

T

hinking on GUI

Programming Based on Matlab

Matlab 图形用户界面编程的几点思考



刘伟 刘光斌 (西安第二炮兵工程学院 710025) 吴红 (北京第二炮兵第二研究所 100080)

摘要: Matlab作为一种科学计算语言,得到了广泛的应用。从Matlab 6.0以来,其图形界面的编程功能得到了增强,本文着重说明了其图形界面编程的基本思想,程序运行的流程以及不同界面的数据的相互传输问题,并给出了一个实例来说明其应用。

关键词: Matlab 图形界面 句柄

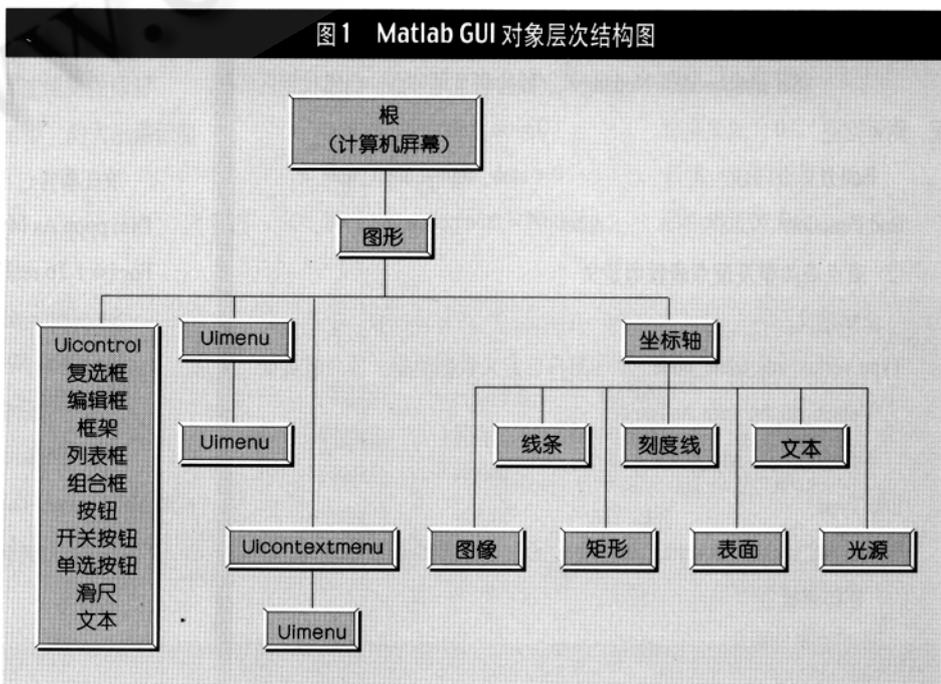
1. 引言

Matlab是Mathworks公司于1984推出的一套高性能的数值计算和可视化软件,它作为新兴的编程语言和可视化工具,有着其他编程语言(如C语言、Fortran语言等)所不能比拟的优势。如Fortran语言,是一种专用于科学计算的语言,但其图形界面的功能较弱,利用其所开发的程序,用户界面不友好,使用起来不方便;而VB、VC等可视化编程语言开发出来的程序界面友好,但由于其不是专用于科学计算的语言,因此其科学计算的功能又较弱,这对

于需要大量进行科学计算的用户来说是不可忍受的。而Matlab同时具备这两个方面的优势,既能进行科学计算,又能开发出所需要的图形界面,特别是Mathworks公司2001年推出的Matlab 6版本加强了其图形界面的编程功能。

图形用户界面(Graphic User Interface),简称为GUI,是指这样一种界面,它包含了窗口、菜单、对话框、按钮和文本等各种控件的用户界面,用户通过它与计算机进行信息交互。比如用户可以通过某种方式来选择或者激活这些对象,而这些对象将会响应用户的动作,并作出相应的反应,最常见的激活方式是利用鼠标来控制指针的移动以及通过鼠标按键来发出一个或者选中对象其他动作的信号。

图1 Matlab GUI对象层次结构图



针对如何利用 Matlab 来进行图形界面编程, 有许多文献以及 Matlab 的帮助文件都作了论述和探讨。但都只是针对如何应用 Matlab 所给出的图形界面工具来设计程序的图形界面, 很少涉及到 GUI 对象层次的论述和图形界面程序的运行流程, 另外在一些程序设计的过程当中, 还会遇到不同界面之间的数据传输问题。本文就从上述几个方面对 Matlab 图形界面编程进行了探讨, 并给出了一个实例。

2 GUI 对象层次结构

可以这样说, Matlab 图形界面程序的核心就是句柄图形的运用。对句柄图形的充分了解, 将使得 Matlab 图形界面程序的编写变得容易。句柄图形是一组底层图形函数的名称, 这些函数用来在 Matlab 中生成图形, 它提供了对图形的高级控制。其基本思想为: Matlab 的每一个可视部分就是一个对象, 每个对象都有一个相应的唯一标识符, 即句柄。通过对句柄图形的操作, 就可以对该句柄图形所对应的对象进行控制, 比如修改其属性, 调用其回调过程等等。Matlab 的图形对象包括 `uimenu`, `uicontrol` 和 `uicontextmenu` 对象以及图形, 坐标轴及其子对象。其对象层次结构如图 1 所示。

根对象可以包含一个或多个图形, 而每个图形可以包含一个或多个坐标轴及其子对象。每个图形还可以包含一个或多个独立于坐标轴的 `uimenu`, `uicontrol` 和 `uicontextmenu`。其中 `uicontrol` 没有子对象, 但其有 11 种类型。而 `uimenu` 和 `uicontextmenu` 对象通常以 `uimenu` 对象为子对象。

3 Matlab 图形界面程序运行流程

Matlab 软件给出了设计图形界面的工具, 通过这些工具, 可以很方便的修改或者调整图形界面元素的各种属性, 比如对象设计编辑器, 菜单编辑器, 对象属性编辑器, 位置调整工具和对象浏览器。但是对有过 Matlab 图形界面编程经验的设计者来说, 它们在一定程度上简化了编程的步骤, 可是其提供的功能也相对有限, 这一点上不能和可视化编程语言相比, 特别是对图形界面程序的运行流程陈述的不够, 比如图形界面的初始化, 图形界面程序输入参数和输出参数的传递, 而这对于两个图形界面程序之间的数据的相互传递是十分重要的, 因此很有必要弄清楚其运行流程。只有这样, 才能对图形界面程序编程有个全局的概念, 也有助于编写出高效的代码。

总的来说, 图形界面程序是基于消息驱动的, 在这一点上 Matlab 也是一样, 但它也有与其他可视化编程语言不同的地方。这也决定了其运行流程的不同, 见图 2。

(1) 初始化图形界面。这一过程是通过函数 `Openfig` 实现的。`Openfig` 函数调用与 M 文件对应的 `fig` 文件来初始化图形界面。在这一过程中, 还存在隐含调用 `*.fig` 的 `CreatFcn` 函数。但这一过程无法使用输入参数, 也就是说, 要用输入参数设置图形界面元素的一些特性, 还必须编写自己的初始化函数。

(2) 创建句柄结构来存储该图形界面所有对象的句柄, 用于回调函数以及自己编写的函数。这一过程是通过函数 `guihandles` 和 `guidata` 来实现的。只有获得了图形界面所有对象的句柄, 才能有效的进行编程, 因为 Matlab 图形界面程序的基础是句柄的应用, 没有了句柄, 也就谈不上 Matlab 图形界面程序的编程。

(3) 在后台建立起消息驱动机制, 等待用户进行操作并作出相应的响应。

(4) 初始化完毕, 给出输出参数。

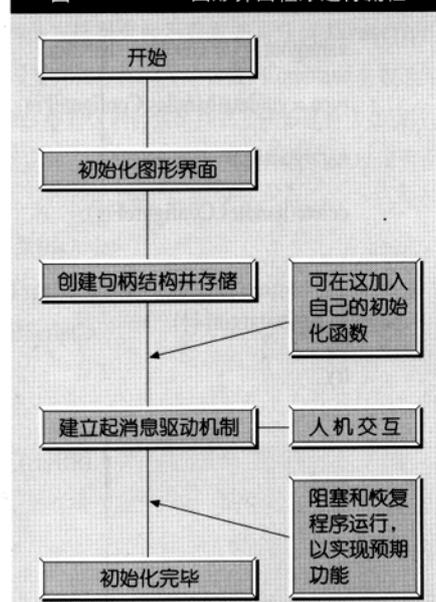
从上面的过程来看, 输入参数没有办法影响到图形界面元素的一些属性设置, 用户所作出的操作没有影响到输出参数的变化。这对于需要根据输入参数设置图形界面以及根据用户所作出的操作来设置输出参数的情形是非常难于实现的, 比如选项对话框的编制。但是通过运行流程的叙述, 就会发现, 如果能在第 2、3 步之间添加自己的初始化函数, 就可以利用输入参数来设置图形界面元素的一些属性, 而在第 3、4 步之间阻塞该程序运行, 但不阻塞消息驱动机制, 从而还能响应用户所进行的操作, 在用户操作完毕后, 恢复该程序的运行, 给出相关的输出参数。

4 实例

在第 2、3 节中, 我们叙述了 Matlab 的 GUI 对象层次结构以及 Matlab 图形界面程序的运行流程, 在这一节中, 将给出一个实例以阐述这些思想的具体实现过程, 并给出了如何实现两个图形界面程序间的数据传输的实现代码。

实例的源代码如下, 其功能是: 利用输入参数设置选项对话框的各个参数, 并能将用户所作出的新的选择构成输出参数并返回到其调用函数。

图 2 Matlab 图形界面程序运行流程



```
function varargout = Configure(varargin)
    if isstruct(varargin{1}) % 装载 GUI
        % 初始化图形界面。
        fig = openfig(mfilename,'reuse');
        % 用系统默认颜色设置图形界面的颜色。
        set(fig,'Color',get(0,'defaultUicontrol
Backgroundcolor'));
        % 创建句柄结构并存储该图形界面所有
对象的句柄,用于回调函数以及自己编写的函数。
        handles = guihandles(fig);
        handles.answer = 'DefaultAnswer';
        guidata(fig, handles);
        % 自己编写的初始化函数,其中用到
了句柄结构。
        SetValue(handles, varargin{1});
        % 阻塞该函数的运行,并等待恢复标
志为有效,以便继续运行该函数。由于这时消
息驱动机制已经建立,并在后台运行,因此并
不影响用户与计算机所进行的一切交互行为。
        set(handles.ConfigureFig, 'UserData', 'Open');
        waitfor(handles.ConfigureFig, 'UserData',
'Close');
        % 根据用户的操作来设置输出参数。
        if nargin > 0
            varargout{1} = GetValue(handles);
            type = guidata(handles.ConfigureFig);
            varargout{2} = type.answer;
            delete(handles.ConfigureFig);
        end
        elseif ischar(varargin{1})
            try
                ...
            catch
                ...
            end
    end
end
```

```
end
而调用该参数设置对话框的程序代码为
function varargout = Configure_Callback(h, eventdata, handles, varargin)
Data = get(handles.TFRFigure, 'UserData');
InputVal = Data.param;
[OutputVal AnswerType] = configure(InputVal);
...
end
其运行时的界面见图 3。
```

5 小结

在叙述 Matlab 图形界面程序编程的基本思想和程序运行的流程的基础上,通过一个实例,给出了它们实现的具体方法。在充分理解 Matlab 图形界面程序运行流程的基础上,文中给出了如何具体实现不同界面的数据的相互传输的源代码,希望能对大家在进行 Matlab 图形界面程序的编写过程中有所帮助。

参考文献

- 1 D.Hanselman, B.Littlefield 著, 张航、黄攀译, 精通 Matlab 6, 清华大学出版社, 2002 年。
- 2 闻新等编著, Matlab 科学图形构建基础与应用(6.x), 科学出版社, 2002。
- 3 胡昌华等编著, 基于 Matlab 的系统分析与设计-时频分析, 西安电子科技大学出版社, 2002。

图 3 参数设置界面

