

# Excel 的 InputBox 方法中有缺陷部分的替代解决方案

孔令旗<sup>1</sup> 吴允波<sup>2</sup> 张琛<sup>1</sup>

<sup>1</sup>(焦作师范高等专科学校计算机与信息工程系 河南 焦作 454000)

<sup>2</sup>(焦作师范高等专科学校现代教育技术中心 河南 焦作 454000)

**摘要** 运用 Excel 2003 中自定义函数、用户窗体与事件处理的“并发”运行和数据传递技术,设计一个 VBA 宏程序,它满足了在任何条件下都能返回用户选定区域引用的 VBA 编程需求。设计该宏的目的是为了替代 Excel 2003 Application. InputBox 方法中有缺陷的部分。程序亦能运行在提供 VBA 环境、但缺少交互式选择区域方法的 WPS 2012 专业增强版上。

**关键词** Excel 软件缺陷 VBA 方法 事件处理 区域 宏

中图分类号 TP391.13 文献标识码 A DOI:10.3969/j.issn.1000-386x.2013.03.084

## AN ALTERNATIVE SOLUTION FOR DEFECTIVE PART OF INPUTBOX METHOD IN EXCEL

Kong Lingqi<sup>1</sup> Wu Yunbo<sup>2</sup> Zhang Chen<sup>1</sup>

<sup>1</sup>(Department of Computer and Information Engineering, Jiaozuo Teachers College, Jiaozuo 454000, Henan, China)

<sup>2</sup>(Modern Education Technology Center, Jiaozuo Teachers College, Jiaozuo 454000, Henan, China)

**Abstract** Using the techniques of “concurrent” execution of user-defined functions, UserForms and events handling as well as data transfer in Excel 2003, we design a VBA (Visual Basic for Applications) macro, which meets the programming requirement of VBA that it can return to citation of user-selected range in any condition. The purpose of designing the macro is to replace the defective part in Application InputBox method of Excel 2003. And it can also run on WPS 2012 professional plus version which providing VBA environment but lacking a method for interactive selection of a range.

**Keywords** Excel Software defect VBA Method Events handling Range Macro

## 0 引言

Excel 是微软办公软件 Microsoft Office 的组件之一,它可以进行各种数据的处理、统计分析和辅助决策操作,广泛应用于管理、统计、财经、金融等领域。尽管 Excel 已从 2003 先后升级到 2007 和 2010,但由于用户使用习惯、软件购置成本,以及信息技术应用教育内容一般相对滞后等因素,Excel 2003 还有相当广泛的用户群。

1994 年发行的 Excel 5.0 版本即具备了 VBA 的宏功能。VBA 基于 Visual Basic for Windows 发展而来,它提供了面向对象的程序设计机制和一种完整的程序设计语言。VBA 可以使用宏记录器记录用户的各种操作,将其转换为 VBA 程序,易于学习掌握。VBA 程序的开发与应用有助于数据处理工作的自动化。

区域(Range)是 Excel 操作中使用最为频繁的对象之一。进行 VBA 程序设计开发时,“先获取用户选定区域,再对区域内对象进行处理”的流程模式,贯彻了“以用户为中心设计和评估”的基本原则<sup>[1]</sup>。Excel 中 Application 对象的 InputBox 方

法<sup>[2]</sup>为程序设计中人机交互功能的实现提供了语言级支持。当 InputBox 方法的 Type 参数取值 8 时,可让用户用鼠标拖动或键盘输入方式确定选定区域,并返回该区域的引用。遗憾的是在 Type 参数取值 8 时,InputBox 方法存在软件缺陷<sup>[3]</sup>。经过反复试验发现,当选择的区域符合下列条件之一时:

- 1) 区域中的单元格有逻辑运算公式的条件格式;
- 2) 区域毗邻的单元格有逻辑运算公式的条件格式。

InputBox 方法极有可能出现程序运行错误。尽管该软件缺陷在 2003 以后的版本中已得到了修复,但它还是困扰着 Excel 2003 的 VBA 开发应用人员。

使用扩展了 VBA 功能的 WPS 2012 专业增强版时,发现它不支持 Application. InputBox 方法,即 WPS 2012 缺少了在 VBA 环境下采用交互方式选定区域的语言级功能,这也在一定程度上影响了国产优秀软件 WPS 的深度应用。

针对上述问题,本文给出了一个替代 Application. InputBox 方法中有软件缺陷部分的 VBA 编程解决方案。

收稿日期:2012-04-24。孔令旗,副教授,主研领域:数据库,数据挖掘。吴允波,讲师。张琛,讲师。

## 1 技术方案

### 1.1 Excel 对象模型简述

VBA 环境下, Excel 是由不同类型的对象构成的应用程序体, 同一类型的对象组成一个对象集合, 不同类型的对象、对象集合又按层次结构进行组织。每一种类型的对象都拥有自己的属性与方法。当对象能主动响应外部发生的随机变化时, 该对象类型还拥有事件。Excel 内建对象的方法代码由 Excel 本身提供, VBA 程序只能调用; 而事件处理代码则由有事件捕获需要的用户自行设计, 事件处理代码的程序框架规范由 Excel 定义。

Excel 的 Workbook 对象代表打开的一个工作簿, 它是 Workbooks 集合的成员。Workbooks 集合包含当前打开的所有工作簿的 Workbook 对象实例。SheetSelectionChange 事件是 Workbook 对象众多事件中的一个<sup>[2]</sup>。当任一工作表上的选定区域发生变化时, 都将触发该事件。Excel 的下级对象定义的事件有时在上级对象中也有定义。例如 Application 是 Workbook 的上级对象, 其中也定义了和 Workbook 一样的 SheetSelectionChange 事件。下面的示例用 Workbook 的 SheetSelectionChange 事件处理代码, 在状态栏上显示选定区域所在的工作表名称及选定区域地址。

```
Private Sub Workbook_SheetSelectionChange(
    ByVal Sh As Object, ByVal Target As Excel.Range)
    Application.StatusBar = Sh.Name & ":" & Target.Address
End Sub
```

在 Excel 对象模型中, Application 位于所有对象的顶部, 代表启动的 Excel 应用。Excel 运行期间, Workbooks 对象实例的多寡依赖于打开工作簿的个数, 但 Application 对象实例有且仅有一个。

### 1.2 方案目标与思路

本方案的目标是设计一个用户自定义函数 UDF (User Defined Function), 它能接受在工作簿的任一个工作表中用鼠标拖动或键盘输入确定的一个或多个区域, 并返回这些区域所对应的一个 Range 对象的引用。程序既能在 Excel 2003 及后续版本上运行, 也能在扩展了 VBA 后的 WPS 2012 上运行。

解决问题的基本思路是运用 UDF、UserForm (用户窗体)、Workbook 对象的 SheetSelectionChange 事件处理的“并发”执行和数据传递技术, 实现交互式选定区域的功能。细化的逻辑设计如下:

设计一个 UserForm, UserForm 中的文本框控件接受用户输入或鼠标选择的区域地址; 用户用鼠标选择区域时, 通过 Workbook 对象的 SheetSelectionChange 事件处理, 把用户选定区域的地址置入 UserForm 中的文本框。

由于 UserForm 运行时不能直接向 UDF 传递数据, 需要用 Global (全局) 变量作为数据交换缓冲, 实现它们之间的数据交换。当 UserForm 关闭时, UserForm 的退出事件处理将地址指向的 Range 对象引用写入 Global 变量; UDF 检测到 UserForm 关闭后, 从 Global 变量中获取 Range 引用, 再予以返回。

方案中 UDF、UserForm、Workbook 及用户四者间的信息流和控制流参见图 1。

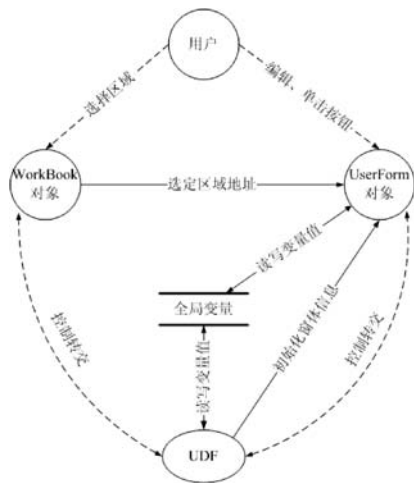


图 1 程序模块间的信息与控制流  
实线表示数据流; 虚线表示控制流

## 2 程序设计

Excel 的对象模型与 WPS ET (电子表格) 的对象模型存在差异。为了保证所设计的代码在两个系统下都能正常运行, 代码设计时选择二者都支持的对象、属性、方法和事件。

### 2.1 Global 变量设计

在 VBAProject 的模块夹中创建“模块 1”, 在其内设计两个 Global 变量:

```
Global rngSelected As Range
Global blnFormRunning As Boolean
```

Range 型变量 rngSelected 存储选定区域的引用; Boolean 型变量 blnFormRunning 供 UDF 检测 UserForm 是否关闭 (判断用户是否完成区域选定)。UDF 启动 UserForm 时将其置为 True, UserForm 关闭时将该变量置为 False。

### 2.2 UserForm 设计

UserForm 的名称为 InputRangeForm, 在 VBAProject 的“用户窗体”夹中创建。窗体内有提示信息标签控件 PromptLabel, 文本框控件 AddressTextBox, 确定按钮 ConfirmButton, 取消按钮 CancelButton。ConfirmButton 按钮的 Click 事件代码设计如下:

```
Private Sub ConfirmButton_Click()
    If AddressTextBox.Text = "" Then
        MsgBox "你未选择区域!"
    Else ' 文本框有非空字符串
        ' 捕获文本框键盘输入区域地址错误
        On Error GoTo ErrorHandler
        Set rngSelected = Range(AddressTextBox.Text)
        ' 改变状态变量
        blnFormRunning = False
        ' 卸载 UserForm
        Unload Me
        Exit Sub
    ErrorHandler:
        MsgBox "输入的区域地址有错误!"
    End If
End Sub
```

CancelButton 按钮的 Click 事件处理完成的操作有: 置变量 rngSelected 为 Nothing, 变量 blnFormRunning 为 False, 卸载

UserForm。

当 UserForm 通过标题栏的关闭按钮关闭时,需要对 UserForm 的 QueryClose 事件处理进行编码,当 UserForm 的 CloseMode 属性不等于 1 时,设置 rngSelected 为 Nothing, blnFormRunning 为 False。代码如下:

```
Private Sub UserForm_QueryClose(Cancel As Integer, _
    CloseMode As Integer)
    if CloseMode < > 1 Then
        Set rngSelected = Nothing
        blnFormRunning = False
    End If
End Sub
```

UserForm 的 Initialize 事件处理将对 AddressTextBox 文本框控件的 Text 属性进行初始化,其值为 Application. Selection. Address。代码如下:

```
Private Sub UserForm_Initialize()
    AddressTextBox.Text = Application.Selection.Address
End Sub
```

## 2.3 SheetSelectionChange 事件代码设计

Workbook 对象的 SheetSelectionChange 事件处理代码比较简单,处理逻辑为当 InputRangeForm 窗体对象存在时,进行数据的推送。即将事件处理过程参数 Target 所携带区域引用的地址值置入 InputRangeForm 中 AddressTextBox 控件的 Text 属性。事件处理代码如下:

```
Private Sub Workbook_SheetSelectionChange(_
    ByVal Sh As Object, ByVal Target As Range)
    If Not InputRangeForm is Nothing Then
        InputRangeForm.AddressTextBox.Text = _
            Target.Address
    End If
End Sub
```

此处的代码放在“ThisWorkBook”夹中。

## 2.4 UDF 设计

设计的 UDF 函数名为 InputRange,它有两个可选参数,其值分别作为 UserForm 的标题和提示信息。函数返回值类型为 Range 引用。

由图 1 可以看出,本方案须使 UDF、UserForm 和 Workbook 事件处理等三个模块“并发”运行。在程序设计初期,本以为 UserForm 无模式显示(目的是让用户能对工作表实施操作)后,UDF 等待 UserForm 关闭,同时用户应该可以对 UserForm 和工作表进行操作。但真实情况是 UDF 进入了查询循环后,独占了临界资源——CPU 资源和变量 blnFormRunning 资源,使该变量的值永远保持为 True,发生了死锁<sup>[4]</sup>。

解决死锁的关键是在 UDF 查询循环后,能让 UserForm 和事件处理过程能获得 CPU 资源的使用权。查阅了 VBA 有关资料后发现,VBA 函数库中提供了一个 DoEvents 函数。该函数执行时将控制传递给操作系统,操作系统处理完事件队列中所有事件和 SendKeys 队列的所有键后,将控制返回到调用 DoEvents 的进程中<sup>[5]</sup>。于是在 UDF 的状态查询的循环体内,加入 DoEvents 库函数的调用,使系统有机会处理事件和键盘输入,进而使 blnFormRunning 有被改变的机会,死锁问题迎刃而解。

下面是 UDF 代码。程序中已进行了注释,此处不再说明。这里的代码存放在“模块 1”内。

```
Function InputRange(Optional ByVal Title As String, _
```

```
Optional ByVal Prompt As String) As Range
```

```
    ' 以无模式方式显示用户窗体
```

```
    InputRangeForm.Show 0
```

```
    ' 设置 UserForm 的标题
```

```
    If Title = "" Then
```

```
        InputRangeForm.Caption = "选择区域"
```

```
    Else
```

```
        InputRangeForm.Caption = Title
```

```
    End If
```

```
    ' 设置 UserForm 的信息提示
```

```
    If Prompt = "" Then
```

```
        InputRangeForm.PromptLabel.Caption = _
```

```
            "请选择一个或多个区域"
```

```
    Else
```

```
        InputRangeForm.PromptLabel.Caption = Prompt
```

```
    End If
```

```
    blnFormRunning = True
```

```
    ' 循环测试,等待 UserForm 关闭
```

```
    ' 使事件处理代码有获得 CPU 资源的机会
```

```
    Do Until Not blnFormRunning
```

```
        DoEvents
```

```
    Loop
```

```
    ' 函数返回值置为用户选定区域的引用
```

```
    Set InputRange = rngSelected
```

```
    End if
```

```
End Function
```

## 2.5 设计附注

前面提及的“并发”运行的实现并非真正意义上的多进程或多线程并发运行,而是借助于事件随机触发性和 UDF 主动出让 CPU 资源占有权来实现的。宏观上就象 UDF、UserForm 和 Workbook 的事件处理在并发运行一样。

上述程序代码在 Excel 和 WPS ET 均可正常运行。实验时发现 Excel 和 WPS ET 在程序运行外观上还是存在细微的差异。对于 Workbook 的 SheetSelectionChange 事件,在 Excel 运行上可理解为:“工作表选定区域改变后”;而 WPS ET 可理解为:“当工作表选定区域改变时”。

## 3 宏的封装与使用

### 3.1 加载宏概述

一个 VBA 加载宏是一个 Excel 启动时可以自动加载的 .xla 文件。一般而言,加载宏包含了一些 Excel 本身不具备的功能函数。创建加载宏是分发用户定义函数、提高软件重用性的一种有效途径。

加载宏创建初期,它是一个工作簿(.xls)文件,用 Visual Basic 可在其中创建 UDF,并进行调试。加载宏中的函数如果都不涉及对象的事件处理,只需进行一些简单处理后将工作簿另存为 .xla 类型文件,就把工作簿文件封装成一个加载宏文件;如若不然,在封装前需要设计程序代码,将扩展了事件处理的对象与相应的对象实例关联起来<sup>[6]</sup>。

VBA 提供了有 WithEvents 选项的声明语句,用该选项声明的对象拥有了对象类型的事件处理机能。可以套用 VBA 事件处理过程的框架来设计事件处理代码。有了事件处理代码的对象还是一个“虚对象”,它的事件处理过程代码不会受外界影响

而触发。虚对象和相同类型的对象实例建立关联后,虚对象的程序代码被“嫁接”到对象实例上,对象实例就拥有虚对象定义的事件处理功能。对象关联需要用“Set 虚对象 = 对象实例”语句来实现。

### 3.2 封装策略与代码实现

将本程序封装为加载宏的要求是只要 Excel 完成本宏的加载,Excel 运行期间所有打开、新建或激活的工作簿都可以调用 InputRange 函数。

2.3 节中的代码扩展的是 Workbook 对象的事件处理,如果定义 Workbook 为虚对象,就意味着打开、新建或激活的工作簿事件发生时,都要进行对象关联操作,处理代码略显繁琐。由于 Application 对象拥有和 Workbook 一样的 SheetSelectionChange 事件,而 Excel 启动后仅有一个 Application 对象实例,事件处理编码就会变得非常简洁。所以这里定义一个 Application 对象类型的变量 app。app 对象的 app\_SheetSelectionChange 事件处理将完成鼠标选定区域地址的推送。事件处理代码设计如下:

```
' 声明具有事件处理能力的对象 app
Dim WithEvents app As Application
' app 对象的 SheetSelectionChange 事件处理过程
Private Sub app_SheetSelectionChange(_
ByVal Sh As Object, ByVal Target As Range)
    If Not InputRangeForm Is Nothing Then
        InputRangeForm.TextBox1.Text = Target.Address
    End If
End Sub
```

上面的程序代码还未实现 app 虚对象与已经存在的对象实例 Application 的关联。

从 Excel 2000 开始,它有一个内置的 Workbook\_Open 事件,当工作簿文件打开时该事件仅会触发一次<sup>[7]</sup>。加载宏可认为是一个特殊的工作簿文件,它被加载时 Workbook\_Open 事件也会触发一次。可在 Workbook\_Open 事件处理中进行一些初始化工作。对于本程序而言,初始化工作就是建立 app 虚对象与 Application 对象实例的关联。Workbook\_Open 事件处理代码如下:

```
Private Sub Workbook_Open()
    Set app = Application
End Sub
```

本小节的代码放在“ThisWorkbook”夹中。

### 3.3 加载宏的创建与使用

创建加载宏的过程是,启动 Excel 创建一个空工作簿文件,单击工具栏中的 Visual Basic 按钮启动 VBE (Visual Basic Editor) 窗口,在 VBE 中完成 UserForm 的界面设计和上述代码的录入;在 VBE 的工程管理器窗口右击顶级的工程名称 (VBA-Project),弹出快捷菜单,再点击“VBAPProject 属性”菜单项后弹出属性对话框,在“通用”选项卡中的工程名称框中输入“交互式选定区域宏”,然后关闭属性对话框;点击工具栏上的保存按钮,关闭 VBE 窗口;将该工作簿另存为 SelectRange.xla 文件。Excel 会将该加载宏放在约定的文件夹 C:\Documents and Settings\\Application Data\Microsoft\AddIns 中。至此就完成了加载宏的创建。

宏的加载与卸载通过 Excel 工具栏的“工具”按钮→“加载宏...”菜单项弹出的“加载宏”对话框完成。如果要加载的宏在对话框左边的列表框中未出现,点击浏览按钮找到要加载的宏文件,文件名就会出现在列表框中。改变列表框中加载项前的

复选框状态,再点击确定按钮来实现宏的加载或卸载。

由于本加载宏内的自定义函数 InputRange 返回 Range 对象的引用,故只适合在 VBA 编程中调用。调用的语法格式为:

```
Set rangeObject = _
    Application.Run("SelectRange.xla! InputRange", _
    TitlePara, PromptPara)
```

执行上述语句后,如果变量 rangeObject is Nothing 为 True,表明用户没有选定区域;否则 rangeObject 为选定区域 Range 对象的引用。下面是 Sheet1 的命令按钮的 Click 事件代码。放在“Sheet1”夹中:

```
Private Sub CommandButton1_Click()
    Dim r As Range
    Set r = InputRange()
    If Not r Is Nothing Then
        MsgBox "选择的区域为:" & Replace(r.Address, _
"$", "")
    Else
        MsgBox "未选中区域"
    End If
End Sub
```

## 4 结 语

本文通过一个具体程序的设计,给出了在 VBA 中实现 UDF、UserForm 和事件处理如何“并发”运行,以及它们之间如何进行数据交换的技术方案。经验也表明,在进行 Excel 或 WPS 的 VBA 应用开发时,若 UDF 运行的时间较长,但又不希望它的运行影响用户与系统的交互,编码中于适当时机调用 DoEvents 函数不失为一种好的选择。

本程序在 Excel 2003、2007 和 2010 三个版本都进行了测试,均能正常运行。该程序完全可以替代 Excel 2003 的 Application.InputBox 方法中 Type 参数为 8 时的功能。本程序同时也使原本不支持 Application.InputBox 方法的 WPS 2012 专业增强版具有了交互式选定区域的 VBA 功能。因为 WPS 2012 还没有加载宏的功能,所以 WPS 的 VBA 程序设计人员可以使用第 2 节的程序代码。

## 参 考 文 献

- [1] 董建明,傅利民,饶培伦. 人机交互:以用户为中心的设计和评估[M]. 2 版. 北京:清华大学出版社,2007.
- [2] Microsoft. Microsoft Excel 2003 language reference[EB/OL]. [2012-2-10]. <http://www.microsoft.com/download/en/details.aspx?displaylang=en&id=7273>.
- [3] Patton R. 软件测试[M]. 周子滨,姚静,等译. 北京:机械工业出版社,2002.
- [4] 汤小丹,梁红兵,哲凤屏,等. 计算机操作系统[M]. 3 版. 西安:西安电子科技大学出版社,2007.
- [5] Microsoft. DoEvents function[EB/OL]. [2012-02-12]. <http://office.microsoft.com/en-us/access-help/doevents-function-HA001228827.aspx>.
- [6] Pearson C. Application events[EB/OL]. [2012-02-12]. <http://www.cpearson.com/excel/appevent.aspx>.
- [7] Microsoft. How to make a macro that automatically starts when you open a workbook in Excel 2000[EB/OL]. [2012-02-12]. <http://support.microsoft.com/kb/265113/en-us>.