

面向对象应用程序的用户界面框架

俞一峻 钱乐秋

(复旦大学计算机科学系, 上海)

摘 要 面向对象应用程序的用户界面框架, 是一个结合面向对象和图形用户界面的优点的类属用户界面。用户界面开发者只要提供分类的操作集合对之加以参数化, 就能得到适合不同应用需要的用户界面。对用户界面使用者来说, 这种用户界面具有图形用户界面的外观特征和面向对象的使用特点, 所见即所得的直观性和超文本的导航性使它易学易用。本文介绍了这个用户界面框架的外观特征和使用特点, 并且用对象模型化方法提供了它的分析模型。

主题词 面向对象 用户界面框架

一、引 言

随着九十年代软件技术的发展, 越来越多的应用程序开始采用面向对象 (OO) 技术开发, 越来越多的用户界面开始采用图形用户界面 (GUI) 技术。通过应用程序设计接口 (API) 将两者结合起来, 已经成为大势所趋。

大多数的 GUI 使用面向对象方法处理对屏幕对象的操纵: 通过类定义引入新类型的窗口, 通过实例化屏幕对象来显示它, 通过在应用程序与用户界面引擎间发送消息来完成屏幕对象状态间的通讯。由于 GUI 的工具箱顶层是预定义的用户界面类层次结构, 所以在面向对象的用户界面基础上进行面向对象的应用程序开发, 是最自然不过的事情了。

尽管许多应用程序已经使用面向对象的开发技术, 但在其用户界面上仍旧沿用传统的命令调用模式。如何在外部的过程调用式用户界面与内部的面向对象实现之间接口, 是这类应用程序开发伤脑筋的问题。与其为每个应用程序增加相应的接口, 不如为整个这类应用设计一个通用的接口来得简单直观。为此, 作者设计了一个用户界面框架, 使之既有 GUI 的外观特征, 又有面向对象的使用特点, 且有类属的功能: 能够方便地同面向对象的应用程序连接, 使不同的应用可以重用它, 而不必为每个不同应用定制一个用户界面。

下文中第一部分, 是关于用户界面框架的外部特征和使用特点, 第二部分, 是用对象模型化方法 (OMT) 的记号描述的用户界面框架的分析模型, 包括对象模型、动态模型和功能模型, 第三部分告诉面向对象应用程序开发者如何使用这个用户界面框架。

1994 年 2 月 16 日收到本文

二、用户界面框架的外部特征和使用特点

1. GUI 的外观特征

一个用户界面框架主要由类选择区、操作选择区和实例选择区构成（见图 1）。

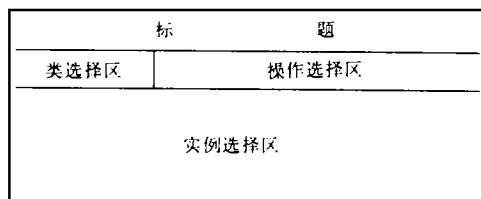


图 1 用户界面框架的外观特征

类选择区只显示一个菜单项，表示当前界面类。该菜单项的下拉菜单显示一组菜单项，每个菜单项分别表示一个界面类。操作选择区显示一组菜单项，每个菜单项对应一个操作。操作显示区只显示当前类的操作集合。实例选择区显示当前类的实例或实例集合。

类选择区和操作选择区一起构成用户界面框架的菜单系统，而实例选择区则是用户界面的主工作区。

2. OO 的使用特点

在上述用户界面框架中，如果在类选择区中选定一个类作为当前类，则操作选择区就切换到显示用户定义的作用于当前类或当前对象上的所有操作；实例选择区就切换到显示当前类的当前对象。

在操作选择区中选择一个操作，则根据该操作是类操作还是实例操作决定是否立即执行（类操作）还是等到与对象汇集时执行（实例操作）。

在实例选择区中选择一个实例或一组实例，则根据是否有实例操作与之汇集来决定是立即执行对该对象的实例操作（有实例操作）还是切换当前类，实现类似超文本 (HYPER-TEXT) 的导航(无实例操作)。

在上述用户界面框架的使用过程中，一个消息（或操作）的真正执行或汇集只有在类或实例与操作都选定后才会进行，即[类|实例]+操作。处理的主体是对象，这同传统的命令调用或用户界面的模式：命令+处理对象有很大不同，体现了面向对象的使用特点。

三、用 OMT 方法分析用户界面框架模型

OMT 方法是比较成熟的 OO 开发方法，它集合了许多其它 OO 方法的长处，能够支持从分析到设计、实现的全过程，具有分析模型与设计模型较为一致的优点。OMT 方法的记号系统较为完备，采用 OMT 方法分析用户界面框架模型，既可清晰地表达用户界面框架中的概念和工作原理，又可明确其使用特点。基于 OMT 分析模型的对象模型、动态模型和功能模型，界面开发人员就可以结合自己的 GUI，直接用 OO 语言设计实现一个能达到上述功能的应用程序用户界面了。

1. 对象模型

一个应用程序可以包含若干信息区，一个或多个面向对象用户界面。每个用户界面由

类选择区、操作选择区和实例选择区组成。类选择区包含所有用户类（元类的实例），对应一个当前用户类；操作选择区包含所有操作集，对应一组当前类的操作集；实例选择区包含所有实例集，对应一个或一组当前类的实例。元类的实例是一个用户类，包含一组用户类的实例和一组用户类的实例操作或类操作。

图 2 即反映了上述分析的 OMT 对象模型。

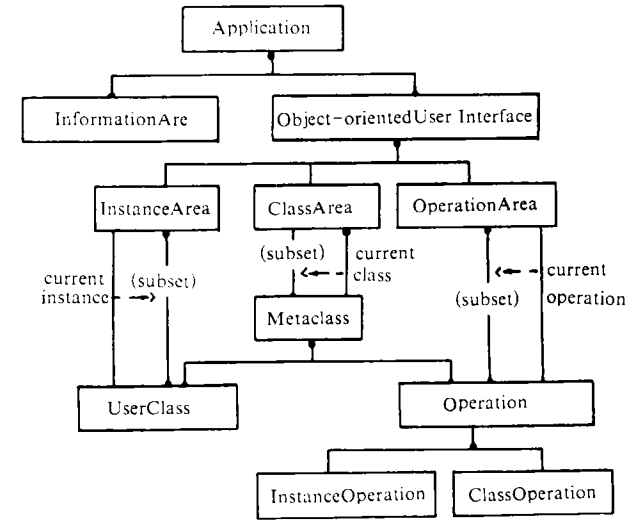


图 2 用户界面框架的对象模型

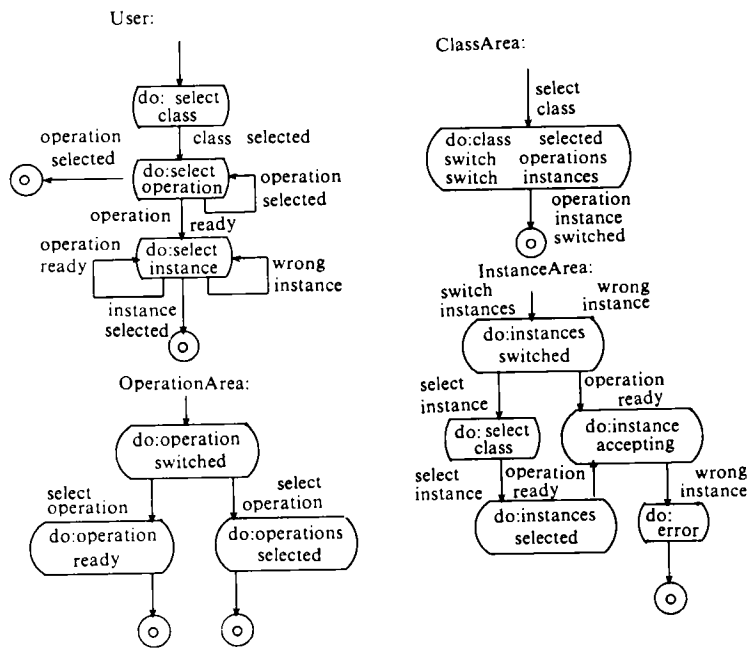


图 3 用户界面子系统的动态模型

2. 动态模型

由于用户界面框架的使用者包括应用程序的用户界面开发人员和终级用户两类，可以

对两个对应子系统的相对独立的脚本加以分析。

图 3、图 4 分别是与上述终端用户和开发人员的使用脚本对应的动态模型。

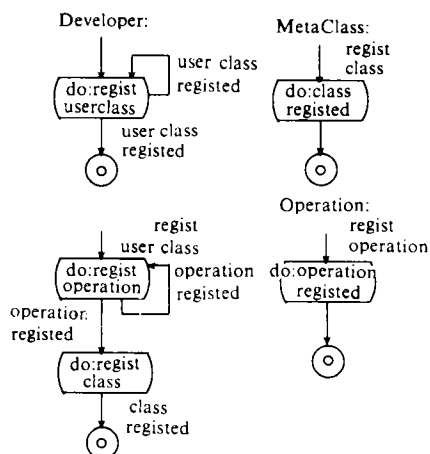


图 4 用户界面框架子系统的动态模型

3. 功能模型

OMT 的功能模型描述系统中值的变换、功能变换、映射变换，限制和功能依赖，通常用数据流图 (DFD) 来表示。

面向对象应用程序的用户界面框架功能模型中，系统与开发人员、终端用户之间形成了最顶层的 DFD，对其进一步功能分解的结果是图 5 所示的 DFD。

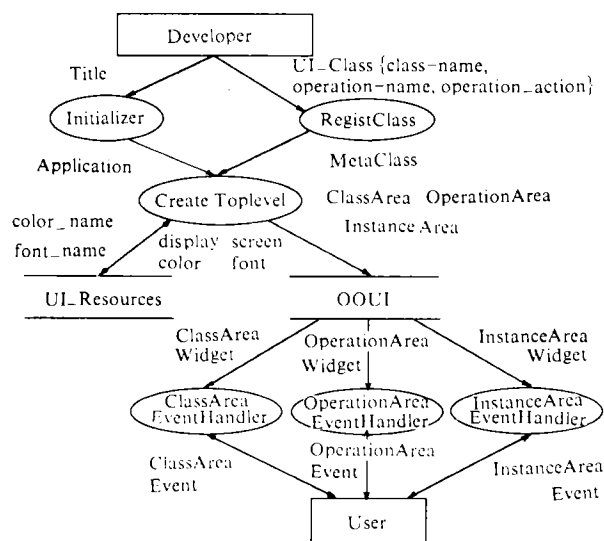


图 5 用户界面框架的功能模型

四、用户界面框架的应用过程

应用上述用户界面框架可以遵循下列步骤：

0. 完成面向对象应用程序中除了用户界面以外的其余各子系统的设计和实现。

1. 将所需用户界面的全部菜单命令按照所处理的对象加以分类。例如，数据库应用的用户命令可按数据库（文件或关系）、记录、字段加以划分；分层图形编辑应用的操作可按文档、层次、图、图元加以区分。

2. 为每一类操作创建一个用户类，作为用户界面框架中 UserClass 的子类。定义其类名、消息（操作名）、方法（操作实现过程）。

3. 将用户类注册到用户界面框架中。

4. 实现所有上述类方法。

5. 为用户类实现其实例选择方法。这个方法因应用而异，目的是实现实例和操作的动态汇集，另外还可实现类似超文本的导航（通过实例选择驱动类的切换）。

五、结束语

面向对象的应用程序开发人员往往对图形用户界面还不熟悉,或者因为开发用户界面占据了宝贵时间而苦恼,用户则经常对开发者提供的界面不满意。面向对象应用程序的用户界面框架,巧妙地结合了面向对象技术和图形用户界面技术的优点,为开发人员提供了一个类属用户界面,还能快速地生成用户界面原型,便于进行螺旋递进式的开发;为终级用户提供了具有面向对象使用特点的图形界面,便于实现所见即所得的开发;为终级用户提供了具有面向对象使用特点的图形界面,便于实现所见即所得、具有类似超文本特征的易学易用的用户界面。

作者用 C++ 语言在 X windows 的图形用户界面上实现了上述用户界面框架,应用于图形编辑、正文编辑及查询工具界面的开发,有效地提高了用户界面的重用度。

参 考 文 献

- [1] A.Goldberg & D.Robson, Smalltalk-80 the Language, Addison-Wesley, 1989.
- [2] P.Coad & E.Yourdon, Object-Oriented Analysis, Second Edition, Yourdon Press, 1991.
- [3] J.Rumbaugh, M.Blaaha, W.Premmerlani, F.Eddy & W.Lorensen, Object-Oriented Modeling and Design, Prentice Hall, 1991.
- [4] OSF / Motif, Revision 1.1, Open Software Foundation, 1990.
- [5] B.Stroustrup, The C++ Programming Language, Addison-Wesley, 1986.

(下转第 41 页)