多层结构的数据库应用系统的开发

VC6.0+CSocket+ADO+SQL server

付勇 著

附赠CD-ROM光盘

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

多层结构的数据库应用系统的开发

VC6.0+CSocket+ADO+SQL server

付勇 著

西安交通大学出版社
XI'AN JIAOTONG UNIVERSITY PRESS

内容简介

本书是结合作者十多年来使用 VC++6.0 从事数据库应用软件开发的经验，参照一个已经商品化的数据库应用软件为蓝本，用时一年多的时间撰写了初稿，又在其后两年中反复修改完成。全书六百多页，约七十万字，用一个完整的实例，以基本程序架构、应用软件服务器端到客户端为主线，用深入浅出、全面细致、循序渐进的方式介绍了多层次 C/S 架构的数据库应用系统开发的过程。该系统使用了 ADO 数据库访问技术、Socket 网络通信技术，并以 SQL Server 作为全局数据库，Access 作为本地数据库的方法，来提高数据访问的效率，其中还应用了事务处理技术、多线程技术、注册表访问技术、类的派生和封装技术等。特别值得一提的是，本书提供了十分详尽的查询报表、统计报表预览打印输出，导出到 Excel 以及报表等比缩放的编程技术和方法。最后还以一种全新的视角给出了帮助文件和安装程序的制作方法。

本书从商品化软件的角度出发，给出了目前多数 VC++数据库编程类图书鲜见的编程方法和核心技术，结合所提供的完整的源程序，成为一本不可多得的好书。

本书既适合具有一定 VC++编程基础的中、高级编程人员阅读，也可以作为计算机类本科生和研究生学习研究的参考书籍。

随书附赠光盘一张，提供书中实例的全部源代码、资源文件、制作帮助文件的所有源文件和相关工具等。

图书在版编目(CIP)数据

多层结构的数据库应用系统的开发/付 勇著. —西安：
西安交通大学出版社，2010.12
ISBN 978-7-5605-3780-1

Ⅰ.①多… Ⅱ.①付… Ⅲ.①数据库系统-系统开发
Ⅳ.①TP311.13

中国版本图书馆 CIP 数据核字(2010)第 229807 号

书　　名　多层结构的数据库应用系统的开发
——VC6.0+CSocket+ADO+SQL server
著　　者　付　勇
责任编辑　王　欣　刘雅洁

出版发行　西安交通大学出版社
(西安市兴庆南路 10 号　邮政编码 710049)
网　　址　http://www.xjtupress.com
电　　话　(029)82668357　82667874(发行中心)
(029)82668315　82669096(总编办)
传　　真　(029)82668280
印　　刷　西安交通大学印刷厂

开　　本　880mm×1230mm　1/16　**印张** 39.75　**字数** 1183 千字
版次印次　2010 年 12 月第 1 版　2010 年 12 月第 1 次印刷
书　　号　ISBN 978-7-5605-3780-1/TP·537
定　　价　69.00 元(附赠 CD-ROM 光盘一张)

读者购书、书店添货、如发现印装质量问题，请与本社发行中心联系、调换。
订购热线：(029)82665248　(029)82665249
投稿热线：(029)82664954
读者信箱：jdlgy@yahoo.cn

前　言

目前，关于使用 VC++编写数据库应用系统的书籍非常多，然而，大多数的书籍都是教科书式的，罗列了许多基本的编程概念、设计理论、SQL 语言、数据库编程方法等，最后给出若干个编程实例，少则三五个，多则十几个。一本书给出这么多的实例，都只能泛泛而谈，很难给出具有一定深度的、细致入微的、具有商品化应用软件水平的内容来。

本书是结合作者十多年来使用 VC++6.0 从事数据库应用软件开发的经验，并以一个已经商品化的数据库应用软件为蓝本，用时一年多的时间撰写完成的。原稿约一千页，近百万字，在其后的两年间，又进行了多次修改，将全书压缩到六百多页，约七十万字。这本书用一个完整的实例，以基本程序架构、应用软件服务器端到客户端为主线，用深入浅出、全面细致、循序渐进的方式介绍了多层次 C/S 架构的数据库应用系统开发的过程。该系统使用了 ADO 数据库访问技术、Socket 网络通信技术，并以 SQL Server 作为全局数据库、Access 作为本地数据库的形式，来提高数据访问的效率，其中还应用了事务处理技术、多线程技术、注册表访问技术、类的派生和封装技术等。特别值得一提的是，本书提供了十分详尽的查询报表、统计报表预览打印输出、导出到 Excel 以及报表等比缩放的编程技术和方法。最后还以一种全新的方式给出了帮助文件和安装程序的制作方法。

第 1 章为本书的概括性说明，给出了本书实例软件开发环境的选择和数据库访问技术选择的原因和理由，通过几点说明介绍了本书的写作思想和方法，并对应用实例的体系架构、多层模式等作了简要说明。

第 2 章详细介绍了编程实例的体系结构以及该实例所采用的用户界面层、界面适配层、网络通信层、业务逻辑层、访问适配层以及数据访问层等多层结构的作用和目的。最后给出了与团队开发人员相对应的层次结构和配置，从而使读者从较高的层次了解和掌握实例的编程思想和方法，为后续的阅读和练习提供了一个清晰的思路。

第 3 章到第 11 章用 9 章的篇幅介绍了应用软件基本框架的编程过程，其中包括了 5 个自定义控件的编程，自定义对话框基类的建立，数据访问层、界面适配层、访问适配层、业务逻辑层的编程，以及软件启动过程的控制、主窗口的控制的编程。这里涉及窗口分割技术、注册表访问技术、ADO 数据访问技术、类的派生和封装技术、多线程编程技术以及应用程序模块的组装和控制技术。特别是软件启动过程的控制、自定义对话框基类的应用、ADO 智能指针的封装以及用于从界面层到数据访问层数据交换的自定义字符串数组 CSbtStrArr 类的巧妙运用，给出了一种全新的编程思想和方法，值得读者认真学习和研究。

第 12 章给出了在 VC++6.0 中共享编程资源和编程代码的方法和编程思路。

第 13 章到第 17 章用 5 章的篇幅介绍了应用软件服务器端的编程过程，包括服务器端编程环境的建立，应用服务器的基本编程，服务器端 CSocket 的编程以及在服务器端进行数据的备份、还原、通信端口的设置、用户 IP 的控制、操作日志的管理等内容。利用 CSocket 通信技术进行数据库应用软件的编程，多用户访问控制技术以及在应用服务器端进行数据库业务逻辑处理的方法应该是读者学习的重点。

第 18 章到第 30 章用 13 章的篇幅介绍了应用软件客户端的编程过程，包括客户端编程环境的建立，客户端 CSocket 的编程，客户端的基本编程和其他有关内容的编程等。其中，客户端功能选择，单位、所属部门和人员选择的控制方法，预览、打印和等比变换，Excel 控制技术，查询报表、统计报表的生成以及在数据处理中采用的事务处理技术等都给人以面目一新的感觉。尤其值得一提的是，本书全面系统地介绍了预览打印输出中页面纸张大小、方向的控制，等比变换的方法，普通一览表、主从表、双行一览表、四行一览表以及统计报表控制技术，Excel 导出控制功能的编程也有独到之处。

第 31 章以应用服务器端为例，介绍了应用软件帮助系统的建立和编程。在帮助页面的建立中从应用软件的视角出发，具有简洁清晰的特点。在帮助文件的建立中重点介绍了目录的组织和索引的建立。最

后，介绍了在应用软件主窗口帮助菜单中，帮助目录和帮助索引的调用方法以及在对话框中相关帮助页的调用方法。

第32章给出了安装程序的建立方法。其中，共享文件的处理，在多种客户端配套软件共享信息时的处理，可覆盖文件和不可覆盖文件的处理，安装界面、安装组件、文件组的组织和安装目的地的控制以及安装资源的设置等，都有独到之处。

本书采用循序渐进的方式叙述了一个应用服务器和客户端的开发编程的全过程，适合具有一定VC++编程基础的中、高级编程人员阅读，初学VC++的读者只要认真按照书中介绍的步骤去做，同样能够完成。本书也可以作为计算机类本科生和研究生学习研究的参考书籍。

本书附赠的光盘中还向读者提供应用实例的全部源代码，帮助文件的全部源代码和一个专用的压缩、解压工具，以便读者在学习中参考和使用。

作者

2010年9月

目　录

第1章 概 述

1.1 引 言

嗨，你好！欢迎你阅读这本书。随着这本书提供的线索，让我们一起进入到多层结构数据库应用系统开发的神奇世界，探索那奇妙无穷的奥秘吧。相信我是一位好向导，一定会带你领略多层结构数据库应用系统开发中那些深奥、奇妙、神秘的东西，探索那些鲜为人知的地方，挖掘出丰富的知识宝藏。认真读完这本书，并按书中的要求去做，就会感受到它带给你的满足、充实和骄傲，因为你也亲手编制了一个具有商品化水准的网络数据库应用系统。

到自然界去探险，需要装备，如粮食、工具和武器，还要学习探险知识和进行必要的训练。探索开发多层结构数据库应用系统的奥秘，也同样需要做一些类似的工作。比如：你手头应有一台性能尚佳的计算机，安装了 VC++6.0、SQL Server 2000 和 Access 2000 数据库管理系统，以便我们能够编写 VC 程序代码和创建所需要的数据库。如果你还读过 VC 编程方面的书籍，有一些这方面的知识，懂得 SQL 语言，甚至还了解 ADO 和 COM+，你的探索旅程会更加顺利，否则就会艰难一些。但即使遇到困难也不必灰心，边看边学也成，这本书采取的循序渐进逐步扩充的学习方法，一定会给你提供必要的帮助和指导。只要在风浪中经受考验，“菜鸟”的翅膀终究会长硬的。当你战胜了重重困难，最终取得了胜利，成为一名编制多层结构网络数据库应用系统的行家里手时，你将会有一种无与伦比的成就感，何乐而不为呢？

1.2 软件开发环境的选择

在这里，关于软件代码的编制，我选择的集成开发环境是 Visual C++ 6.0。开发数据库应用软件，可以使用的开发工具是多种多样的，例如 C、VB、Borland C、C++ Builder、Power Builder、Delphi、Visual FoxPro 以及.NET 等等。选择 VC 一定有我自己的理由。

(1) 1983 年开始学习 BASIC 语言，1984 年使用 dBASE II，1986 年使用汇编语言开发电视字幕软件，1989 年使用 Borland C，1993 年开始使用 Quick C，1995 年起就开始使用微软的 Visual C++，版本从2.0、4.0、5.0、6.0 到 Visual C++ .NET。这期间，还使用过 Visual FoxPro、Power Builder、Delphi、C++ Builder 和 C# .NET 开发数据库应用软件。无疑，VC 是我使用时间最长、最为熟悉的开发环境了。俗话说“熟能生巧，巧能生精”。使用 VC 来讲解数据库应用系统的开发，奉献给你的一定是精彩。

(2) 在我看来，用 VC 编制出来的应用程序界面端庄、典雅、优美，更具专业气质。

(3) VC 与 Windows 操作系统结合是最为紧密的，编译出来的代码体积小、运行快、环境要求低。

(4) VC 为程序员提供了非常广阔的想象空间和多种多样的开发手段，非常适合于开发底层模块和特殊控件，这正是我们所需要的。如果仅仅使用现成的控件，是不可能编写出独具特色和风格的应用软件的。

(5) 用 VB 编写用户程序界面简单方便，Delphi 则在数据库应用方面独具魅力，而许多人认为用 VC 编写数据库应用程序很难，其实不然。究其原因，一是 VC 的数据库应用开发环境的提供时间较晚(VC 是从 4.0 版开始提供的)，熟悉的人较少；二是 VC 的数据库应用接口编程方式与其他软件不大相同，使习

惯于其他方式的编程人员感到不适应而已。其实,在数据库应用软件编程方面 VC 有后来居上之势,采用的技术更新,工作效率更高,运行速度更快,使用起来也十分方便。

(6) .NET 是近年来微软极力推广的新型软件开发集成环境,采用了许多新的技术,面貌令人感觉耳目一新。然而,需要明确的是:.NET 是为 Client/Server 和基于 Web 的应用程序设计的,它并不那么适用于桌面的应用程序;而且,新技术、新方案并非一年半载就能被通透地掌握和熟练地运用;此外,用 .NET编制的应用程序通常会显得肥大,对运行环境也要求较高。这也算是选择 VC 的一个理由吧。

操作系统选择的是 Windows 2000 Server,一个原因是主数据库管理系统要使用 SQL Server 2000,再一个是许多开发工具在 Windows 2000 环境下运行十分可靠,编译出来的应用程序在 Windows 2000、Windows Me、Windows XP 或 Vista 等操作系统环境中都能正常运行。而在 Window XP 环境下编译出来的应用程序总会出现问题,这大概是 VC++6.0 同 Window XP 之间兼容性不好的缘故。

数据库管理系统除了使用 SQL Server 2000 外,还要使用 Access 2000。SQL Server 2000 用于创建和管理数据库服务器中的主数据库,而 Access 2000 则是用于创建和管理客户机和应用服务器在本地使用的辅助数据库。

另外,需要为我们编制的这个数据库应用系统设计 HTML Help 风格的帮助文件,这就需要一个帮助文件制作工具和编译工具。帮助文件制作工具我选用的是 Macromedia Dreamweaver 8,编译工具选择的是 HTML Help Workshop 4.74。而安装软件制作工具使用的是 InstallShield 6.22,图像制作工具使用的是 Adobe Photoshop 7.0,抓图工具使用的是 SnagIt 7。这些工具完全可以根据自己的使用习惯来选择。

1.3 数据库访问技术的选择

在 Visual C++下开发数据库应用程序可以使用的数据库访问技术是多种多样的,这些技术各具特点,提供了多种简单、灵活、访问速度快和扩展性好的数据库访问途经。Visual C++提供的数据库访问技术主要有以下几种:

- ODBC API(Open DataBase Connectivity API)
- MFC ODBC(Microsoft Foundation Classes ODBC)
- DAO(Data Access Object)
- OLE DB(Object Linking and Embedding DataBase)
- ADO(ActiveX Data Object)

ODBC API 数据库访问技术是一种传统的数据库访问技术,它提供了一组对数据库访问的标准 API(应用程序编程接口)。一个基于 ODBC 的应用程序对数据库的操作不依赖任何数据库管理系统(DBMS),不直接与 DBMS 打交道,所有的数据库操作由对应 DBMS 的 ODBC 驱动程序完成。它的最大优点是能以统一的方式处理所有的数据库。ODBC API 是一种底层的访问技术,编程难度较大,此外还需要专门配置数据源。

MFC ODBC 是 VC++中对 ODBC 进行了封装的 MS 基础类库,用以简化对 ODBC API 的调用,从而使 ODBC 编程的复杂性大大降低。

DAO 是一组 Microsoft Jet 数据库引擎的 COM 自动化接口。DAO 通常用来访问本地桌面数据库资源,如 Microsoft Access、Microsoft FoxPro、Paradox 等,也可用来访问远程数据库资源。DAO 对 Microsoft Access 的访问是最为快捷的,因此一般用来访问 Access 数据库。

OLE DB 是一套通过 COM(组件对象模型)访问数据的 ActiveX 接口。它是一种底层数据访问技术,为各种应用程序提供了最佳的访问接口,具有应用灵活、网络通信量小、占用内存少、访问速度快等特点,而且还能支持一些不规则数据系统。OLE DB 的缺点是编程代码量大,对底层的操作比较复杂。

ADO 是以 OLE DB 为基础的应用级编程接口,是对 OLE DB 的封装。它继承了 OLE DB 应用灵活、

网络通信量小、占用内存少、访问速度快等优点，具有良好的兼容性和扩展性，支持事务处理和存储过程，在使用上也十分简单方便。

显然，从上述各种数据库访问技术的特点来看，ADO 应该是我们数据库访问技术的第一选择。

1.4 几点说明

这本书的根本目的是引导读者去探索多层结构的数据库应用系统的编程方法和相关技术的奥秘，通过实际编程的过程，让读者去体味，去理解。因此，这本书不打算写成 VC++的教科书，也不打算作很深的理论探讨，而是更注重于实际。编制程序的过程中对所需要的基本知识以及相关的原理也会做一些介绍和说明，在这里够用即可。如果读者想要深入研究一下有关问题，那就再找一两本有关的书籍读读。

这本书中的编程实例，实际上是笔者主持研发的一个商品化的面向机关事业单位的人事信息管理系统的提炼和简化，更加玲珑精巧，是一个实用的信息管理系统。

作者在本书实例的编写和开发过程中，于浩瀚的 VC 代码资源中发现了许多极具使用价值的程序模块和控件代码，经过部分修改和调整，应用到了本程序实例当中。其中有来源或作者姓名的，都原封不动保留了相关信息，当然更多的是根本不知作者姓名的情况，对于他们，只能给于衷心的感谢和诚挚的敬意。

编写本书时，并没有假设读者是一位富有经验的 VC++软件编程高手，因此，开始阶段会循序渐进，较为细致地介绍软件的编程过程，设计的思想方法，以便使初学者也能比较轻松地进入到软件开发的境界之中。随着开发过程的逐步推进，将逐渐简化一般过程，更加突出相关问题的核心内容。从第 3 章起，我们就从应用程序界面的设计入手，构建一个初步的程序框架，在以后的编制过程中，像滚雪球似的逐步扩充和完善。这也正是软件工程学中螺旋形开发模型和渐增式开发模型的一种体现。

在程序设计和代码编写过程中，我们会遇到设计原理，注意事项，基本原则，编程技巧，以及诸如此类的问题，本书中将以“注意”、“原则”、“技巧”这样的条目加以说明和解释，使这样的内容以鲜明的实例形式逐步融入到读者的思想之中。

包含表示层、业务逻辑层和数据库访问层的三层结构是数据库应用系统的一种典型结构。在 C/S 架构下，基于图形用户界面的数据库应用系统，客户机端通常包括表示层和业务逻辑层，业务逻辑层通过 ADO 这样的数据库访问接口，利用 ADO 内在的网络通信功能到服务器端进行数据库的访问。这种结构就是所谓的“胖客户机”。本书打算在三层结构的体系上，在客户机端与服务器端之间增加一个由 CSocket 构成的网络通信层，把业务逻辑放在服务器端，这就形成了多层结构。客户机端只有表示层、全局业务逻辑接口和网络通信接口，这种结构就是“瘦客户机”。本书将突出多层结构的数据库应用系统的设计，在各种层次上，能分解为一个新的层次的就尽量分解，使各层更为鲜明。例如，为了降低表示层与业务逻辑层之间的耦合度，可以在其间增加一层业务逻辑适配层（有些书称之为代理层），这样分解的结果，将会使层次增多（多于四层），但每个层面结构更为简洁，逻辑更为清晰，具有更好的数据独立性。然而，层次的增多，必然会导致数据多次的接力传递，从而使执行效率所有降低（事实上并不明显）。因此，在实际应用中，应该根据程序的规模、数据传递量、运行的时间效率等多种因素，综合考虑软件层次结构的划分。大型软件的开发，需要一个具有不同层次人员的开发团队，多层结构的设计由于适配层的隔离作用，反而更易于组织、管理、开发、测试和组装。例如，你可以安排新手进行程序界面（对话框）的实现，让有一定经验的程序员进行界面编程，让熟悉业务逻辑和数据库访问接口的软件工程师进行业务逻辑的编程，不同层次的软件测试人员进行不同层次的产品测试，而高级软件工程师则进行软件的组装以及组织组装测试和总体测试等工作，其优点非常多。

另外，在数据库的操作中，本书中提供的实例没有使用存储过程。这是由于某些中小型数据库管理系统不提供存储过程，考虑代码的通用性，也就不采用存储过程的调用方式了。

想说明的问题太多，还是就此打住吧！如果希望同我联系，请发 E-Mail 到：TLFUYONG@263.net。

第 2 章　编程实例的体系结构

2.1　体系结构图

俗语说得好:站得高看得远。在开始探索之旅之前,先纵览全局,了解行程路线,掌握探索过程中的任务、目的和要点,一定会事半功倍的。图 2-1 就是本书要实现的编程实例的体系结构图,也是该程序基本结构的蓝图。

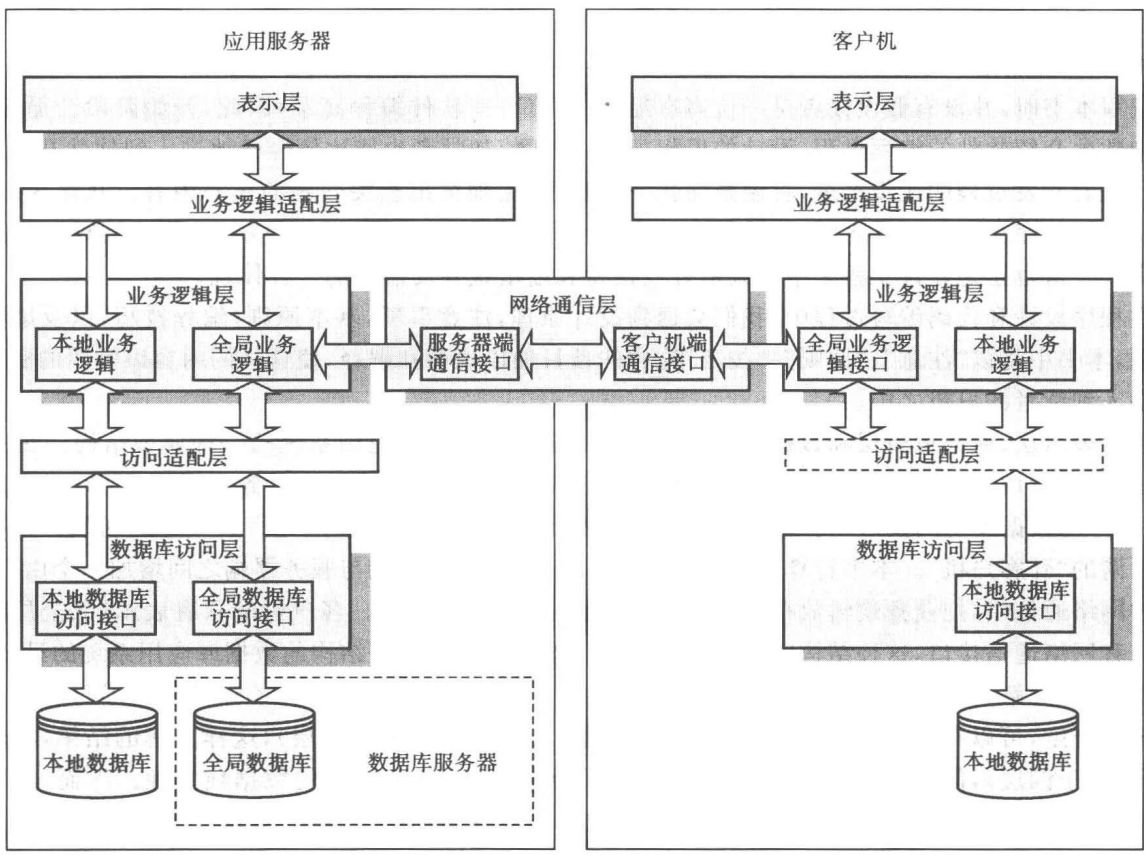

图 2-1　C/S 架构下多层结构的数据库应用系统的体系结构图

从图 2-1 中我们可以看出,无论是应用服务器端,还是客户机端都具有表示层、业务逻辑层和数据访问层这三个基本的层面。在客户机和应用服务器之间还增加了一个网络通信层,这样,从客户机到全局数据库之间,就具有五个基本结构层。细心的读者还会发现,在表示层和业务逻辑层以及业务逻辑层和数据库访问层之间还多了两层结构,一是业务逻辑适配层,二是访问适配层。这样,从客户机到全局数据库之间就有了七层结构(是不是有点吃惊)。此外,无论是应用服务器端还是客户机端,都有一个本地数据库以及与之相关的本地业务逻辑和数据访问接口。你一定会问:“为什么要这样呢?”这个问题我们留待后面再作解释。

2.2　软件的层次结构

2.2.1　表示层

表示层是用户与计算机交互的通道。用户的输入和计算机的输出通常都是在这里体现的。用户输入的信息或命令，往往是业务逻辑处理过程的输入数据，而业务逻辑产生的结果要通过表示层传递给用户或其他输出设备。对此恐怕大家都是非常清楚的，不必多说。

表示层是由各种各样的诸如图形、图片、文字、声音、窗口、对话框、按钮、列表框、文本框等编程资源和与之对应的应用程序类、主窗口框架类、文档类，以及与各种业务对应的一系列对话框类和其他辅助类构成的。

2.2.2　业务逻辑适配层

业务逻辑适配层显然不是必需的。但对于表示层而言，如果有了业务逻辑适配层，就会有许多好处和便利。对于业务逻辑适配层的问题，我们从客户端的情况来讨论更具有典型性。用户从用户界面输入的信息和指令，都是需要传递给业务逻辑进行加工处理的。如果没有业务逻辑适配层，表示层会直接将信息和命令传递给本地业务处理逻辑接口，或者通过网络通信接口传递给应用服务器端的业务处理逻辑。这样，表示层就需要为不同的连接形式编写不同的代码，业务逻辑或通信接口上的一些变化，势必影响表示层的表示。一个应用程序的输出窗口或对话框少则几个，多则几十个。业务逻辑或通信接口上的微小变化，对表示层的影响可就大了，真有“牵一发动全身”的感觉。如果用软件工程学的专业术语来说，那就是层面之间的耦合度太高，各个层面的独立性太差。如果有了业务逻辑适配层，表示层与底层的接口代码只针对业务逻辑适配层即可，形式可以统一，而且业务逻辑适配层中对应的接口只需要一套就够了，而不是几套、十几套甚至几十套。业务逻辑或通信接口发生了改变，只需要修改业务逻辑适配层的代码就可以了，表示层则通常不需要改动或改动很小。

说到这，我不禁联想到电源插座板。插座板上有不同类型的插座，用来连接不同形式的插头。如果连接设备的插头形式上发生了较大的变化，换一个适应新形式插头的插座板就解决问题了。看来，界面适配器的作用和插座板的作用用“异曲同工”来形容还是十分恰当的。

业务逻辑适配层是用一个自定义的一般类(Generic Class)来实现的。

2.2.3　网络通信层

网络通信层用来建立网络环境中不同计算机之间的连接。计算机之间通过某种网络协议进行数据的传递，例如 TCP/IP(传输控制协议/网际协议)、UDP(用户数据包协议)、FTP(文件传输协议)、SMTP(简单邮件传输协议)等。在我们这个应用实例中，网络通信层用于局域网环境中的客户机与应用服务器之间的通信联络和数据传递。而网络通信接口我们采用的是 VC++中基于 TCP/IP 协议的 CSocket 类。在数据库应用环境中客户机和应用服务器双方还要对业务进行沟通，所以需要在 TCP/IP 协议的基础上再加一层应用层，即业务命令协议层。客户机通过网络通信层向应用服务器端的业务逻辑层发出业务请求，服务器端的业务逻辑在完成业务处理后再通过网络通信层向客户机反馈信息。

在我们这个应用实例中，网络通信类是由 MFC 中 CSocket 类的派生类组成的。有关这方面的内容很多，也是我们这套体系结构中的重要组成部分，后面将有专门的章节进行详细介绍。

2.2.4　业务逻辑层

业务逻辑层是由处理业务逻辑的类组成的。在该类中，有一系列用于处理各种业务逻辑的函数。可以把各种业务逻辑的处理放在同一个类或几个类中，构成业务逻辑层，使程序的结构更为清晰明了，同时

也便于进一步的扩充。如果一个应用软件需要扩充软件功能，增加新的业务逻辑，只需要添加相应的表示层模块和业务逻辑处理函数即可。由于业务逻辑适配层和访问适配层的存在，不会影响到其他层面。

业务逻辑层也是由自定义的一个或多个一般类构成的。

2.2.5 访问适配层

访问适配层与业务逻辑适配层一样，也不是必需的。与业务逻辑适配层的作用相似，它起到了隔离业务逻辑层和数据访问层的作用。无论是业务逻辑层发生了改变(例如增加了新的业务处理功能)，还是数据访问接口发生了改变(例如不采用 ADO 而直接使用 OLE DB)，只需对访问适配层加以调整就可以了。访问适配层可以大大改善业务逻辑层和数据访问层的独立性，降低耦合度，从而大大提高软件的性能和质量，为软件的维护、改进和扩展提供了有利的条件。因此，在我们的这个实例中还是选择使用访问适配层。

客户端的局部业务逻辑和本地数据库访问接口之间的关系是比较简单的，似乎不加访问适配层更加简洁方便，其实不然。现在的软件编程讲究的是统一、规范。类似的业务逻辑，类似的数据访问，采用不同的连接方式，给编程造成了负担，需要考虑多种情况，采用不同的代码形式，这无形中增加了差错率，也增加了二义性的可能，从而影响软件的质量。同时还会影响软件测试、功能分析等一系列开发进程，对以后的软件维护产生不利的影响。

访问适配层也是由自定义的一个或多个一般类构成的。

2.2.6 数据访问层

数据访问层可以由一种或多种数据访问接口构成。常见的数据访问接口包括 ODBC API、MFC ODBC、DAO、OLE DB、ADO 等。通常情况下都只选一种，我们这个系统选择了 ADO。数据访问层可以直接使用某种数据访问接口，也可以使用数据接口类的派生类或封装类。

2.3 全局数据库与本地数据库

一个网络数据库应用软件，一定有一个用于读取和存放核心数据的、可以由多个客户机应用程序共享的数据库。在本实例中，这个数据库就是放在应用服务器上的全局数据库。由于数据库访问接口本身就具有网络通信功能，因此，全局数据库既可以放在安装了应用服务器软件的同一台计算机上，也可以放在另一台通过网络连接的计算机上。安装了服务器端数据库应用软件、能给客户机提供相关业务逻辑服务的计算机称为应用服务器；存放全局数据库的计算机被称为数据库服务器。我们这个实例中，应用服务器和数据库服务器是同一台计算机。

从图 2-1 中可以看出，无论服务器端还是客户机端都有一个本地数据库。本地数据库并非必需，其中的相关数据可以建立在全局数据库中。一个网络数据库应用软件，总有一些数据只与服务器有关或只与客户机有关，也有一些数据是持久性数据，极少发生更替和变化。例如，用于记录服务器提供服务的日志数据，通常是由服务器进行管理的；客户机也有需要自己管理的数据，不需要与服务器进行通信；在对人事信息进行管理的系统中，籍贯、学历、学位、职务名称等列表信息都具有持久性，可以在客户机中存放一个副本，这样就不必通过网络通信来获取，既减少了网络通信的负担，也提高了系统的整体效率。因此，在本地建立一个或多个辅助数据库就是为了解决这样的问题。

2.4 编程人员的层次与任务

多层体系结构的软件设计为软件编程提供了一个非常好的思路，创造了层次分解和模块分解的有利

条件，给不同能力层次的软件编程人员提供了不同的开发环境和任务，且将相互的影响降低到了一个很低的程度。一个具有一定规模的应用软件的开发，通常是由一个开发队伍共同来完成的。开发队伍中包括从事系统设计、文档编写、界面设计、数据库设计、软件编程、模块测试、系统测试组装等不同任务的成员，其中软件编程人员也有以下几种不同的层次：

- 初涉软件开发的“菜鸟”
- 有初步开发经验的程序员
- 有一定开发经验的软件工程师
- 有较丰富开发经验的高级软件工程师

当一个软件的基本框架构建好之后，主要的编程任务可以分为以下几个层次：

- 软件界面制作（最主要的是对话框制作）　　难易程度：★
- 界面程序模块编程（表示层和业务逻辑适配层）　　难易程度：★★
- 业务逻辑的编程（业务逻辑适配层、业务逻辑层和访问适配层）　　难易程度：★★★
- 模块的组装测试　　难易程度：★★★☆

由于采用多层结构设计，不同的编程任务之间有了明显的分界，因此可以根据编程人员的能力分配编程任务。

2.4.1　软件界面的制作和人员配置

软件界面制作人员的主要任务是根据设计要求以及界面设计规范进行以下几项工作：

（1）界面图片、图标等界面元素的制作；

（2）制作对话框以及添加对话框中的控件；

（3）按要求填制对话框以及对话框控件的 ID；

（4）调整对话框中控件的 Tab（跳表）顺序。

这项任务属于表示层，内容比较简单，但工作量较大，可以分配给初涉软件开发的“菜鸟”来完成。但界面图片、图标的制作则需要具有美术专业技能的人员来完成。

2.4.2　软件界面的编程和人员配置

软件界面的编程人员主要任务是检测软件界面制作人员提交的对话框，根据设计要求进行对话框类以及相关类的编程，同时也涉及业务逻辑适配层，是表示层最主要的工作。如果业务逻辑适配层有适当的函数可以调用，则可直接调用；如果没有适当的函数可供调用，则要在业务逻辑适配层类中编写可用于测试的简单函数（直接返回一些期望值，称之为“桩”或“桩函数”）。这项任务的工作量较大，是由人数较多的初级程序员根据规定好的编程规范和编程样板（模板）来完成的。

2.4.3　业务逻辑的编程和人员配置

业务逻辑的编程是应用软件编程的核心任务，要根据不同的业务要求，实现业务逻辑层类中的相关编程，要求编程人员有较高的业务逻辑分析能力，熟悉数据库访问适配层的相关函数。其主要工作是检测程序员提交的软件界面的相关代码，进行业务逻辑函数的编写，同时负责业务逻辑适配层桩函数的更改。这项任务通常是由经验丰富的软件工程师来完成的。

2.4.4　系统组装的编程和人员配置

系统组装则是将程序员和软件工程师们提供的软件模块组装成一个应用系统。这是由软件开发机构的高级工程师（核心开发成员）负责完成的。

当然，这是理论上的层次分配，根据具体的情况，也可能会融合交叉。例如，界面的制作和界面的编程也可以融合在一起来完成。这是由软件的规模决定的，软件的规模越大，编程人员层次的划分越清晰，任务分工越单一。因此，这种分层体系结构的软件开发方式更适合于中、大型软件的开发。