进一步透视 ActiveX 控件的框架结构、对外接口以及它的工作机理。

在控件的设计中，包括接口设计和外观设计两大部分内容。

3.1 ActiveX 控件的骨架构造与接口影射

ActiveX 控件最为简单的接口可由已有的一个或若干个系统提供的标准控件或用户定制的扩展控件构成，此时控件外观的设计与标准的 VB6.0程序用户界面的设计过程相同。而对 ActiveX 控件接口的设计可使用 VB6.0的“ActiveX 控件接口向导”来完成。

VB6.0的“ActiveX 控件接口向导”位于 VB6.0主菜单的“外挂程序”菜单中，这项功能可通过“外挂程序”的“外挂程序管理器”选取并双击之后来获得。在新建工程并选取了“ActiveX 控件”之后的操作界面与代码设计步骤与标准的程序设计一模一样，在此不再重述。

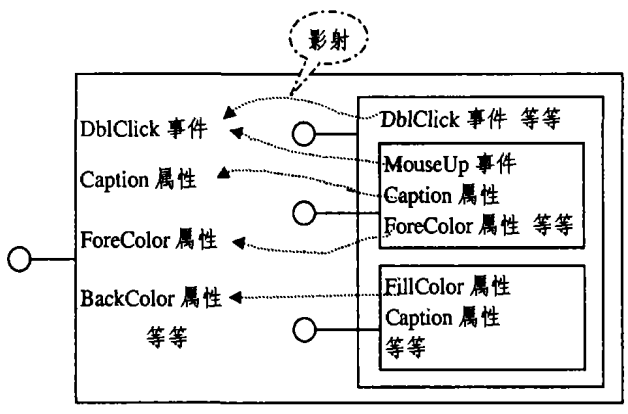

图2 ActiveX 控件接口与组成它的控件接口之间的影射

要通过“ActiveX 控件接口向导”来完成 ActiveX 控件接口的设计，其主要的工作是完成被设计的控件接口(包括系统提供的和自己定义的)与组成它的控件接口之间的影射(图2)。当启动向导并选择控件后，向导将检查组成控件，并产生一份属性、方法和事件的列表，其中包括了组成控件的所有接口和被设计控件对象中出现的属性、方法和事件列表，其中包括一些标准的属性。进而在列表中选取用于被设计控件所需的属性、方法和事件。当接口被委派之后，我们可以通过代码设计窗口来观察所自动形成的过程和代码。代码窗体中的内容由四部分组成：声明段部分、设置和读取属性值部分、报告事件部分和设置以及读取属性值事件部分(参见下节的进一步描述)。每一个属性用 Property Get 和 Property Set 过程来完成对它的读写任务；ActiveX 控件的方法就是一个标准的过程；而每一个事件对应一个包含 RaiseEvent(之后的事件名是在声明段中用 Event 声明过的)语句的事件响应过程。

此外，对 ActiveX 还要进行控件属性页接口的设计。当在“外挂程序”的“外挂程序管理器”中选取“属性页向导”之后，就可以利用此向导非常方便地完成 ActiveX 控件的属性页的设置和建造。

3.2 ActiveX 控件可视界面绘制模式

控件的可视部分是控件的外观界面，这种界面是通过系统图形的绘制来完成的；而控件的接口界面是它被用来开发新软件的使用界面，是通过该控件的属性、方法和事件来体现的。在 VB6.0环境下，对 ActiveX 控件外观界面的创建有三种模式：完全用户绘制、扩充一个已有的控件和用现有的多个控件提供外观。第一种模式完全由用户完成控件的绘制工作。如果是后两种模式，控件的组成控件会部分或全部地自动完成该控件的绘制工作，所以只需关心组成控件在该控件中的放置位置即可。

3.2.1 *完全用户绘制控件* 该模式下可对控件的外观进行完全自我控制，同时还需要知道何时绘制控件，它当时处于什么状态，以及是否对焦点矩形进行绘制等等。

当自己绘制控件的外观时，绘制代码只能编写在控件的 Paint 事件过程中。控件容器在重新绘制所在区域时，控件将接收 Paint 事件并完成相应的绘制工作。如果控件要根据用户的要求来改变外观，则调用控件的 Refresh 方法来引发 Paint 事件。当利用控件的内置绘图方法不能满足要求时，可以调用 Windows 的 API 函数来完成绘制任务。对于文本的绘制，可用 Print 方法。

对于获取焦点的控件，Windows API 的 DrawFocusRect 函数可以用于绘制类似于 CommandButton 这样的单像素点划线。对于非矩形没有提供 API 函数。

如果要在控件中显示文本，可以在 Paint 事件响应过程中调用 Print 方法，但此时显示的内容总是以省缺字体形式。要解决这一问题，可增加一个 Font 对象，并在 Font 对象的 FontChange 事件中调用控件的 Refresh 方法。并且用：

```
Option Explicit
Private WithEvents mFont as stdFont
```

对 Font 对象进行声明。

3.2.2 *扩充一个已有的控件* 对于由一个已有控件扩展控件外观的情况，已有控件肯定要占据被绘制控件的全部外观空间和可见画面。在被扩展控件的 Resize 事件响应过程中，可使用已有控件的 Move 方法来进行大小的调整。由此可见，被扩展控件的外观完全由已有的控件来决定，此时已有控件的属性和方法是影响扩展控件外观的主要因素。

维普资讯 http://www.cqvip.com

3.2.3 用现有的多个控件提供外观 对于通过多个控件的组合来决定外观的控件，在 Resize 事件响应过程中，要充分考虑对多个控件的位置和尺寸的调整。

由于在多个组成控件中，有些控件可能有它的最小尺寸，因此被制作的控件也有最小尺寸问题。企图要通过 Resize 事件将被制作控件的 Height 或 Width 的值进行调整，会产生 Resize 事件的第二次发生。这可以通过设置标志变量的方法来解决。

4 对 ActiveX 控件实现的进一步描述

基于 VB6.0中开发的控件实际上是一个控件类。当它被放入窗体(容器)中时就创建了该控件的实例。用户在设计时把控件放入窗体上就创建了一个真正的控件类实例，这个实例被称为设计时实例，当窗体关闭时，设计时实例就会被释放掉。如果工程处于运行状态，当窗体加载时就会创建一个运行时实例，窗体卸载后，运行时实例被释放掉。控件依赖于容器的结果是，某些特性不一定能在所有的容器中使用，此时这些特性是无效的。

属性、方法和事件构成了 ActiveX 控件的对外接口，在控件的设计和使用中扮演着十分重要的角色。下面分别给以进一步讨论。

4.1 ActiveX 控件与容器

由于控件总是放在容器上的对象，因此，容器所提供的部分额外的属性、方法和事件看上去是控件的一部分。另外，控件的许多属性需要容器的支持，某些特性不一定能够在所在的容器中使用，这些属性将是无效的。

对于容器所提供的属性、方法和事件可通过两种方式获得，一是使用扩展对象的 Parent 属性来访问，二是通过控件的 Ambient 属性来获取。

(1)容器的 Extender 对象

在控件中由容器提供的属性并不是我们所定制的，它们是控件的无缝扩展。多数容器会实现这些属性，如 Left、Top、Width、Height 等。

当控件实例的属性与 Extender 属性相同时，则 Extender 属性优先。例如，假设控件具有 Tag 属性，当把该控件放在窗体容器上时，Tag 属性将由窗体的 Extender 提供。也就是说，对于语句：控件名.Tag="TestExample"，其结果是将字符串"TestExample"存放在窗体 Extender 对象的 Tag 属性中。此时如果容器没有 Tag 属性，则上述代码将把字符串存入控件的 Tag 属性中。

(2)使用 AmbientProperties 对象

在 VB6.0中，控件对容器环境变量的访问是通过 AmbientProperties 对象进行的，而对 AmbientProperties 对象的引用是通过控件对象的 Ambient 属性返回的。对大多数的 Ambient 属性可以忽略，如 ScaleUnits、ShowHatching、UIDead 等；但有些属性则不能忽略，如表示控件处于设计模式还是运行模式的 UserMode 属性。

对于创建在运行时可读，但只能在设计时设置的属性，可在 Property Let 和 Property Set 过程中，通过检查 AmbientProperties 对象的 UserMode 属性分别加以处理即可。

对于控件的外观与容器的协调，主要有两种处理办法。一是在 InitProperties 事件中可以把控件的属性，如 ForeColor、BackColor、Font 和 TextAlign 等设置为 Ambient 相应的属性，控件的任何一个实例首次放置在容器中时都会接收到它。二是在 AmbientChanged 事件响应过程中加入使控件外观符合控件环境的代码，根据容器的变化调整控件的外观和行为。

4.2 ActiveX 控件实现中的属性

控件的属性主要在两个方面区别于 VB6.0的其它对象属性。一是在设计时属性显示在属性窗口和属性页对话框中；二是属性保存在容器的源文件中。由此可见，实现控件的属性比其它类型对象的属性具有更高的要求和选项。

在通过属性过程 Property Let 实现控件的属性赋值时，如果不调用 PropertyChanged 方法，VB6.0就不能把控件实例标记为需要保存的，因此它们将不会收到 WriteProperties 事件，在控件的使用中将会丢失在设计时设置的属性值。另外，控件的属性值可能在多个地方显示，因此，当属性发生变化时必须通知开发环境，以便它们能够同步显示在属性窗口、属性页对话框等位置上。对于非数据绑定的、且只在运行时可用的属性，不需要调用 PropertyChanged 方法，但仍需要用属性过程来实现。

对于通过控件容器的 Extender 对象提供的属性，并不需要做更多的工作。按照省缺规定，控件对象的属性和组成它的控件的属性对最终用户是不可见的，这样，控件的制作者可以充分自由地确定控件的接口。

要创建只在运行时可用的属性，在设计时只要将属性过程设置为无效即可，因为 VB6.0的属性窗口中不会对设计时无效的属性进行显示。

4.3 ActiveX 控件实现中的方法

除了以上介绍的利用 ActiveX 控件接口向导进行被设计控件的接口委派，进而完成对控件的方法设计之外，还可通过在代码窗口中直接加入 Public 类型的过程或函数的方式来完成对控件方法的设计。

如果控件在运行时是可见的，那么应提供 Refresh 方法。此方法对于用户绘制的控件将产生 Paint 事件；而对于已有控件组成的控件，Refresh 方法会强制组成控件进行刷新。

4.4 ActiveX 控件实现中的事件

属性和方法被认为是向内的，而事件是向外的。也就是说，属性和方法是由使用控件的开发者从控件的外面调用的，而事件是从控件的内部向外传播到开发者，以便能够执行事件过程中的代码。

众所周知，Windows 程序的执行过程全由事件驱动[1,6]。所以，对于 ActiveX 控件来说，正确地把握事件发生的前后顺序和时间，对我们开发 ActiveX 控件显得非常重要。大概包括如下的几个方面：

(1)控件使用人员将控件添加到窗体上时，发生如下事件：

Initialize：初始化控件；

InitProperty：给控件属性赋默认值。

(2)从设计切换到运行时，发生如下事件：

WriteProperties：将控件使用人员对控件属性的设置保存起来；

Terminate：终止控件设计时的实例；

Initialize：初始化控件新的运行实例；

ReadProperties：读取控件使用人员对控件属性值的设置。

(3)从运行切换到设计时，发生如下事件：

(下转第180页)

维普资讯 http://www.cqvip.com

$Trans_C(String)_2$ this_LC){…} …

·结构、类类型的变量定义及初始化例如:String x=new String;

变换为 Java: $Trans_{Java}$ (String) $Trans_{Java}$ (x) = new $Trans_{Java}$(String)( );

变换为 C:$Trans_C(String)_2$ $Trans_C(x)_1$=NULL;

{ $Trans_C(String)_2$ concat($Trans_C(x)_1$,"_1")=NULL;

concat($Trans_C(x)_1$,"_1")=($Trans_C(String)_2$)malloc (sizeof(struct $Trans_C(String)_1$));

$Trans_C(x)_1$= concat($Trans_C(x)_1$,"_1");}

3.4 数组

数组的定义、使用的语法语义与 Java 完全相同。例如:int [][] var=new int [10][5];

变换为 Java:int [][] $Trans_{Java}$(var)= new int [10][5];

变换为 C:int** $Trans_C(var)_1$=NULL;

{int len_LC_1 = 10, len_LC_2 = 5; int itr_LC_1, itr_LC_2;

int** concat ($Trans_C$ (var)$_1$,"_1") = (int**) malloc (sizeof(int*)*len_LC_1);

for (itr_LC_1 = 0; itr_LC_1 < len_LC_1; itr_LC_1 ++)

concat($Trans_C(var)_1$,"_1")[itr_LC_1]=(int*)malloc (sizeof(int)* len_LC_2);

$Trans_C(var)_1$=concat($Trans_C(var)_1$,"_1");}

3.5 ALL 程序

完整的 ALL 程序包括:include 声明、struct、class 定义、函数定义和程序入口点 main 函数。ALL 程序中没有全局或者静态变量,main 函数必须形如:void main(){…},这些对语言的简化也基于消除和算法与数据结构无关的语言成分的考虑。

main 函数变换为 Java:class $Trans_{Java}$(该函数定义所在的文件名){public static void main(String [] args){…} },变换为 C:void main(){…}

结束语 依据领域语言的思想设计的小语言 ALL 在用于算法与数据结构教学时有望达到概念清晰、描述简便、易于变换、间接可执行等特点。我们现在已经实现了 ALL 的语法、静态语义检查及其到 Java 的变换系统,建立了使用 ALL 进行算法和数据结构程序实习的环境,并用 ALL 语言完成了基本数据结构和算法的描述,实际检验了 ALL 语言的效果和变换系统的功能,为辅助教学打下了良好基础。但是要将 ALL 语言真正用于教学实践还需要为实习环境增加一些功能,如:变换结果的格式化输出、程序调试等,更需要的是一本使用 ALL 语言描述算法和数据结构的教材,从而充分发挥 ALL 的特点,实现第一节提出的课程教学目标。

参 考 文 献

1 Van D A, Klint P. Little Language: Little Maintenance. Journal of Software Maintenance, 1998, 10: 75~92

2 张乃孝,裘宗燕. 数据结构——C++与面向对象的途径(修订版). 北京: 高等教育出版社, 2001

3 许卓群,张乃孝,杨冬青,唐世渭. 数据结构. 北京: 高等教育出版社, 1987

4 张乃孝,郑红军,裘宗燕. 语言的抽象、封装与变换型开发方法. 软件学报, 1998, 9(7): 496~500

5 和华. Garment 定义语言方法及实例研究:[北京大学硕士学位论文]. 2000

(上接第166页)

Terminate:终止控件设运行时的实例;

Initialize:初始化控件新的设计实例;

ReadProperties:读取控件使用人员对控件属性值的设置。

(4)关闭工程时发生如下的事件:

Terminate:终止控件设计时的实例。

由上可知,控件的运行实例和设计实例是两个完全不同的实例。从运行切换到设计时,并没有发生 WriteProperties 事件将运行时的属性保存起来。这显然是合理的,因为设计时的属性值的改变对运行仍然有效;反之则不然。

为了保存控件使用人员对控件属性值的设置,VB6.0提供了一个特殊的类对象 PropertyBag。该对象有两个重要的方法:ReadProperty 和 WriteProperty,分别用于属性的读取和赋值。向导生成代码的最后部分是与设置和读取属性值有关的,主要包括 ReadProperties 和 WriteProperties 两个事件过程,在这两个过程中包含了 PropertyBag 对象对属性的读写语句。同时需要注意的是,可以对向导生成的这些内容进行相应的调整和修改。

我们知道,容器对控件的使用是非常重要的。容器的 Extender 对象提供了许多的事件,此时控件的制作者不需要作任何工作即可获得 GotFocus、LostFocus、DragOver 和 DragDrop 四个事件。

可以为控件设置缺省事件。在控件设计时,调用 VB6.0 "工具"菜单的"过程属性"菜单,在对话框中选取相应的属性,并在"高级"按钮单击后选取"省缺用户界面"即可。此时,在使用控件的代码窗口中当选取此控件对象之后,会自动形成该控件对象的省缺事件的过程框架。

最后的工作是将 ActiveX 控件编译成扩展名为.ocx 的文件,此时会在本机上自动注册。当然,在其它的机器上使用时要重新注册。

结束语 ActiveX 能被支持 OLE 标准的任何程序语言或应用系统所使用,比 Plug-in 模式更灵活、更方便。ActiveX 控件实现了软件模块最大程度的复用。

另外,对于 Web 应用来说,与传统应用的最大不同是客户端的程序主要在常用的浏览器中运行,而 ActiveX 控件在页面中的使用实现了浏览器无法实现的功能,许多任务不必再求助服务器,从而大大减小了网络和服务器的负担。ActiveX 是以构件的形式出现的,可以灵活配置,而且还可以回调服务器上其它构件的方法,在很大程度上增强了应用的兼容性、开放性和安全性。

最后要说明的是,由于 ActiveX 控件是一种构件模块,因此为基于软件总线机制的大规模软件构架技术的实现奠定了基础。

参 考 文 献

1 潘爱民,等. COM 原理与应用[M]. 北京:清华大学出版社, 2000

2 王栋. Visual BASIC 程序设计实用教程[M]. 北京:清华大学出版社, 2002.3

3 张树兵,等. Visual BASIC6.0入门与提高[M]. 北京:清华大学出版社, 2001.1

4 谭淑英,李赫男,左贵启. 服务器端的动态网站开发技术[J]. 计算机应用研究, 2002, 5: 143~149

5 陈章渊等译. 智能 CORBA [M]. 北京:电子工业出版社, 1999

6 钱力棚,等. Visual InterDev6.0网络编程技术[M]. 北京:人民邮电出版社, 2000.1

7 王映辉,冯德明. 大规模软件构架技术[M]. 北京:科学出版社, 2003.6