

文章编号: 1000-5013(2007)02-0135-04

以数据库为中心的应用系统开发的一种新模式

余金山, 朱永辉

(华侨大学 信息科学与工程学院, 福建 泉州 362021)

摘要: 结合当前软件开发的若干新概念和新技术, 如体系结构、设计模式、面向 GUI (Graphical User Interface)、软件重用与软构件、软件演化等, 对以数据库为中心的应用系统的开发提出一种新的模式, 实现相关的基础设施和支持工具。它首先设计和选择一个合适的体系结构, 再通过设计模式命名、抽象化, 确定一个通用设计结构, 保证使用设计模式的软件子系统有较好的灵活性和可重用性。通过对应用系统的基本行为分析, 建立一个以 GUI 为驱动的基本操作行为模式构件库。该数据库可为终端用户的应用程序开发提供一种高效灵活、简单易用, 且能适应用户个性化和特殊应用逻辑需求, 以及能进行演化以适应业务发展的变化的机制。

关键词: 信息管理; 体系结构; 设计模式; 软构件; 面向 GUI

中图分类号: TP 311.13

文献标识码: A

目前的管理信息系统都是采用以数据为中心的结构, 这些应用系统涉及范围广, 但其可扩展性和可重用性都不强。因此, 在应用系统开发的过程中, 无论是初期的整体规划还是后期的二次开发, 都给开发者带来了很大的麻烦, 浪费了大量的重复劳动, 并缺乏实用快速的开展工具。本文对比了当前各种应用系统的开发特点, 提出了一种新的开发模式, 设计并实现了能很好地支持终端用户进行应用开发的基础设施、操作行为模式库和相关工具。

1 应用系统的基本行为分析

1.1 批处理模式

批处理以一个画面为基本单位。它由系统送出一个屏幕的画面内容到终端设备后, 就等待用户从终端设备输入数据。用户将一个屏幕的数据输入完毕后, 由终端机将整个屏幕的数据送回主机进行处理。这个特性将加入、修改、删除、查询等基本功能分别进行独立处理, 且由用户从系统的菜单中选择。

1.2 联机处理模式

联机处理是指用户的输入始终由系统监视着, 并可即时作出回应。从实现的角度看, 系统必须提供并实现以下 3 个基本操作行为。(1) 单表格操作行为。它提供并实现对数据库中每个基本表的操作, 可归纳为指令(菜单)分割法和主键分割法。前者直接通过菜单、图标、按钮等, 或通过指令的方式去区分不同的操作; 后者根据输入的主键值去区分用户想要进入的操作。(2) 列表式操作行为。它也有两种处理方法:(a) 标准表格处理方式。这种处理方式是用户利用光标在上下记录之间, 以及在左右栏位之间的游走, 到达要处理的位置后才进行操作。这种方式直观易用, 但实现比较复杂。(b) 窗口式表格处理方式。这种处理方式是现行指针所指的记录直接映射到一个固定的窗口, 只有窗口上的记录才可以修改。用户同样可以用光标在上下两个记录间游走, 但只有在固定的窗口上才可以作栏位间的移动。窗口式的表格处理方式, 可以看成是标准表格处理模式与单表格处理模式的混合体。(3) 主-细目式操作行为。主-细目式操作提供并实现对具有主-细目信息结构的处理, 其信息结构由一个包含交易资料的公共部分的主表格, 以及一个或多个记录详细交易内容的细目表格所组成。主-细目型应用程序的行为特征

收稿日期: 2006-07-09

作者简介: 余金山(1952-), 男, 教授, 主要从事软件工程与网络计算和人工智能的研究。E-mail: yjs@hqu.edu.cn.

基金项目: 国务院侨办科研基金资助项目(05QZR14)

与前面所讲的行为特征的主要区别在于细目表的处理,而处理细目表的较好方式是列表式方法,它与前面所讲的列表式的处理方式有所不同. 我们的设计主要考虑了以下 2 个问题. (1) 要加入细目部分信息的控制. (2) 把细目信息的删除和主-细全条信息的删除明显区分开来,并施加完整性保护,以避免混淆和误删.

2 设计思想

首先,我们从简单易用的用户需求和构件化、可重用的思想出发,给出了一种合适的体系结构. 接着,根据体系结构,以规范性、可演化^[1]、灵活性、个性化等为目标,融入设计模式、GUI 驱动、构件化和构件演化的概念,设计并实现了能很好地支持应用开发的操作行为模式库和相关工具. 最后,给出了用户进行应用开发的一种以 GUI 为驱动,以操作行为模式库为基础的,通过继承和演化的开发模式.

2.1 系统的体系结构

软件体系结构是系统的总体框架,涉及系统的整体性问题^[2-3],它对信息系统及其支撑工具的开发具有重要意义,其设计与选择的结构如图 1 所示. 图中,视窗系统为 GUI 数据操作组件(Data Operation Component)的实现提供基础支持;数据库组件(Database Component)是一个服务器,负责对数据库中存储信息的进行实际操作,并使得不同的信息源对 GUI 数据操作组件是透明的;连接器负责协调消息的传递. GUI 数据操作组件由 4 个部分组成. (1) 存储状态和实现操作的内部对象. (2) 各由一组允许传送和接收消息所组成的接口. (3) 对话映射. (4) 包装器. 包装器监视所有的操作请求并负责传送告知消息,对话映射负责把接收到的操作、请求消息映射为内部的实际操作,或把告知消息映射为显示的说明.

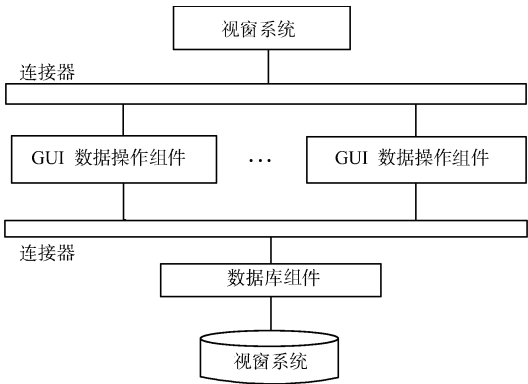


图 1 系统的体系结构

Fig. 1 System architecture

2.2 设计模式

针对问题的目标和设计原则的要求,我们采用所谓的观察者模式(Observer Pattern)^[4]. 该模式允许一个目标有任意数目的观察者,可独立地改变目标和观察者,或单独复用目标对象或观察者,或在不改动目标和其他观察者的前提下,增加新的观察者或删除已有的观察者. 因此,可以看到这种设计模式和我们设计的体系结构的思想要求也是一致的.

2.3 操作行为模式库

操作行为模式库是一个以 GUI 为驱动^[5]的基本操作行为模式构件库,为终端用户的应用开发提供一种规范的、高效灵活且简单易用的支持.

3 基本行为模式库的设计

前面我们对以数据库为中心的应用系统的一般操作行为进行了分析、归类和归纳,其目的是要对应用系统开发时经常碰到及使用到的一些行为习性进行组合,定义出相关接口,以制作出可复用的构件. 不同操作行为的构件集合,便可构成一个基本操作行为模式库. 这个模式库的主要功能是让用户可以针对自己的应用系统的具体需要,取用不同的行为模式构件,从而大大地简化开发工作,并且使系统的整体运作有一个相对稳定的操作模式. 同时,对于用户的一些个性化要求和例外情况处理,则可通过接口参数的简单调整、继承、包含或添加自己所需的代码等方式实现. 下面分别给出各主要构件的设计. 图 2 描述了这一基本思想和相互之间的关系,在结构上可分为 3 个大层次. 第 0 层取自一般的通用视窗对象,第 1 层是本项目实现的模式库中的构件,第 2 层用于用户的个性化定制和所需的例外处理.

3.1 单表格通用操作构件的设计

单表格通用行为构件(Data Window)的实现,满足以下 4 点要求. (1) 容易使用. (2) 灵活性高. (3) 开发应用所需编写的代码量少或无需编写代码. (4) 支援当今的大部分数据库. Data Window 提供如下

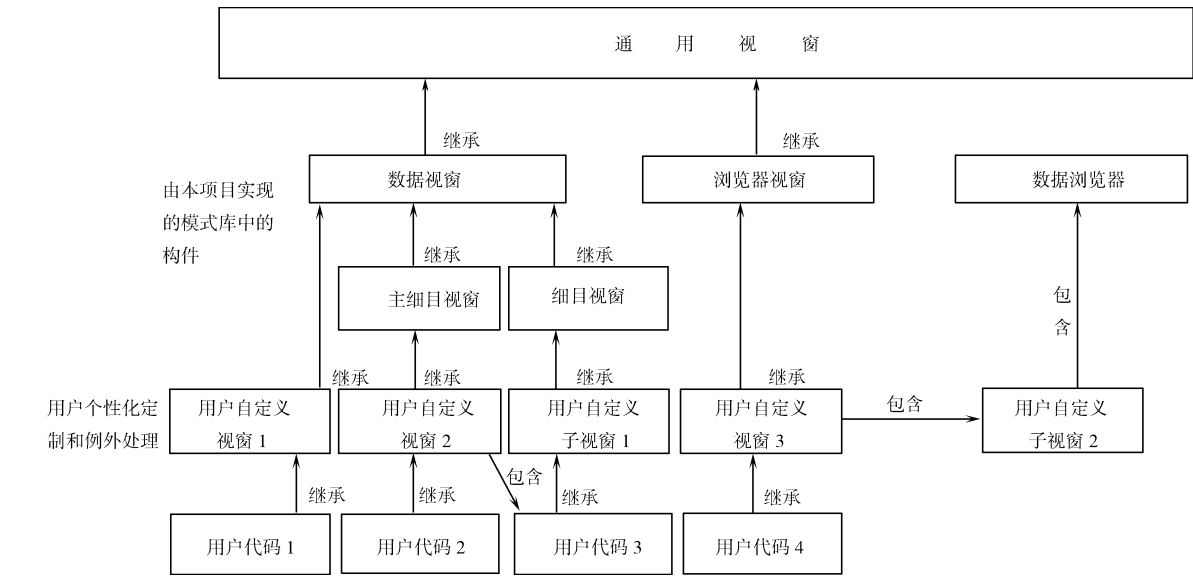


图 2 基本行为模式构件及其结构

Fig. 2 The components and structure of components for essential operating behaviors

8 种的能力。(1) 数据通晓。即将数据库表格上当前最新的字段名和字段值直接映射到图形用户界面上。(2) 缓冲修改能力。允许用户取消所作的修改。(3) 允许程序员在中间加插源代码进行特殊的处理。(4) 主键字段划分。能由用户自行划分和描述主键字段。(5) 对不被当前窗体包含的主键的处理能力。(6) 对不同的数据库源,如 Access, Oracle, IBM DB2, SQL Server 等,均提供相同的数据通晓能力。(7) 在不同的操作状态下,能自动对屏幕上的各种功能菜单、按钮、图标等进行激活或灭活处理。(8) 用户通过简单的操作,就能定制或重新安排包括窗体、按钮、菜单、图标等在内的整个界面。

3.2 表式通用操作构件的设计

列表式通用操作构件的结构,如图 3 所示。在实际应用中,程序员所关心的并不是构件的功能,而是从宏观来看整个建构于 DataBrowser 之上的 BrowserWindow 的工作行为是否满足一般的应用系统。因此,DataBrowser 要有足够支援 Browser Window 所提出的各种要求的能力。对于 DataBrowser 的扩展处理,除了要有基本的数据浏览功能外,还要加入数据列的写保护及虚拟列的设立。至于对数据的加工处理与储存,可以留待 BrowserWindow 进行。这样做可以保持与 DataWindow 对数据处理的统一性,又可以将数据处理的层面集中到与程序员直接接触到的 BrowserWindow 上,减少涉及 BrowserWindow 的内部处理流程。BrowserWindow 提供与 DataWindow 类似的各种能力。

3.3 主-细目式构件的设计

由于要处理主-细表格之间的对应关系和数据的完整性问题,主-细目式构件的设计与实现也复杂得多,并不是单表式和列表式的简单叠加。我们用两个构件 MasterDetailWidow 和 DetailWindow,并通过继承关系和包含关系加以实现,如图 4 所示。MasterDetailWidow 内包含有管理细目表格的功能。为了方便地开放接口给程序员使用,列表式的输入部分以一个独立的子视窗的形式存在——DetailWindow;而 MasterDetailWidow 与其内的子视窗 DetailWindow 的关系是一个包含的关系。定义子视窗是为了将来升级的简易性和更高的灵活性。由用户绘制的子视窗,并不直接为主视窗所包含,即主视窗内所定义的子视窗的名称,并非主视窗直接引用的名称,而是经过继承后的用户构件的名称。由于 MasterDetailWindow 亦都要对子视窗进行操控,因此,用户子视窗必须继承子视窗构件 DetailWindow。另外,由于主-细目视窗的行为模式与单表格通用操作构件 DataWindow 的行为模式相类似,因此,主-细目式的通用构件可以继承 DataWindow 的一切行为及特

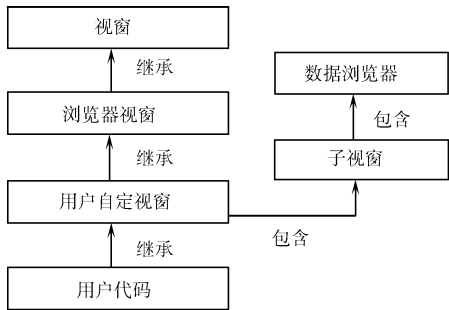


图 3 列表式操作构件

Fig. 3 The component for listing style operation

征.然而,子视窗由于是相对独立运作,与 DataWindow 的行为不同,因此要另行定义.

3.4 应用

终端用户利用本模式进行应用系统开发,其基本过程有 5 个方面.(1) 以 GUI 为驱动建立系统的行为模型.(2) 以每个基本的 GUI 和基本行为为基础,选用相应的操作行为构件(或其组合).(3) 用以上选择的构件构造出系统原型.(4) 运行原型.(5) 对于用户的一些个性化要求和例外情况处理,可通过接口参数的简单调整、继承和包含等方式实现(图 2).

4 结束语

在实际的应用中,我们用这种方法和已实现的相关工具,开发了包括客户资料处理、发票系统、销售管理系统、佣金计算系统、成本及财务报告系统、年度生产及财务预算系统等.在开发的过程中,充分体现了它的效率、灵活性及适应性.相较于过往开发相同系统的经验,开发时间减少了 50%~ 80%,而需编写的程序代码量减少 70%~ 90%.由于所编写的代码不需要对程序的行为进行监管,有助提高代码的可维护性.另一方面,程序的基本行为已规范化,故对所编写的代码起到一定的统一作用.

参考文献:

[1] 符进强,汪强,钱乐秋.基于动态构件框架的构件演化[J]. 计算机科学, 2001, 28(1): 21- 24
[2] SHAW M, GARLAN D. Software architecture: perspectives on an emerging discipline[M]. New Jersey: Prentice Hall, 1996: 69- 129.
[3] 胡华,高济,何志均.软件高层抽象设计构造的需求及发展[J]. 计算机科学, 1999, 26(2): 64- 73.
[4] GAMMA E, HELM R, JOHSON R, et al. 设计模式: 可复用面向对象软件的基础[M]. 李英军,译. 北京: 机械工业出版社, 2000: 194- 201.
[5] 王咏,徐贯东.面向 GUI 的软件开发模型——I&F 模型[J]. 计算机科学, 2000, 27 (8): 47- 49.

A New Paradigm for the Development of
Database Centered Applications

YU Jir-shan, ZHU Yong-hui

(College of Information Science and Engineering, Huaqiao University, Quanzhou 362021, China)

Abstract: A new paradigm for the development of database centered applications is proposed by fusion of multiple new software concepts and technologies such as architecture, design pattern, GUI (graphical user interface) orientation, reuse and software component, evolution, etc. By building a GUI—driven components repository for essential operating behavior, which are also addressed in the paper, the paradigm provides for the development of database centered applications a flexible, efficient, simple and easy to use method and facility which will satisfy every demand of end users including personalization, the requirements of special business logic and the adaptation to meet the changing and developing of business.

Keywords: information management; architecture; design pattern; software component; GUI oriented

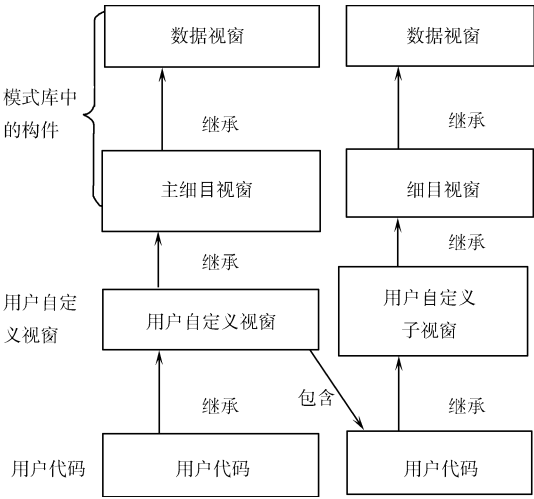


图 4 主细目式操作构件

Fig. 4 The component for master details style operation