

信息系统可重用界面构件的研究与实现

Research and Implementation of Reusable Interface Component in Information Systems

孙鹏飞 (北京邮电大学软件学院 北京 100876)

摘要:信息系统的开发工作量大,目前还没有有效的基于可重用构件的开发方法。本文首先对信息系统界面进行了分析,然后提出了一种信息系统内设计和实现可重用界面构件的方法,并且用实例阐述了可重用界面构件的设计和运行,最后对下一步的工作做了探讨和展望。

关键词:可重用 构件 界面 xml

1 引言

随着构件技术的发展,基于构件的软件重用技术逐渐成为现实的、主流的软件重用技术,因为构件体现了产生软件重用思想的初衷——让软件生产像硬件生产一样,通过组装已有的部件(即构件)而不是重新生产来制造新的产品(即软件系统)。

2 信息系统界面分析

现代图形用户界面都是由窗口(其上有各种控件)组成,窗口是屏幕上被边框围住的一部分,它有标题而且通常有一个系统菜单,可以在屏幕上移动,可以关闭、最大化或最小化,可以在屏幕上或在其它窗口中移动(如 MDI 应用程序)。各窗口的相互关系通过窗口间的消息传递来实现,一个窗口向另一个窗口发送一条消息,就激活了该窗口^[1]。常见的控件如:按钮、文本编辑框、列表框、下拉框等。

根据本课题组在多个系统的经验,我们对其界面部分进行归纳总结,分析出信息系统界面部分通常都包括如下的几类窗口。

其中图 1 中上方的单向箭头为输出接口,带有输出接口的窗口经过用户的一些操作(比如录入数据的操作)之后,将要传递的数据通过输出接口传递给对输出数据感兴趣的其它窗口。如查询条件设置窗口通过输出接口输出查询条件——将查询条件传递给对其有需要的

调用窗口(如查询窗口需要它来选择符合一定条件的数据实体)。

下方的单向箭头为输入接口,带有输入接口的窗口如果要完成本窗口的业务功能,要通过输入接口传入本窗口需要的数据集。如二维表显示窗口、树形结

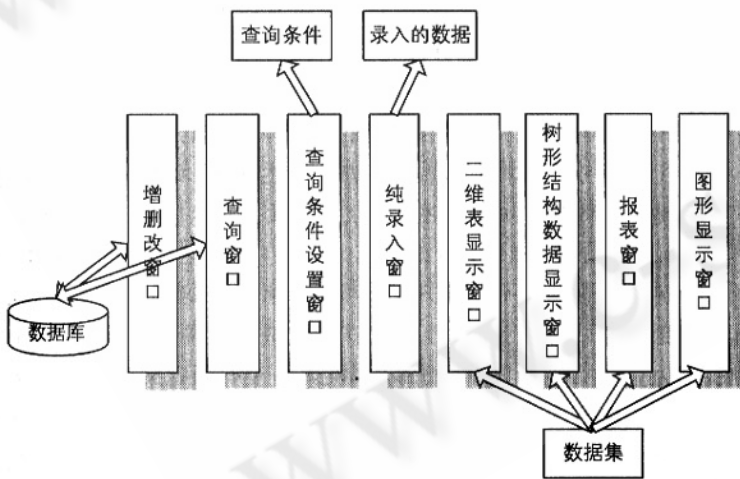


图 1 信息系统常见窗口

本文首先分析信息系统的界面,然后提出设计实现可重用界面构件的方法,最后用实例介绍了其设计样式,在运行时显示出了其可重用性。

构数据显示窗口、报表窗口和图形显示窗口需要数据集作为其显示的数据来源,这个数据集一般是从其它窗口传递过来的。

左侧的双向箭头为双向接口,表示箭头两侧需要彼此互相传递信息,如增删改窗口和查询窗口需要数据库操作接口与后台 DBMS(数据库管理系统)交互以完成其增加、删除、修改或查询的操作^[2]。

通过上述这些接口,这些窗口就可以协作以完成信息系统界面部分的基本业务功能。

3 可重用界面构件的研究和实现

当前 4GL 语言的集成开发环境(如 Delphi、VB)所开发的用户界面(由一个个窗口组成)可分为两部分,一部分是窗口上的各个界面元素(文本编辑框、下拉框等等),另一部分是和窗口元素相关联的处理代码,这两部分是绑定在一起的。当开发不同的信息系统时,由于针对的具体应用不同,所以窗口上元素的类型、个数等也会不同,这样牵涉到处理代码的变动量很大。

开发新系统时如果想重用原来的程序,因为窗口元素的变化,开发人员常常都需要大量改动原有的代码,工作量非常大。由于信息系统界面部分业务处理的相似性,这种工作往往是重复性的。因此,我们考虑到应该抽取相似窗口的共性部分,将其个性部分设计成为可定制或可替换的(这部分也称为可变性)。

3.1 可重用界面构件的设计实现

基于上面的考虑,本文提出了一种设计实现可重用界面构件的方法。

对于上面分析的信息系统界面部分包含的几类窗口,将其上面的元素分解成共性和个性两部分,共性部分仍然在原来的开发环境下在窗口上设计完成,称为窗口模板;对于个性部分,用中立的描述语言来描述,形成可变性描述文档;要实现某个界面业务功能,可能需要一个或几个窗口来协作完成,设计一个主控单元,将相关的几个窗口及其元素的处理代码都分离出来而在主控单元中编写,并利用主控单元将这几个窗口封装为一个界面构件。

本文的研究工作是在 Delphi 的开发环境下进行验证的。按照上面的研究思路,下面详细介绍以 ActiveX 构件技术为基础的可重用界面构件的三部分的设计实现。

(1) 窗口模板。模板相当于工业生产中所用的

“模具”。信息系统界面的同一类窗口,往往提供相同的业务处理功能,所不同的是要处理的数据实体。以增删改窗口为例,因为对具体数据实体所进行的处理都相同,如增加、删除、修改记录等等,所以这类窗口都有一些共同的元素,如增加、删除、修改等按钮。窗口模板表示了同类业务处理都具有的布局样式,包含了同类业务处理都需要共有的窗口元素,它是窗口创建的基础。实现方式是在 Delphi 的开发环境下,利用 Delphi 中的 Form,在其上添加一些同类窗口都有的共同元素,实现窗口模板。

(2) 可变性描述文档。窗口模板只包含了同类窗口共性的部分,但是同类窗口中的每一个窗口又有其个性的部分,比如所要操作的数据库和表不同,所以每个窗口都会有自己的个性化的窗口元素。即使窗口元素都相同,它们在窗口上的布局也有可能不同(本文中所述的窗口元素的布局,不仅仅指窗口元素在窗口中的位置,也指为了表现一个具体的数据项,所用的表现数据的控件)。

虽然各种开发环境对于窗口以及其上元素的内建描述格式都有自己的一套定义,但是它们所描述的内容相同或者相似。它们大都是描述窗口中有些什么界面元素、这些界面元素有什么样的特征、它们和数据项的关系等等。

对于这种层次性的对窗口元素的描述方式,完全可以采用 xml 语言来进行描述,利用 xml 的元素、属性等可以足够描述窗口上的元素及其之间的关系。重要的是,由于 xml 的语言无关性,采用这样的标准,有利于不同的开发环境交流和重用这样的描述文档。目前对于 xml 也已有许多的成熟工具支持,可以充分地利用现有的资源。xml 本身没有预定义元素类型或内容,每个 xml 文档定义其自己的结构、元素类型和内容,其可扩展性保证了可以定义新的用于描述复杂的窗口元素及其关系的信息结构。

因此对于同类窗口可变元素,我们采用了 xml 语言来描述。

(3) 主控单元。和通常的开发模式不同,将和窗口元素相关联的代码分离出来单独编写,作为独立的函数或过程。要编写处理什么业务的函数或过程,和窗口要进行什么样的业务处理有关系,以增删改窗口为例,则必定有用于增加、删除、修改等业务处理的函

数或者过程。这一部分也在 Delphi 的开发环境下完成,实现时将分离编写的代码都放在一个 ActiveXForm 单元中 (Delphi 中提供,符合 ActiveX 构件标准)。

依照前面对于信息系统界面分析所得到的基本窗口,我们设计了信息系统的基本窗口模板,如:增删改窗口模板、查询条件设置窗口模板、二维表显示窗口模板等等。同时将原来在各个窗口中的业务处理函数或过程分离编写,如果窗口组合在一起协同完成某业务功能,就把和它们(包括其上元素)相关联的业务处理代码都放在同一 ActiveXForm 主控单元中。作为主控单元,ActiveXForm 同时负责将窗口封装起来形成界面构件。

3.2 共性、可变性的结合

上面的设计已经分离了界面构件的共性和可变性,而何时将它们结合在一起,或者说何时将可变性部分引入到程序中,是有两种方式的(生成时引入方式和运行时引入方式)。使用这两种引入方式均不受可变性信息如何表示的限制,但都要清楚地将可变性从程序中分解出来。它们的不同之处在于:对于生成时引入方式来讲,其可变性信息通过一个创建定制程序的程序生成器读入,生成定制程序之后,编译运行之;而对运行时引入方式来说,其可变性信息一般保存在一个外部文件(特性文件或数据库等)中并在程序运行时通过程序读入^[3]。如果使用生成时引入方式,当可变性部分改变了,需要重新生成程序,重新编译,才能够重用。而使用运行时引入方式,可以在程序运行的时刻通过外部文件将可变性部分引入进来,不需要重新编译,直接达到重用的目的。我们采用运行时引入方式作为构件共性和可变性的结合方式。

界面构件运行时,首先运行主控单元,接着生成所需要窗口模板的实例。然后,读入窗口模板对应的可变性描述文档,根据文档,在窗口模板上添加个性化元素。接着在主控单元中定义对于窗口模板中以及在其上动态创建的元素引用,将主控单元中的已经定义的函数或过程指针赋给这些窗口元素的事件处理函数,即将这些窗口元素和主控单元中的各个函数或过程挂接起来。图 2 是运行时候的流程。

3.3 实例阐述可重用界面构件

接下来我们介绍一个单表数据录入与维护的界面功能,我们将其设计为一个数据录入与维护界面构件。图 3 所示的是构件中封装的增删改窗口的窗口模板,

其中窗口标题区和数据实体区是可变的,右侧的按钮区域是属于窗口模板的。

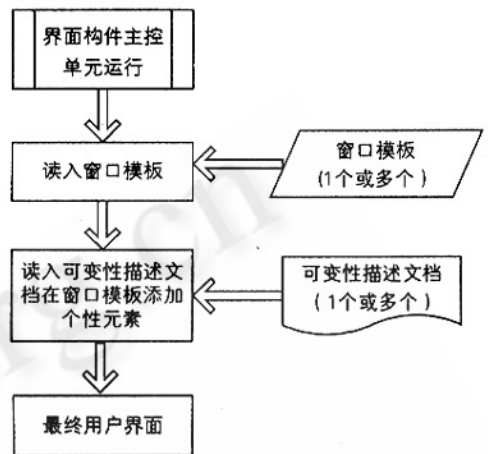


图 2 运行时结合共性和可变性

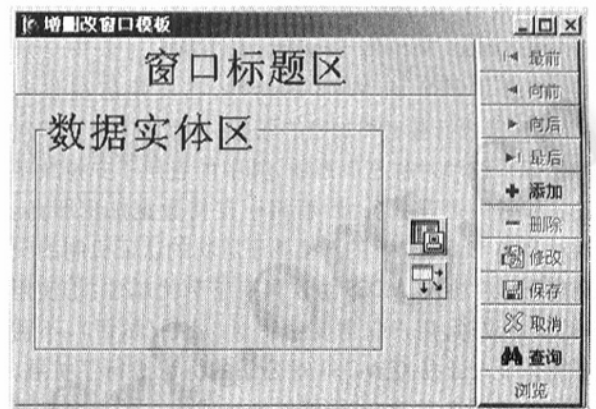


图 3 增删改窗口模板

在信息系统的开发过程中,用户的需求经常会有变化,反应在界面上就是开发人员常常要在窗口上增加或减少一些数据显示控件,或者在窗口中用其它的控件来显示数据项等,然后导致修改相关的代码,开发工作大量反复。而现在只需要编辑描述窗口可变元素的 xml 文档,就可以实现所需的业务要求。这样可以有效地重用原来的构件,不必因为只是一些窗口元素的变动,就去修改原来的程序代码。

在主控单元 ActiveXForm 中我们为该构件定义一个 getParam 函数,接收用于描述 xml 文档的相关参

(下转第 46 页)

数。界面构件运行时,xml 文档的路径、名字等参数由这个接口传入。对于同类的业务处理,不同的数据实体,只需要通过接口传入不同的参数,不需要对程序做任何变动,就可以满足用户的要求。按照这种方式,在不同的信息系统之间,也可以透明地重用开发的界面构件。

从上面的例子可以看出,只需要定义一个新的可变性描述文档,就可以“透明地”重用原来的构件。对于各种不同的窗口,我们已分别做出了窗口模板,实现了通用的业务处理代码,并封装成处理信息系统常用界面业务(如录入与维护、统计分析、查询显示等)的一些界面构件。在实际应用的时候,开发者只要用 xml 描述和具体业务相关的窗口元素,就可以充分利用已有的软件资源,这样就大大提高了可重用性。

4 结束语

本文介绍了一种信息系统可重用界面构件的设计

和实现方法,对减少信息系统开发维护的工作量是很有益的,使用这种“透明的”构件重用方法,可以大大减轻开发人员的负担。

参考文献

- 1 刘艺, Delphi 模式编程[M],北京 机械工业出版社, 2004。
- 2 张鹏,界面构件软件工厂的研究与实现[D],北京 北京航空航天大学计算机科学与工程系,2003。
- 3 J. Craig Cleaveland. 用 xml 与 java 创建程序生成器 [M],北京 科学出版社,2002。
- 4 飞思科技, Delphi 7 下 IntraWeb 应用开发详解 [M],北京 电子工业出版社,2003。
- 5 杨芙清,软件复用及相关技术[DB/OL],中国期刊网 www.cnki.net,1999。