维普资讯 http://www.cqvip.com

计算机科学1997Vol.21№.4

一种异种平台间通用图形界面编程接口

An Universal API for Graphic Programming on Hetero-platforms

蔡学武　尤晋元

（上海交通大学计算机系　上海200030）

摘　要　This article provides a technique based on an OO-method which can solve the reusability of the programs on hetero-platform without transplantation. We implement this method by constructing respective class libraries and take out the analogous graphic elements on different platforms. This technique supplies an universal API for graphic programming and regards the expansibility of system which make furthur development easier

关键词　Object-Oriented, Graphic, Hetero-platform, API

今天，用户的程序开发工作，很大程度上是在进行界面设计，其工作量往往占到整个开发工作的60%—70%。目前开发图形界面决不仅限于Windows，在小型机，工作站，甚至某些高档微机上，UNIX仍占据相当优势。在UNIX上，其图形系统称为X协议，主要的开发工具(ToolKit)有两种：OSF发布的Motif和以Sun的Open View为代表的开发工具。这两种不同的ToolKit，无论是在编程风格还是在界面风格上都有很大的不同。随着网络时代的到来，UNIX平台在网络服务器及工作站上有着举足轻重的作用，许多软件(如Netscape Navigator)都有许多不同平台上的版本。用户希望针对同一功能的软件开发出不同平台上的版本，使界面与功能都保持一致。然而用户要在不同平台上开发，首先要对这些平台的编程风格与编程技巧有相当的了解。有些编程工具(如Motif)，本身就是一个庞大的系统，这便给用户的软件移植带来许多困难。因此，构造异种平台间通用的图形界面编程接口更具有实际的意义。为此，我们专门设计了一套能够在W(Windows)与X(X Motif)上通用的编程接口，称之为wxWindow。用户利用系统所提供的编程接口(API)编制图形界面程序，在两种不同的平台下编译，便会得到相似的界面，可以避免移植图形程序的麻烦。

一、wxWindow系统设计

1. 系统介绍

wxWindow为用户提供了一个独立于平台的编程界面。用户利用这个界面编制的图形界面程序，从一个平台上可以不加修改地移植到另一个平台上。目前，wxWinodw支持的平台为Window3.1和X Motif。这个系统向用户提供一套完整的类库，可以基本上满足用户进行图形界面设计的需要。同时它还具有以下特点：

·编程容易：wxWindow对不同平台的编程接口(API)作高一层抽象，呈现给用户一个通用的以各种类为基础的接口，用户可脱离不同平台编程时所遇到的各种复杂、具体的细节问题。从程序框架到每个图形元素都以对象的方式提供给用户。用户要么利用系统提供的各种图形框架完成一个图形界面程序的拼装过程，要么对用户有必要介入和修改的类进行继承，对某些成员函数(如用户自定义的回调函数)进行重载以完成用户特定的功能。

·移植方便：用户在一种平台上编制的程序(如Windows)，可以完全不加修改地移植到另一个平台上(Motif)，重新进行编译、连接，两种平台上的界面是相似的。

·扩展容易：系统类库在基本类中实现所有与平台无关的功能函数，定义基本变量，而将与平台有关的函数以虚拟成员的形式表示。目前，两种平台(X Motif，Windows)上需要的类都是进一步继承基本类得到的。若要该系统扩展到第三种平台，所作的工作只是根据第三种平台的需要，继承基本类以及对与平台有关的成员进行重载。另外，类中的一些成员函数为用户提供了默认方式，也允许用户采用自己的解决办法。

2. wxWindow图形界面构造的层次结构

在window3.1的基本编程方式中，首先需要构造main window，然后编程者要写一个完整的消息

维普资讯 http://www.cqvip.com

循环的处理过程。对于 X Motif 来说，其构造思想就复杂许多：首先，用户要构造一个或多个 shell（在 Motif 中，shell 用来与 X 的窗口管理程序通信），然后选择适当的管理单元（Manager Widget）作为 shell 的儿子，最后构造用户需要的 widget（相当于 window3.1中的 Control）。整个结构是一种树形结构，用户不需要自行书写消息循环。

wxWindow 的应用程序也是由一系列窗口构成，窗口由一个框架（Frame）外加一个或多个子窗口（subwindow）构成。Frame 是一种抽象，综合了 main window（window3.1）与 shell（X Motif）的特点。一个 Frame 本身无父亲，可以最大化、最小化，可以具有菜单条，工具栏与控制菜单。一个 Frame 规定允许有四种子窗口作为其儿子（图1）。其中 Panel：容纳各种 Panel Item（button，checkbox 等）；Canvas：绘制各种曲线；Menubar：为 Frame 提供菜单条；Toolbar：为 Frame 提供工具条。

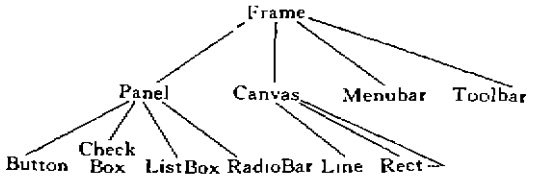

图1

这种结构本身比较简单，容易实现，同时这种抽象又是合理的，适用于大多数平台的情况，舍弃了各个平台的特异性。

二、wxWindow 类库设计思想

2.1 类库的可扩展结构

整个类库分为两部分：Win31 类（针对 Windows）和 X 类（针对 X Motif），这两个类库都由 Base Class 派生出来。类库在功能上分解也分两部分，界面构造类库（用来直接构造图形界面）和非界面构造类库（辅助功能，用来传递信息，存储数据等）。界面类库由统一的基类 wxobject 派生，见图2。

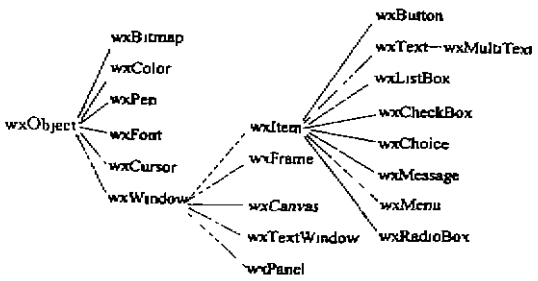

图2

2.2 界面构造类库的设计

2.2.1 管理窗口类 是一个图形界面程序的框架，或者说是最上层父亲窗口，作用相当于 Windows 中的（main window）或者是 Motif 中的 topLevelShell。wxFrame 类抽象出 Windows 与 Motif 窗口管理器的共同特性，虽然这两种图形系统的性质不同，但窗口界面都基本符合 OSF 的风格标准，都具有标题栏、最大最小化钮、边框、系统菜单。wxFrame 的主要功能是为用户提供程序的基本框架，还可以完成应用程序初始化以及程序退出时的收尾工作。它作为应用程序的主窗口使用，直接被操作系统的窗口管理程序管理，也就是说，它是程序的最上层窗口，没有父窗口存在。其主要的成员函数及其功能如下：

```
wxFrame::Command( ):处理系统菜单命令
wxFrame::GetTitle( ):获得标题信息
wxFrame::GetMenuBar( ):获得菜单栏的句柄
wxFrame::GetToolBar( ):获得工具栏的句柄
wxFrame::CreateStatusLine( ):创建状态行
wxFrame::Maximize( ):使窗口最大化，完成最大化钮的功能
wxFrame::Iconize( ):使窗口最小化，完成最小化钮的功能
wxFrame::OnSize( ):用户可以重载这个成员，当 Frame 大小发生变化时，用户自行指定要作的工作
wxFrame::OnMenuCommand( int id):当标识为 id 的菜单被选中后的动作，用户可以重载，根据实际情况自行处理
wxFrame::OnClose( ):当 Frame 被系统杀死（如在 Windows 中选择系统菜单的 close 或 Motif 中窗口管理程序杀死 shell）时执行该函数
wxFrame::SetMenuBar( ):为 Frame 设置菜单栏
wxFrame::SetToolBar( ):为 Frame 设置工具栏
```

以下是一个重载 OnClose 的例子：

```
Bool MyFrame::OnClose( void)
    {
      if (subframe)
        delete subframe;
        return TRUE;
    }
```

这说明当窗口管理程序结束 Frame 运行时，用户的重载函数可以显式地结束用户自己产生的子窗口。

2.2.2 子窗口类 有四种：wxCanvas、wxPanel、wxToolBar 和 wxMenuBar。

wxCanvas 的主要功能是用来绘制几何图形及书写文本，类似于 Windows 中的 paint brush，提供一组绘制基本图形的工具。其主要成员及其功能如下：

```
wxCanvas::Clear( )清除绘图区
wxCanvas::CrossHair( )使绘图区光标呈十字
wxCanvas::DrawArc( )绘制弧线
wxCanvas::DrawEllipse( )绘制椭圆
wxCanvas::DrawLine( )绘制直线
wxCanvas::DrawPolygon( )绘制多边形
wxCanvas::DrawPoint( )画点
wxCanvas::DrawRecttangle( )绘制矩形
wxCanvas::DrawRoundedRec( )绘制圆边矩形
```

维普资讯 http://www.cqvip.com

```
wxCanvas::DrawText( )书写文本
wxCanvas::SetBackGround( )设置背景
wxCanvas::SetBrush( )设置刷子
wxCanvas::SetFont( )设置字体
wxCanvas::SetPen( )设置笔形
```

wxPanel 主要为下述的控制单元类(Panel Item)提供一个容纳空间,类似一块黑板(Board),这种想法起源于 Motif 中的 BulletinBoard Widget。Panel 只起容纳的作用,实现起来比较简单。Panel 并不提供布局(Layout)功能,而由专门的类来实现。

wxToolBar 和 wxMenu 分别为 Frame 提供工具栏和菜单条。

2.2.3 控制单元类 由许多小类组成,对应于 Windows 中的 control 和 Motif 中的 widget。当然,control 与 widget 有一些不同之处,要作进一步抽象:所有这些小类(item)都是由一个基类 wxItem 派生,wxItem 由一些基本的成员控制这些小类的共同属性。

```
wxItem::GetBackgroundColor( ):获得 Item 的背景颜色
wxItem::GetButtonColor( ):获得 Button 的背景颜色
wxItem::GetLabelColor( ):获得 Label 的前景颜色
wxItem::SetBackgroundColor( ):设置 Item 的背景颜色
wxItem::SetButtonColor( ):设置 Button 的背景颜色
wxItem::SetLabelColor( ):设置 Label 的前景颜色
```

Panel Item 小类共有 12 种:wxButton、wxCheckBox、wxChoice、wxGauge、wxGroupBox、wxListBox、wxMultiText、wxMenu、wxMessage、wxSlider 和 wxText。

三、面向对象的编程模式

整个系统编程接口的设计思想是用户利用系统提供的类库,拼装自己需要的图形界面,整个应用程序本身也作为一个整体看成是一个类。系统为用户提供一个基本类 wxApp,用户程序整体看成是 wxApp 的子类,用户程序中不出现 main( )过程,这个过程包含在类库中。用户首先继承 wxApp,然后重载虚成员 wxApp::OnInit( ),最后产生一个用户子类的实例。wxApp 包含三个成员函数:

1)wxApp::OnInit( ) 编制用户程序时,用户首先继承wxApp类,然后重载OnInit( )成员,在重载函数 OnInit( )中,用户可以利用界面构造类库中所提供的 API 进行编程,但用户至少要先建立一个 Frame,即应用程序的最上层窗口,然后建立 wxPanel 或 wxCanval 的实例,最后再实例化 wxItem(控制类),如 Button,CheckBox。例如,用户可以采用以下作法:

```
Class DerivedApp::public wxApp
{
   pubulc
      wxFrame * OnInit(void);
};
wxFrame *DerivedApp::OnInit(void)
{
   wxFrame *the_Frame=new wxFrame(void);
   return the_Frame;
}
```

2)wxApp::MainLoop( )系统提供的消息循环成员函数。

3)wxApp::Initialize( )系统初始化。

结束语 作为一种不同平台间程序复用的尝试,wxWindow 提供了一种方法,但尚存在许多作者没有解决的问题,例如,Frame 的构造对一些特殊的界面设计,尤其在 X Motif 下,便显得无能为力,这说明还有许多工作有待改进。

参考文献

[1] Dan Heller 等,Motif Programming Manual for OSF/Motif Release 1.2,O'Reilly & Associates,Inc.

[2] Paula M. Ferguson 等,Motif Reference Manual for OSF/Motif Release1.2,O'Reilly & Associates,Inc.

[3] David Flanagan,X Toolkit Intrinsics Reference Manual for X11 Release 4 & Release5,O'Reilly & Associates,Inc.

[4] Adrian Nye 等,X Toolkit Intrinsics Programming Manual OSF/Motif1.2,Edition O'Reilly & Associates,Inc.

[5] UNIX SYSTEM V Programmer's Guide:ANSI C & Programming Support Tools,Prentice-Hall Inc.

[6] 蔡明志,Borland C++4.0使用与编程指南,清华大学出版社