Windows编程中的面向对象技术

陈文宇*

(电子科技大学计算机科学与工程学院 成都 610054)

【摘要】Windows编程是事件驱动的交互程序设计方法，虽然以动态链接库的方式提供了大量的API函数，但是用户界面难以实现，并且重用性、维护性和扩充性较差；面向对象的思想和技术，为Windows提供了许多基本用户界面对象，如窗口、各种控件等，针对面向对象技术的特点，结合Windows编程的优势，给出了Windows应用程序的用户界面的实现方法，结果表明，程序交互性好，而且便于重用、维护和扩充。

关 键 词 面向对象； Windows编程； 消息； 事件驱动

中图分类号 TP311

1 Windows程序设计特点

Windows要求以一种全新的思维方式进行程序设计，主要表现为以下两点。

1.1 事件驱动的程序设计

传统的程序主要采用顺序、关联、过程驱动的程序设计方法。一个程序是一系列预先定义好的操作序列的组合，它具有一定的开头、中间过程和结束，程序直接控制程序事件和过程的顺序。

事件驱动程序设计是一种全新的交互程序设计方法[1]，它不是由事件的顺序来控制，而是由事件的发生来控制，而事件的发生是随机、不确定的，并没有预定的顺序，允许程序的用户用各种合理的顺序来安排程序的流程。它是一种面向用户的程序设计方法，在程序设计过程中除了完成所需功能之外，更多的考虑了用户各种输入，并针对性地设计相应的处理程序，还是一种“被动”式程序设计方法。程序开始运行时，处于等待用户输入事件状态，然后取得事件并做出相应反应，处理完毕又返回并处于等待事件状态，其框图如图1所示。在图中，输入界面1,2,…n并没有固定的顺序，用户可以随机选取，以任何合理的顺序来输入数据。

1.2 消息循环与输入

事件驱动围绕着消息的产生与处理展开，一条消息是关于发生事件的消息，事件驱动是靠消息循环机制来实现的。Windows应用程序的消息来源有以下四种：

1) 输入消息：包括键盘和鼠标的输入。这一类消息首先放在系统消息队列中，然后由Windows将其送入应用程序消息队列中，由应用程序来处理消息。

2) 控制消息：用来与Windows的控制对象，如列表框、按钮、检查框等进行双向通信。当用户在列表框中改动当前选择或改变了检查框的状态时发出消息，一般不经过应用程序消息队列，而是直接发送到控制对象上去。

3) 系统消息：对程序化的事件或系统时钟中断做出反应。一些系统消息，如DDE消息(动态数据交换消息)要通过Windows的系统消息队列，有的则不通过系统消息队列而直接送入应用程序的消息队列，如创建窗口消息。

4) 用户消息：这是程序员自己定义并在应用程序中主动发出的，一般由应用程序的某一部分内部处理。

2001年8月27日收稿

* 男 32岁 硕士 讲师

在Windows下，由于允许多个任务同时运行，应用程序的输入输出是由Windows来统一管理。

Windows操作系统包括GDI、KERNEL、USER三个内核基本元件。其中GDI(图形设备接口)负责在屏幕上绘制像素、打印硬拷贝输出，绘制用户界面包括窗口、菜单、对话框等。系统内核KERNEL支持与操作系统密切相关的功能，如进程加载、文件I/O、内存管理、线程管理等。USER为所有的用户界面对象提供支持，用于接收和管理所有输入消息、系统消息并把它们发给相应的窗口的消息队列。消息队列是一个系统定义的内存块，用于临时存储消息，或是把消息直接发给窗口过程。每个窗口则维护自己的消息队列，并从中取出消息，并利用窗口函数进行处理，其框图如图2所示。

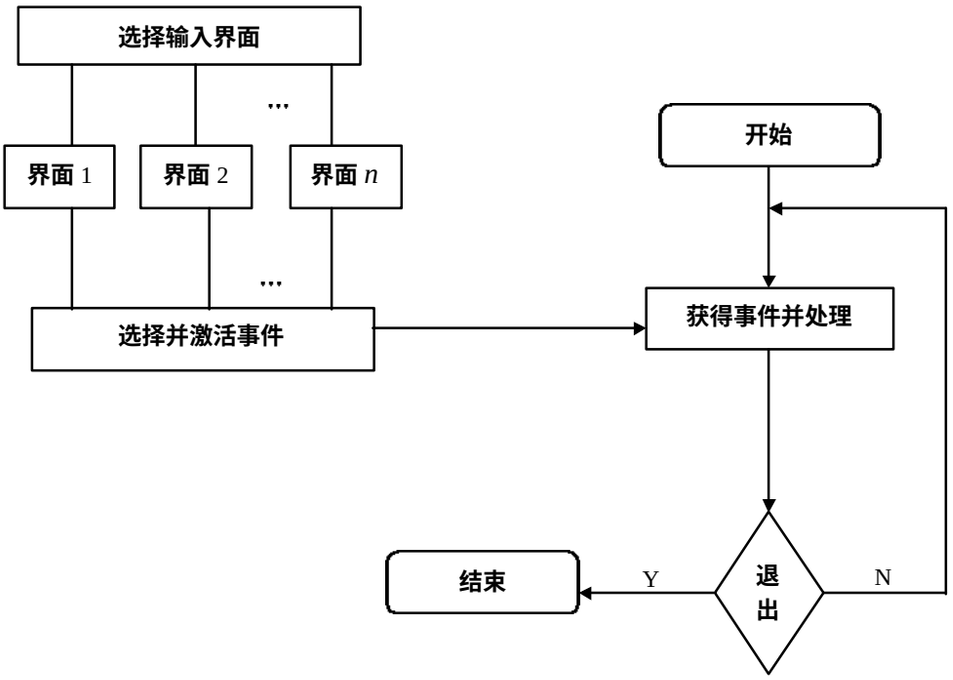

图1 事件驱动程序模型

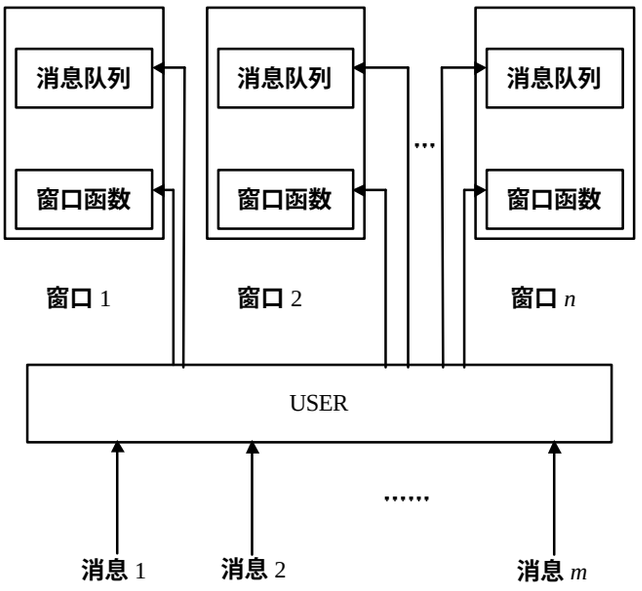

图2 消息驱动模型

2 面向对象技术特点

面向对象技术是目前流行的系统设计开发技术[2]，它包括面向对象分析和面向对象程序设计。面向对象程序设计技术的提出，主要是为了解决传统程序设计方法——结构化程序设不能解决的代码重用、可维护性和扩充性问题。

结构化程序设计从系统的功能入手，按照工程的标准和严格的规范将系统分解为若干功能模块，系统是实现模块功能的函数和过程的集合。由于用户的需求和软、硬件技术的不断发展，按照功能划分设计的系统模块必然是易变和不稳定的，开发出来的模块可重用性不高，可维护性和扩充性比较差。

面向对象程序设计从所处理的数据入手，以数据为中心而不是以服务(功能)为中心来描述系统。它把编程问题视为一个数据集合，数据相对于功能而言，具有更强的稳定性。

结构程序设计方法的核心是逐步细化，而自顶向下的方法是通过不断在程序的控制结构中增加细节来开发程序，生产的模块往往为了满足特定需要，可重用性较差。面向对象程序设计以数据结构为中心开发模块，同时一体化地考虑操作的功能，抓住了程序设计中最不易变的部分——数据，因此对象常具有良好的可重用性。面向对象程序设计同结构化程序设计相比最大的区别就在于前者关心的是所要处理的数据，而后者关心的是功能。

面向对象程序设计是一种围绕真实世界的概念来组织模型的程序设计方法，采用对象来描述问题空间的实体。一般认为，对象是包含现实世界物体特征的抽象实体，反映了系统为之保存信息和(或)与它交互的能力，是一些属性及服务的一个封装体，在程序设计领域，可以用“对象=数据+作用于这些数据上的操作”这一公式来表达。

类是具有相同操作功能和相同的数据格式(属性)的对象的集合，可以看作抽象数据类型的具体实现。抽象数据类型是数据类型抽象的表示形式，数据类型是指数据的集合和作用于其上的操作的集合，而抽象数据类型不关心操作实现的细节。从外部看，类型的行为可以用新定义的操作加以规定。类为对象集合的抽象，规定了对象的公共属性和方法，对象就是类的一个实例。如苹果是一个类，而放在桌上的苹果则是一个对象。对象和类的关系相当于一般的程序设计语言中变量和变量类型的关系。

消息是向某对象请求服务的一种表达方式。对象内有方法和数据，外部的用户或对象对该对象提出的服务请求称为向该对象发送消息，合作是指两个对象之间共同承担责任和分工。面向对象的编程方法具有以下四个基本特征。

2.1 抽象

抽象就是忽略一个主题中与当前目标无关的问题而只是选择其中的一部分，暂时不用部分细节。如设计一个学生成绩管理系统，考察学生这个对象时，只关心他的班级、学号、成绩等，而不用去关心他的身高、体重等信息。抽象包括过程抽象和数据抽象。过程抽象是指任何一个明确定义功能的操作都可被使用者看作单个的实体，尽管这个操作实际上可能由一系列更低级的操作来完成。数据抽象定义了数据类型和施加于该类型对象上的操作，并限定了对象的值只能通过使用这些操作修改和观察。

2.2 继承

继承是一种联结类的层次模型，并且允许和鼓励类的重用，它提供了一种明确表述共性的方法。一个新的类可以从现有的类中派生，称为类继承。新的类继承了原始类的特性，称为原始类的派生类(子类)，而原始类称为新类的基类(父类)。派生类可以从基类继承方法和实例变量，并且类可以修改或增加新的方法使之更适合特殊的需要。这也体现了大自然中一般与特殊的关系。继承性很好地解决了软件的可重用性问题。如所有的Windows应用程序都有一个窗口，可以看作都是从一个窗

口类派生出来的。但是有的应用程序用于文字处理，有的应用程序用于绘图，这是由于派生出了不同的子类，各个子类添加了不同的特性。

2.3 封装

封装是面向对象的特征之一，是对象和类概念的主要特性。封装是把过程和数据包围起来，对数据的访问只能通过已定义的界面。面向对象技术始于这个基本概念，即现实世界可以被描绘成一系列完全自治、封装的对象，这些对象通过一个受保护的接口访问其他对象。一旦定义了一个对象的特性，则有必要决定这些特性的可见性，即哪些特性对外部世界是可见的，哪些特性用于表示内部状态，在阶段定义对象的接口。通常，应禁止直接访问一个对象的实际表示，而应通过操作接口访问对象，称为信息隐藏。事实上，信息隐藏是用户对封装性的认识，封装则为信息隐藏提供支持。封装保证了模块具有较好的独立性，使程序维护修改较为容易。对应用程序的修改仅限于类的内部，因而可以将应用程序修改带来的影响减少到最低限度。

2.4 多态性

多态性允许不同的类的对象对同一消息作出响应。如同样的加法，把两个时间加在一起和把两个整数加在一起肯定不同。又如，同样的选择编辑-粘贴操作，在文字处理程序和绘图程序中有不同的效果。多态性包括参数化多态性和包含多态性。多态性语言具有灵活、抽象、行为共享、代码共享的优势，很好地解决了应用程序函数同名问题。

面向对象程序设计具有开发时间短、效率高、可靠性高等优点，所开发的程序更强壮。由于面向对象编程的可重用性，可以在应用程序中大量采用成熟的类库，从而缩短开发时间；应用程序更易于维护、更新和升级，继承和封装使得应用程序的修改带来的影响更加局部化。

3 Windows编程中的面向对象技术

在Windows中，程序的基本单位不是过程和函数，而是窗口。一个窗口是一组数据的集合和处理这些数据的方法和窗口函数。从面向对象的角度来看，窗口本身就是一个对象，Windows程序的执行过程就是窗口和其他对象的创建、处理和消亡过程[3, 4]。

Windows中消息的发送可以理解为一个窗口对象向别的窗口对象请求对象的服务过程。因此，用面向对象方法来进行Windows程序的设计与开发极其方便和自然。

采用面向对象的方法进行Windows程序设计还可以简化对资源的管理。当将资源映射成一个对象时，对资源的使用可以翻译成以下顺序：1) 创建一个对象；2) 使用对象；3) 撤消该对象。

一个对象的创建是对一个对象的定义过程，可以由对象的构造函数处理对资源的请求过程。当某一个对象退出活动范围时，它的撤消可以由编译器来自动管理。各种资源和Windows结构都能以这种方式处理，如设备上下文、画笔、字体、画刷等等。

4 Windows用户自定义界面的形成

Windows提供了基本用户界面对象，包括窗口、标题栏、图标、光标、插入符号、对话框、控件等。在Windows编程中，窗口是用户界面中最重要的部分，它是屏幕上与一个应用程序相对应的矩形区域，是用户与产生该窗口的应用程序之间的可视界面。每当用户开始运行一个应用程序时，应用程序就创建并显示一个窗口(利用窗口类的构造函数实现)；当用户操作窗口中的对象时，程序会作出相应反应(利用窗口类或窗口包含的对象的类的函数实现)。用户通过关闭一个窗口来终止一个程序的运行(利用窗口类的析构函数实现)；通过选择相应的应用程序窗口来选择相应的应用程序(窗口类可以有不同的窗口实例，每个窗口对应一个应用程序)。

用户界面必须包含一个窗口，是Windows中窗口类的一个实例(即对象)，可以将Windows提供

的其他基本用户界面对象定义在窗口对象内部，根据各个对象的位置、大小、颜色等属性，构成了整个用户界面，用户可以不关心各个对象的实现，只需要了解对象有哪些属性和能提供的操作(服务)功能，可以选择合适的对象使用。加上可视化技术的支持，用户在定义窗口对象和窗口内包含的对象时，实际上定义了一个在程序运行时的一致性的用户界面。

5 Windows图形输出的实现

Windows的图形输出是由图形设备接口(GDI)来完成的。GDI的图形输出面向窗口，面向窗口包含以下两层含义：

1) 每个窗口作为一个独立的绘图接口来处理，有自己的绘图坐标。当程序在一个窗口中绘图时，首先建立缺省的绘图坐标，原点(0，0)位于窗口用户区的左上角，每个窗口必须独立地维护自己的输出；

2) 绘图仅对于本窗口有效，图形在窗口边界会被自动裁剪，即窗口中的每一个图形都不会越出边界。即使想越出边界，窗口会自动地防止其他窗口传过来的任何像素。故在窗口内绘图时，不必担心会偶然覆盖其他程序的窗口，从而保证了在Windows下同时运行多个任务时各个窗口的独立性。

6 结 束 语

在Windows的界面设计和软件开发环境中，处处贯穿着面向对象的思想，故用面向对象方法来进行Windows程序的设计与开发极其方便和自然。

参 考 文 献

1 陈永远. Windows 应用程序设计原理·方法·技术. 北京: 电子工业出版社，2000

2 杨正甫. 面向对象分析与设计. 北京: 中国铁道出版社，1999

3 王予溶. 基于组件的应用程序设计. 北京: 电子工业出版社，1998

4 丁志强. 基于对象数据流图的可复用方案. 电子科技大学学报, 1999, 28(3): 306-311

Method of Oritened-object in Windows Programming

Chen Wenyu

(College of Computer Science and Engineering，UEST of China Chengdu 610054)

Abstract Windows programming is a interactive method which based on event-drive, it has many API functions which in DLL, but it is difficult to complete the user interface, and reusability、maintain and extendence is poor; oriented-object method and technology supply many base user interface objects, such as windows and controlled units. Based on the oriented object and windows programming feature, supply a method which can release the user interface of the application program ,and interactive、reusability、maintain and extendence is good.

Key words oriented-object; windows programming; message; event-drive