

CAD

应用程序开发详解

——Visual C++与OpenGL综合应用

王清辉 李静蓉 编著



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>



提供完整源代码

CAD 应用程序开发详解

——Visual C++与 OpenGL 综合应用

王清辉 李静蓉 编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书系统地阐述了在 MFC 与 OpenGL API 的集成开发环境下,用面向对象的技术开发三维 CAD 软件的有关知识与方法,包括:总体程序框架的分析与设计、功能模块的划分,以及相关 DLL 库的开发与使用、CAD 基础几何类库的开发、在 MFC 环境下使用 OpenGL 进行图形绘制、开发面向 CAD 应用程序的 OpenGL 通用绘图类、使用面向对象技术设计 CAD 软件的几何内核、CAD 软件的图形交互、软件界面设计等。在介绍过程中,以一个完整的三维 CAD 软件(STLViewer)为例,将其开发过程贯穿于全书各章节,并附有完整的 Visual C++实现代码。全书面向开发实例进行分析与介绍,讲解透彻、易于理解。读者通过对本书的阅读和学习,能够掌握使用 Visual C++进行具有一定复杂程度的软件的设计与实现方法。

本书适合从事图形和 CAD 软件开发的技术人员,以及具有一定 Visual C++基础的大专院校相关专业学生阅读。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

CAD 应用程序开发详解: Visual C++与 OpenGL 综合应用/王清辉,李静蓉编著. —北京:电子工业出版社, 2012.9

ISBN 978-7-121-17970-9

I. ①C… II. ①王… ②李… III. ①AutoCAD 软件—程序开发工具②C 语言—程序设计③图形软件—程序设计 IV. ①TP391.72

中国版本图书馆 CIP 数据核字(2012)第 194255 号

策划编辑:陈韦凯

责任编辑:陈韦凯 特约编辑:蒲 玥

印 刷:三河市双峰印刷装订有限公司

装 订:三河市双峰印刷装订有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本:787×1092 1/16 印张:27 字数:691 千字

印 次:2012 年 9 月第 1 次印刷

印 数:4 000 册 定价:58.00 元(含光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010) 88254888。

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

前 言

笔者长期从事计算机辅助设计 (CAD) 系统的研究与开发工作, 将自己从事 CAD 系统设计与开发的经验提炼和总结并汇集成书, 一直是笔者的一个心愿。2003 年, 编著了《Visual C++ CAD 应用程序开发技术》一书。该书出版后, 读者发来电邮 100 余封, 给予了积极评价, 并交流了在各自专业领域从事 CAD 开发的一些技术问题。同时, 该书所附带光盘中的程序也在国内软件开发的主要论坛被广泛转载, 成为国内 OpenGL 及 CAD 开发的主要参考资料之一, 并被国内科技论文作为参考文献广泛引用。

随着近年来 CAD 技术的发展, 以及近年来与读者的交流与体验, 笔者感觉有必要在原书的基础上, 对该书的内容进行重要的扩充与修改, 并增加更多新的内容。恰逢电子工业出版社来我校组织约稿, 遂将上述想法付诸实施。

CAD 软件开发是工程软件开发中的一个重要领域。开发一个三维 CAD 软件所涉及的知识面很多, 如何规划和展开软件的开发工作是系统开发成功与否的关键问题之一。目前, 单纯讲解 OpenGL 技术、Visual C++ 使用, 以及面向对象的编程技术的出版物较多, 本书的特点在于, 从系统开发的角度, 将开发一个基于 OpenGL 显示的三维 CAD 软件所涉及的各环节逐步展现给读者, 将计算机图形学、OpenGL、面向对象的软件开发技术三个方面的知识予以融会贯通, 将许多技术难点、抽象的概念融入具体的开发实例中讲解, 着重讲解系统开发过程中的实际问题, 易于读者领会, 具有新颖性和强的实用性。在讲解中, 以一个三维 CAD 原形系统 STLViewer 的开发为主线, 贯穿全书, 并提供详细的代码注解。

Visual C++ 是 Microsoft 公司迄今开发的功能最为强大的软件开发工具, 是新一代 CAD 软件的主要开发平台。随着面向对象程序设计技术广泛应用、Visual C++ 优秀的开发环境、Microsoft Foundation Class (MFC) 以及 MFC 程序框架、Windows 操作系统对 OpenGL 的支持等, 为在 Windows 系统上开发三维 CAD 软件提供了极大的方便。读者通过本书的学习, 将能够循序渐进地了解和掌握使用 Visual C++ 开发三维图形及 CAD 应用软件的相关技术。通过对应用实例的具体剖析, 希望能使读者从软件开发的思维方法上对面向对象的程序设计技术有更深入的了解。

本书内容共分 10 章, 分别介绍基于 MFC 的总体程序框架的分析与设计; 功能模块的划分以及相关 DLL 库的开发与使用; CAD 的基础几何类库的开发; 在 MFC 环境下使用 OpenGL 进行图形绘制; 开发面向 CAD 应用程序的 OpenGL 通用绘图类; 使用面向对象技术设计 CAD 软件的几何内核; 曲面的数学描述及 OpenGL 绘制; CAD 软件的图形交互; 及软件界面设计等。

本书所附带的光盘中, 根据每章讲述的内容提供了全部的实现代码。本书的代码适用于 Visual C++ 2005 及以上版本。

本书的内容是作者长期从事 CAD 系统设计和开发的一些经验的提炼和总结。在内容或

方法上若有疏漏和不妥之处，恳请各位读者给予指正。

参加本书编写工作的还有华南理工大学的熊巍、阎汉生、黄仲辉、张小明，在此向他们表示感谢。

作者

目 录

第 1 章 三维 CAD 软件的系统框架分析.....	1
1.1 关于微软基础类 MFC	1
1.2 MFC 应用程序的文档/视图结构	2
1.2.1 文档/视图结构概述	2
1.2.2 文档与多个视图的关联	3
1.2.3 文档模板及主要组成类	4
1.3 实例分析: 三维 CAD 模型浏览软件 STLViewer	7
1.4 STLViewer 的程序框架	8
1.5 STLViewer 中类的层次设计及软件模块结构划分	15
1.6 建立和使用动态链接库	16
1.6.1 动态链接库的基本概念	17
1.6.2 基于 MFC 的动态链接库	18
1.6.3 查看执行程序 EXE 与 DLL 库的层次关系	19
附: 本章相关程序使用说明	19
第 2 章 矢量计算工具与几何计算类库设计	21
2.1 矢量运算概述	21
2.1.1 点	21
2.1.2 矢量	22
2.1.3 齐次坐标与齐次变换矩阵	25
2.2 设计点、矢量和齐次变换矩阵类	26
2.2.1 点类 CPoint3D	27
2.2.2 矢量类 CVector3D	31
2.2.3 变换矩阵类 CMatrix3D	34
2.3 三维图形的几何变换	38
2.3.1 三维齐次变换矩阵	38
2.3.2 平移变换	38
2.3.3 旋转变换/绕空间任意轴的旋转变换函数的实现	39
2.3.4 几何缩放	42
2.3.5 对称变换/沿空间任意平面的对称变换函数的实现	43
2.4 矢量计算及其工具函数集设计	44
2.4.1 判断两个空间矢量是否平行	45
2.4.2 计算两个矢量的夹角	46
2.4.3 计算三角面片的面积	46

2.4.4	正交投影及点到直线之间的距离计算	47
2.4.5	直线求交	48
2.4.6	创建过平面三个点的圆	49
2.5	设计几何基本工具库 GeomCalc.dll	51
2.5.1	GeomCalc.dll 中的输出类与输出函数	51
2.5.2	创建几何基本工具库 GeomCalc.dll 的步骤	53
2.5.3	使用 GeomCalc.dll	54
2.6	有关源程序代码	55
2.6.1	文件 CadBase.h	55
2.6.2	文件 CadBase.cpp	64
2.6.3	文件 CadBase1.cpp	91
附:	本章相关程序使用说明	96
第 3 章	创建基于 MFC 的 OpenGL Windows 程序	97
3.1	OpenGL 概述	97
3.2	在 Windows 环境下使用 OpenGL	99
3.2.1	OpenGL 的函数库	99
3.2.2	获取与安装 OpenGL	99
3.2.3	OpenGL 与 GDI	100
3.2.4	渲染场境	101
3.2.5	像素格式	103
3.3	OpenGL MFC 应用程序创建实例	108
3.3.1	创建一个应用程序框架	109
3.3.2	修改视类 CGLView	111
3.3.3	使用 OpenGL 的双缓存技术为应用程序增加动画效果	119
3.4	程序清单	120
3.4.1	文件 GLView.h	120
3.4.2	文件 GLView.cpp	122
附:	本章相关程序使用说明	129
第 4 章	设计封装 OpenGL 功能的 C++ 类	130
4.1	封装 OpenGL 的 C++ 类的设计	130
4.2	照相机类 GCamera 的设计	131
4.2.1	视点坐标系和视图变换	132
4.2.2	投影变换与视景体	134
4.2.3	视口变换	135
4.2.4	设计照相机类 GCamera	138
4.3	类 COpenGLDC	141
4.4	修改类 CGLView	146

4.5	运行应用程序.....	150
4.6	源程序清单.....	150
4.6.1	类 GCamera 的声明代码.....	151
4.6.2	类 GCamera 的实现代码.....	152
4.6.3	类 COpenGLDC 的声明代码.....	154
4.6.4	类 COpenGLDC 的实现代码.....	155
4.6.5	类 CGLView 的声明代码.....	158
4.6.6	类 CGLView 的实现代码.....	159
附:	本章相关程序使用说明.....	162
第 5 章	设计基于 OpenGL 的 CAD 图形工具库	163
5.1	创建动态链接库 glContext.dll.....	163
5.2	照相机类 GCamera 的功能增强	164
5.2.1	选择典型的观察视图	166
5.2.2	景物平移	170
5.2.3	景物缩放	171
5.2.4	使用 OpenGL 的选择模式	172
5.3	模型真实感渲染与 OpenGL 光照设置.....	173
5.3.1	光照的组成	174
5.3.2	定义 OpenGL 的光源	174
5.3.3	物体材质	176
5.3.4	单面光照与双面光照	177
5.4	类 COpenGLDC 功能的增强	178
5.4.1	实现和 Windows 窗口的关联.....	182
5.4.2	有关光源的操作	182
5.4.3	有关颜色的操作	185
5.4.4	图形绘制函数	191
5.4.5	选择模式	197
5.5	增加类 CGLView 中的功能	198
5.6	glContext 类的输出和调用	202
5.7	源程序清单.....	202
5.7.1	文件 Camera.h (类 GCamera)	202
5.7.2	文件 Camera.cpp (类 GCamera)	204
5.7.3	文件 OpenGLDC.h (类 COpenGLDC、CGLView)	210
5.7.4	文件 OpenGLDC.cpp (类 COpenGLDC)	214
5.7.5	文件 GLView.cpp (类 CGLView)	225
附:	本章相关程序使用说明.....	228
第 6 章	设计几何内核模块.....	229
6.1	CAD 实体模型的常用表示方法	230

6.1.1	边界表示法与体素构造法简介	230
6.1.2	多边形网格实体建模的概念	231
6.1.3	立体光造型 (STL) 文件格式	233
6.2	几何对象类的设计	235
6.2.1	类的层次设计	235
6.2.2	几何对象基本类 CEntity	238
6.2.3	三角形面片对象类 CTriChip	241
6.2.4	STL 几何模型类 CSTLModel	243
6.2.5	高级几何模型类 CPart	247
6.3	串行化 (Serialize) 实现文档存取功能	250
6.3.1	为什么要使用串行化	250
6.3.2	CArchive 类	251
6.3.3	串行化类的设计步骤	252
6.3.4	CObArray 的 Serialize() 函数	253
6.3.5	应用程序的文档串行化实例剖析	253
6.4	利用多态性实现 CAD 模型的绘制	256
6.4.1	虚拟函数与多态性	256
6.4.2	利用多态性统一 CAD 元素的绘制接口	258
6.4.3	纯虚拟函数	259
6.4.4	实现 CPart 模型的 OpenGL 显示	260
6.5	建立几何内核库 GeomKernel.dll	262
6.6	程序清单	263
6.6.1	文件 Entity.h	263
6.6.2	文件 Entity.cpp	267
附:	本章相关程序使用说明	276

第 7 章 CAD 应用程序的模块化实现 277

7.1	网格模型可视化程序 STLViewer 的解决方案	278
7.1.1	网格模型可视化程序 STLViewer 的模块结构	278
7.1.2	创建应用程序的解决方案	278
7.1.3	为解决方案配置目录与路径	280
7.2	设计应用程序框架	283
7.2.1	增加界面资源	283
7.2.2	修改框架类 CMainFrame	284
7.2.3	修改文档类 CSTLViewerDoc	286
7.2.4	修改视图类 CSTLViewerView	287
7.3	运行 STLViewer.exe	293
7.3.1	输入 STL 模型	293
7.3.2	存储 STLViewer 自己的文档 (*.mdl)	293

7.3.3	模型显示缩放	294
7.3.4	使用键盘平移场景	294
7.3.5	模型视图切换	294
7.3.6	模型着色模式切换	294
7.4	源程序清单	295
7.4.1	文件 MainFrm.h	295
7.4.2	文件 MainFrm.cpp	296
7.4.3	文件 STLViewerDoc.h	299
7.4.4	文件 STLViewerDoc.cpp	300
7.4.5	文件 STLViewerView.h	302
7.4.6	文件 STLViewerView.cpp	304
附:	本章相关程序使用说明	309
第 8 章	曲线和曲面的 OpenGL 绘制及类设计	310
8.1	关于曲面造型技术	310
8.2	曲线和曲面的参数表示	312
8.2.1	曲线的非参数表示	312
8.2.2	曲线的参数表示	312
8.2.3	插值与逼近	313
8.3	曲线的样条描述	314
8.3.1	参数样条曲线的基本概念	314
8.3.2	Bezier 曲线、曲面的数学表示	315
8.3.3	使用 OpenGL 绘制 Bezier 曲线	316
8.3.4	使用 OpenGL 绘制 Bezier 曲面	319
8.3.5	NURBS 曲线的数学表示	321
8.3.6	使用 OpenGL 绘制 NURBS 曲线	322
8.3.7	使用 OpenGL 绘制 NURBS 曲面	326
8.4	在应用程序中集成曲线和曲面功能	330
8.4.1	添加 Bezier 曲线功能	330
8.4.2	添加 Bezier 曲面功能	336
8.4.3	添加 NURBS 曲线功能	338
8.4.4	添加 NURBS 曲面功能	341
8.5	程序清单	344
8.5.1	类 CBezierCurve 的源代码	344
8.5.2	类 CBezierSurface 的源代码	348
8.5.3	类 CNurbsCurve 的源代码	353
8.5.4	类 CNurbsSurface 的源代码	359
8.5.5	COpenGLDC 中新增的曲线和曲面绘制函数代码	365
附:	本章相关程序使用说明	370

第 9 章 CAD 应用程序的界面设计	371
9.1 STLViewer 的界面增强	371
9.2 工具栏的排列	372
9.3 使用快捷菜单	374
9.4 创建类似 Visual Studio 风格的浮动窗口	375
9.4.1 控制条与停靠栏	375
9.4.2 开发具有 Visual Studio 风格的浮动窗口	378
9.4.3 CTabCtrl 控件的功能增强	380
9.4.4 建立界面工具库 DockTool.dll	387
9.5 使用树型视图 CTreeView 显示和管理文档数据	387
9.5.1 树型视图与树型控件概述	388
9.5.2 在 STLViewer 中创建 CPartTreeView	388
9.5.3 树视图 CPartTreeView 与文档的关联/文档多视图	390
9.5.4 在树型控件中使用图标	392
9.5.5 使用树视图控件显示文档中几何模型的结构和属性	393
9.5.6 通过树视图控件对文档数据进行操作	395
附：本章相关程序使用说明	398
第 10 章 三维 CAD 图形交互功能的实现	399
10.1 使用 OpenGL 选择模式	400
10.1.1 OpenGL 的三种操作模式	400
10.1.2 使用选择模式	401
10.2 一个 OpenGL 选择模式的应用程序	406
10.3 OpenGL 的选择功能与 CAD 应用程序的集成	410
10.3.1 定义选择视景体/修改类 CCamera	411
10.3.2 对选择过程的操作/修改类 COpenGLDC	412
10.3.3 自动给对象命名/对类 CSTLModel 的修改	414
10.3.4 在 STLViewer 中调用拾取功能	415
10.3.5 运行程序	417
10.4 三维空间的视景交互	417
10.4.1 鼠标移动跟踪	417
10.4.2 OpenGL 空间与 Windows 窗口的坐标转换	419
附：本章相关程序使用说明	421
参考文献	422

第1章 三维 CAD 软件的系统框架分析

本章要点:

- 关于微软基础类库 MFC。
- MFC 应用程序的文档/视图结构概述。
- 三维 CAD 应用程序的模块化结构分析。
- 动态链接库的创建、使用、层次关系。

1.1 关于微软基础类 MFC

在 VisualC++2.0 以后的版本中, Microsoft 公司推出了 MFC (Microsoft Foundation Class) 类库。MFC 随微软 Visual C++ 开发平台发布, 以一系列 C++ 类集的形式提供给程序设计人员使用, 为程序员提供一个面向对象的 Windows 编程接口。MFC 的实质是提供了一种基于 Windows 系统的应用程序框架 (Application Framework), 该框架具有高效运行的特性, 也是一些资深的程序员认为无所不能的开发工具。使用 MFC 类库的好处是, 可以简化应用程序的开发, 从而缩短开发周期, 而且代码的可靠性和可重用性也大大提高。首先, MFC 提供了一个标准化的应用程序框架, 这样开发人员不必从头设计创建和管理一个标准 Windows 应用程序所需的所有代码, 而是“站在巨人肩膀上”, 从一个比较高的起点编程, 故节省了大量的时间; 其次, 它提供了大量的代码, 指导用户编程时实现某些技术和功能。对用户来说, 用 MFC 开发的最终应用程序具有标准的、熟悉的 Windows 界面, 这样的应用程序易学易用; 另外, 新的应用程序还能立即支持所有标准 Windows 特性, 而且是用普通的、明确定义的形式。很多读者可能已经有了使用 MFC 的经验, 并对 MFC 编程有了浓厚的兴趣。这里, 我们首先从分析 MFC 的应用程序框架开始, 分析如何采用面向对象的技术来设计 CAD 软件的程序框架和软件模块结构。

什么是应用程序框架? 一种定义是“提供一个一般应用程序所需要的全部面向对象的软件组件的集合”。应用程序框架设计的好坏与否, 直接决定了程序功能的实现难易程度和软件开发与维护的代价高低。任何一个应用程序从本质上来说都是对数据的操作, 因此, 一个好的程序框架结构就意味着对应用数据管理上的友好安全和处理上的简便通用。应用程序框架的实例很多, 工程应用的计算机辅助设计/制造/分析 (Computer Aided Design/Manufacturing/Engineering, CAD/CAM/CAE) 软件多采用文档与视图相结合的程序框架。本章将结合 MFC 提供的应用程序的文档/视图结构这一工程软件的常见框架进行阐述和分析。为了便于阐述, 在本书中, 作者将一个应用于快速原型制造系统 (Rapid Prototyping Manufacturing,

RPM) 的 CAD/CAM 软件作了大量简化, 去除了主要的专业功能, 使之成为一个浅显易懂的三维 CAD 示例软件贯穿全书, 整个软件系统和模块设计均采用了面向对象的编程技术。在本章中, 将具体分析这一三维 CAD 软件的模块化结构设计, 使读者对工程 CAD 软件的应用程序框架设计有一定的理解和认识; 同时, 本章也在例程中讲解了基于 MFC 的有关动态链接库的具体创建和使用过程, 以期加强读者的理解和实际开发能力。

以下的各章中, 通过循序渐进地介绍, 读者将可以对以下主要技术有所掌握:

- (1) 面向对象的编程技术。
- (2) 软件的结构及模块化设计。
- (3) 动态链接库的开发与应用。
- (4) 几何计算基础类库的开发。
- (5) CAD 软件的界面设计和交互操作。
- (6) CAD 系统的几何内核设计。
- (7) 基于 OpenGL 的 CAD 模型显示。
- (8) 开发面向 Windows 应用程序的 OpenGL C++ 类库等。

1.2 MFC 应用程序的文档/视图结构

1.2.1 文档/视图结构概述

MFC 中的文档/视图结构 (Document/View Architecture) 是 MFC 的精髓, 它的基本思想是实现模型数据与显示交互的相互分离。文档 (Document) 负责管理模型, 视图 (View) 负责模型的显示与交互操作。文档/视图结构通过将模型和视图的分开表示, 提供了很好的数据层次和表现层次的解耦。在具体实现上, MFC 提供了两种典型且实用的基于文档与视图的应用程序框架模板, 按照其应用程序生成向导的引导步骤 (MFC AppWizard) 就可以创建所需的基于文档/视图结构的 MFC 应用程序框架。在此框架的基础上, 设计和插入相关的对象, 就可以实现交互式的用户界面、几何模型的管理和操作、图形图像的显示, 以及其他各种专业功能。

在 MFC 的文档/视图结构的应用程序框架中, 文档类和视图类是成对出现的。文档用于管理应用程序的模型, 而模型则封装了数据并提供对数据的操作接口; 视图用于显示文档中的模型, 并接收用户请求处理与用户的交互信息。模型是独立的; 而视图依赖于模型, 它从模型获取数据进行显示, 并向模型发送用户请求, 根据返回结果刷新视图显示。MFC 通过文档类和视图类的划分, 使数据的存储和显示既相对独立, 又相互关联。

在 MFC 所提供的框架结构中, 文档与视图的关系可以由图 1-1 简要表示。如图 1-1 所示, MFC 中的视图和文档是由视图类 (CView class) 和文档类 (CDocument class) 分别表示的。视图类可以调用其本身的成员函数 GetDocument(), 获得一个指向文档类的指针, 从而能够访问文档类中的数据。例如, 在视图类的 OnDraw() 函数中, 视图类通过调用 GetDocument() 函数获得一个指向文档类的指针。然后, 通过这个指针获取文档类中的数

据, 并使用 CDC 类 (负责处理应用程序显示设备接口的 MFC 类) 中的函数将这些数据绘制在视图窗口中。视图可以通过图形、图像、文字、表格等多种方式以视图对象 (CView object) 来显示实际文档 (文档对象 CDocument object) 中的数据。同时, 视图对象也负责接收鼠标、键盘等用户输入信息, 并通过这些与用户之间的交互信息来操作和修改文档中的数据。

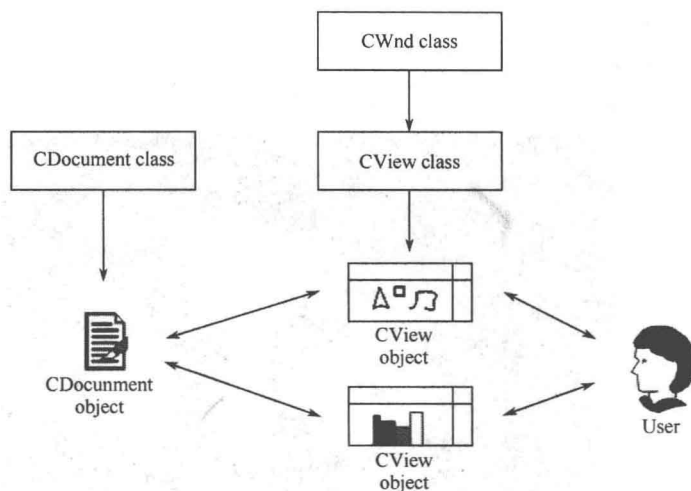


图 1-1 文档与视图的关系

1.2.2 文档与多个视图的关联

使用 MFC AppWizard 自动创建的 MFC 应用程序中, 为每个文档类的对象创建并关联了一个视图类的对象。实际上, 文档和视图的关系可以是一对一的, 也可以是一对多的, 即一个文档可以关联一个或一个以上的视图, 也即可以用多个视图来显示和操作文档中的模型数据。通常, 设计不同的视图是为了以不同的方式来显示文档内容, 如图形、图表、结构、文字等。然而, 当需要用多个视图显示同一个模型时, 情况发生了变化: 当一个视图修改了模型数据以后, 不但本身要刷新显示, 其他所有视图也要刷新显示。如果由该视图通知其他视图, 它就需要能够找到其他所有视图。每个视图都可能发生修改, 这样每个视图都要找到其他所有视图, 这种关联过于复杂, 不但难以维护, 而且不便于增加新的视图。用 MFC 提供的文档与视图的关联模式, 则可以很好地解决上述问题, 从而实现由模型通知视图更新, 而模型不依赖于具体的视图, 而且不同视图之间保持相互独立。

如图 1-1 所示, 一个文档对象就同时关联了多个视图对象。当在一个视图对象中对文档数据作了修改后, 可以调用文档类的 UpdateAllViews() 函数来更新所有与文档相关联的视图类对象的显示, 以此来保持所有视图相对于同一文档数据变化的同步显示。一个文档关联多个视图为应用程序提供了很多的方便。例如, 程序中使用两个视图, 分别以表格和图形的方式来显示文档中的数据, 图形显示用户直观的感觉, 而用户可以通过表格来访问和操作文档中的具体数据, 这种方式在 CAD 应用程序中较为常用。在 CAD 程序中, 使用一个大的三维图形窗口显示几何模型的同时, 通常还会使用一些其他的视图以不同的方式来显示文档

数据,例如使用一个树型结构来显示模型的组成结构、属性信息、创建过程等。这样的多视图图显示的例子可以参见图 1-2,该图显示的是本书中使用的 CAD 示例软件——STLViewer 的主界面。在图形用户界面(Graphic User Interface, GUI)中,左侧浮动窗口是一个树型结构的视图(CTreeView)类的对象,用于显示几何模型的结构与属性;右侧的视图窗口用于模型的 OpenGL 三维图形绘制和交互操作。这两个视图对象都与同一文档对象(存储几何模型的信息)相关联。在本书第 9 章还将进一步介绍单文档多视图关系的实现及有关内容。

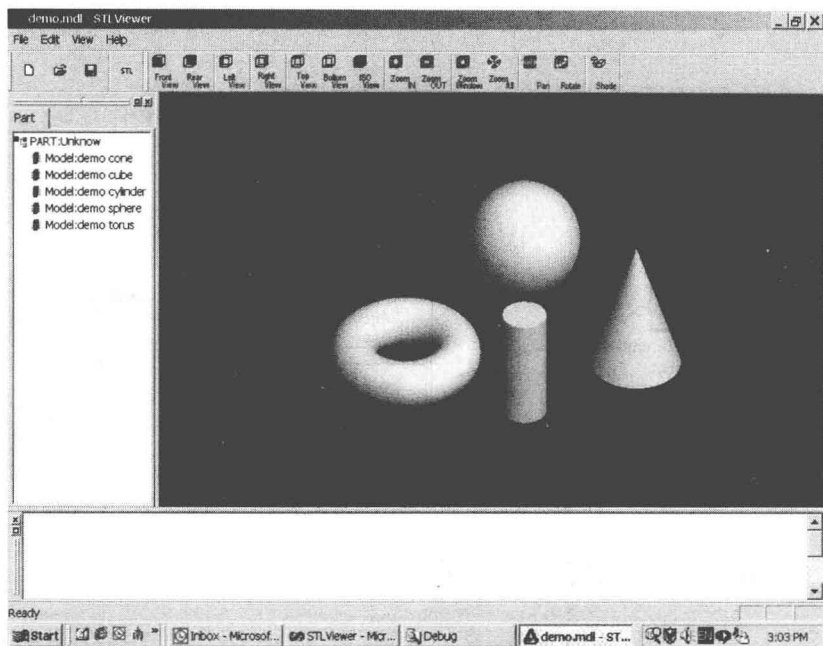


图 1-2 STLViewer 的主界面(单文档多视图结构)

1.2.3 文档模板及主要组成类

MFC 中,文档/视图的结构关系是由文档模板(Document Template)定义的。文档模板用于存放与应用程序文档、视图和框架窗口有关的信息。MFC 类库提供两种文档模板类,即用于单文档(Single Document Interface, SDI)应用程序的 CSingleDocTemplate 和用于多文档(Multiple Document Interface, MDI)应用程序的 CMultiDocTemplate。CSingleDocTemplate 每次只能创建并管理一个文档,而 CMultiDocTemplate 可以创建并管理多个文档。图 1-3 和图 1-4 分别显示了基于 MFC 单文档和多文档的模板结构。

文档模板的创建和维护是在 MFC 的应用程序类 CWinApp 的对象中实现的。在类 CWinApp 调用成员函数 InitInstance() 进行初始化时,必须为应用程序创建一个文档模板对象,并使用函数 AddDocTemplate() 向新创建的应用程序加载这个模板。在应用程序对象中,也可以创建并加载多个文档模板。下面列出的代码是在应用程序类 CSTLViewerApp (CWinApp 的派生类)的成员函数 InitInstance()中创建并加载一个单文档模板的例子。

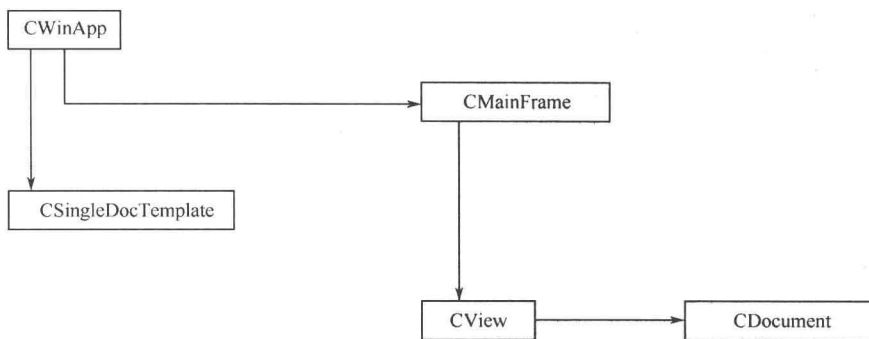


图 1-3 MFC 的单文档的模板序结构

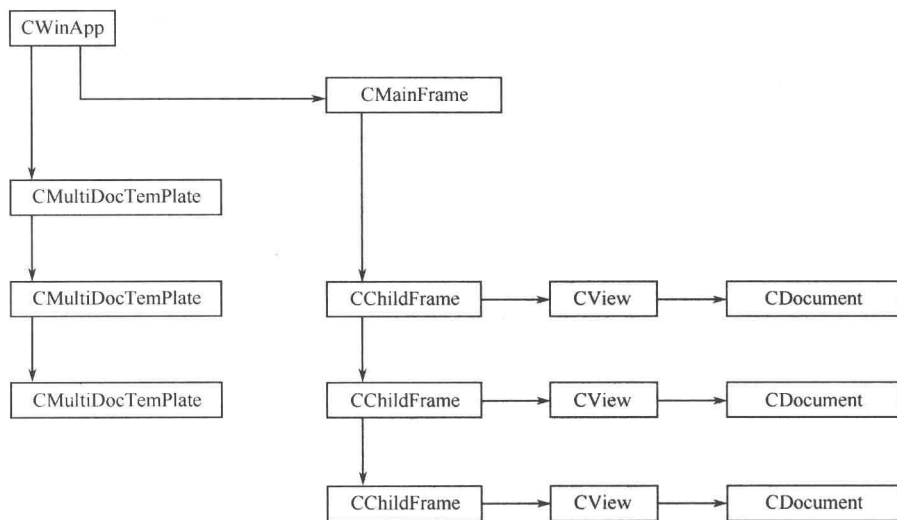


图 1-4 MFC 的多文档的模板结构

```

BOOL CSTLViewerApp::InitInstance()
{
    .....
    CSingleDocTemplate* pDocTemplate;
    pDocTemplate = new CSingleDocTemplate(
        IDR_MAINFRAME,
        RUNTIME_CLASS(CSTLViewerDoc),
        RUNTIME_CLASS(CMainFrame),           //main SDI frame window
        RUNTIME_CLASS(CSTLViewerView));
    AddDocTemplate(pDocTemplate);
    .....
}
  
```

由文档模板定义的 MFC 文档和视图程序框架包括四个主要的类：

- 文档类 CSTLViewerDoc。

- 视图类 `CSTLViewerView`。
- 主框架类 `CMainFrame`。
- 应用程序类 `CSTLViewerApp`。

应用程序的主要功能的实现代码分配在这四个主要的类中。`MFC AppWizard` 为每个类自动生成了源文件。以此为基础,开发人员进一步在这些类中插入需要的代码和对象,就可以实现软件的具体功能。下面将对这四个类的主要功能分别加以介绍。

1. 应用程序类

应用程序类由 `MFC` 提供的用程序基类 `CWinApp` 所派生,负责从总体上实现对程序的管理,管理从程序开始到结束的全过程,例如程序的初始化、运行,以及进行最后的程序清除工作。`MFC` 的文档/视图结构、程序的主窗口都在应用程序类中定义和创建。基于 `MFC` 框架生成的 `Windows` 应用程序必须有且仅有一个从 `CWinApp` 派生的类的对象,在创建窗口之前先构造该对象。

2. 文档类

文档类是由 `MFC` 提供的文档基类 `CDocument` 所派生,负责存放应用程序的数据,并管理程序和磁盘文档文件之间的存储与读取。在建立一个 `CAD` 系统时,几何模型的数据通常是文档中最主要的数据,应该存放在文档类中。文档类把数据处理从界面中分离出来,同时提供一个与其他对象交互的接口。

3. 视图类

视图类是由 `MFC` 基类 `CView` 所派生的,负责显示文档类中的数据,显示的设备可以是计算机屏幕,也可以是打印机或其他设备。视图类还负责处理用户的交互输入,它可以处理多种类型的输入命令,如键盘输入、鼠标输入、菜单及工具栏命令等。在设计一个 `CAD` 系统时,屏幕上的图形绘制、打印机的绘图等功能都需要在视图类中开发完成。

不仅如此,视图类还可通过 `GetDocument()` 函数来获取文档指针,用之读取和操作文档中的数据,从而实现对文档数据的修改。如前提及,当文档中数据发生变化时,可通过调用 `CDocument::UpdateAllViews()` 函数来更新视图的显示内容。

特别值得说明的是, `CView` 是个虚拟类,虚拟类中由于包含了纯虚函数(例如 `CView::OnDraw()` 就是一个纯虚函数),它本身不能直接用于声明对象,但在 `CView` 中封装了许多对视图进行操作的成员函数,可以被其派生类使用。

4. 主框架类

主框架窗口也就是应用程序的主窗口,是由主框架类管理。如图 1-2、图 1-3 所示,单文档应用程序和多文档应用程序的主框架类的 `MFC` 基类是不同的。对单文档应用程序(`SDI`),主框架窗口的基类为 `CFrameWnd` 类;而在多文档应用程序(`MDI`)的情况下,其框架窗口所继承的类为 `CMDIFrameWnd` 类,文档框架窗口所继承的类则为 `CMDIChildWnd` 类,每个文档都有一个文档框架窗口。在用户界面上,一个文档框架窗口至少含有一个视图以显示该文档数据。