

信息科学与技术丛书

程序设计系列

● 王清辉 王 彪 编著

Visual C++ CAD 应用程序开发技术



 机械工业出版社
CHINA MACHINE PRESS



信息科学与技术丛书

Visual C++ CAD 应用程序开发技术

王清辉 王 彪 编著



机械工业出版社

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。内容包括:总体程序框架的分析与设计、功能模块的划分、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)的开发实例贯穿于全书各章节,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。

本书适合于从事图形及 CAD 软件开发的技术人员和具有一定 Visual C++基础的大专院校相关专业学生阅读。

本书提供的开发实例可在 Visual C++ 6.0 版本上实现。

图书在版编目(CIP)数据

Visual C++ CAD 应用程序开发技术/王清辉,王彪编著. —北京:机械工业出版社,2003.7

(信息科学与技术丛书)

ISBN 7-111-12383-2

I. V... II. ①王...②王... III. ①C 语言—程序设计②计算机辅助设计
IV. ①TP312②TP391.72

中国版本图书馆 CIP 数据核字(2003)第 047133 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

策 划:胡毓坚

责任编辑:刘 青

责任印制:闫 焱

北京京丰印刷厂印刷·新华书店北京发行所发行

2003 年 7 月第 1 版·第 1 次印刷

787mm×1092mm 1/16·17.5 印张·432 千字

0 001—5 000 册

定价:32.00 元(含 1CD)

凡购本图书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010) 68993821、88379646

封面无防伪标均为盗版

出版说明

随着信息科学与技术的迅速发展，人类每时每刻都会面对层出不穷的新技术、新概念。毫无疑问，在节奏越来越快的工作和生活中，人们需要通过阅读和学习大量信息丰富、具备实践指导意义的图书，来获取新知识和新技能，从而不断提高自身素质，紧跟信息化时代发展的步伐。

众所周知，在计算机硬件方面，高性价比的解决方案和新型技术的应用一直备受青睐；在软件技术方面，随着计算机软件的规模和复杂性与日俱增，软件技术受到不断挑战，人们一直在为寻求更先进的软件技术而奋斗不止。目前，计算机在社会生活中日益普及，随着因特网延伸到人类世界的层层面面，掌握计算机网络技术和理论已成为大众的文化需求。正是这种在社会各领域的全方位渗透，信息科学与技术正在电工、电子、通信、工业控制、智能建筑、工业产品设计与制造等专业领域中得到充分、广泛的应用。相应地，这些专业领域中的研究人员和工程技术人员将越来越迫切需要汲取自身领域信息化所带来的新理念和新方法。

针对人们对了解和掌握新知识、新技能的热切期待，以及由此促成的人们对语言简洁、内容充实、融合实践经验的图书迫切需要的现状，机械工业出版社适时推出了“信息科学与技术丛书”。这套丛书涉及计算机软件、硬件、网络、工程应用等内容，注重理论与实践相结合，内容实用，层次分明，语言流畅，是信息科学与技术领域专业人员学习和参考不可或缺的图书。

现今，信息科学与技术的发展可谓一日千里，机械工业出版社欢迎从事信息技术方面工作的科研人员、工程技术人员积极参与我们的工作，为推进我国的信息化建设作出贡献。

机械工业出版社

前 言

CAD 软件的开发是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较多,如何规划和展开软件的开发是系统开发成功与否的关键问题之一。本书的特点在于,从完整的、模块化结构的软件设计与实现的角度逐步讲述一个三维 CAD 软件从系统设计到使用 Visual C++编程实现的各个主要环节。着重于讲述使用 Visual C++、OpenGL 进行图形及 CAD 软件开发的有关技术问题,将所涉及到的许多技术难点融入到具体的开发实例中,使读者易于理解和掌握。在讲述中,以一个三维 CAD 原型系统 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的广泛应用,Visual C++优秀的开发环境、Microsoft 推出的 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 and 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。

本书内容共分九章,分别介绍了基于 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;CAD 软件的图形交互;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++ 6.0 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。作者的电子邮件地址: wangqh15@yahoo.com.sg。

作 者

目 录

出版说明

前言

第 1 章 基于 MFC 的三维 CAD 应用

程序框架结构分析	1
1.1 MFC 应用程序的文档/视图结构	2
1.1.1 文档/视图结构概述	2
1.1.2 文档与多个视图的关联	2
1.1.3 文档模板及主要组成类	3
1.2 实例分析——三维 CAD 示例软件 STLViewer	6
1.3 STLViewer 的程序框架	6
1.4 STLViewer 中类的层次设计及软件模块结构划分	11
1.5 建立和使用动态链接库	12
1.5.1 动态链接库的基本概念	13
1.5.2 基于 MFC 的动态链接库	13
本章相关程序	14

第 2 章 几何基本工具库的开发

2.1 点、矢量和齐次变换矩阵	15
2.1.1 点	15
2.1.2 矢量	16
2.1.3 齐次坐标与齐次变换矩阵	17
2.2 设计点、矢量和齐次变换矩阵类	18
2.2.1 点类 CPoint3D	19
2.2.2 矢量类 CVector3D	22
2.2.3 变换矩阵类 CMatrix3D	25
2.3 三维图形的几何变换	28
2.3.1 三维齐次变换矩阵	28
2.3.2 平移变换	29
2.3.3 旋转变换/绕空间任意轴的旋转变换函数的实现	29
2.3.4 几何缩放	32
2.3.5 对称变换/沿空间任意平面的对称变换函数的实现	33

2.4 设计几何基本工具库

GeomCalc.dll	34
2.4.1 GeomCalc.dll 中的输出类与输出函数	34
2.4.2 创建几何基本工具库 GeomCalc.dll 的步骤	35
2.4.3 使用 GeomCalc.dll	37
2.5 有关源程序代码	37
2.5.1 文件 CadBase.h	37
2.5.2 文件 CadBase.cpp	40
2.5.3 文件 CadBase1.cpp	49
本章相关程序	50

第 3 章 基于 MFC 的 OpenGL

Windows 程序的创建	51
3.1 OpenGL 介绍	51
3.2 在 Windows 环境下使用 OpenGL	52
3.2.1 OpenGL 的函数库	52
3.2.2 OpenGL 与 GDI	53
3.2.3 渲染场境	53
3.2.4 像素格式	56
3.3 OpenGL MFC 应用程序创建实例	60
3.3.1 创建一个应用程序框架	61
3.3.2 修改视图类 CGLView	63
3.3.3 使用 OpenGL 的双缓存技术为应用程序增加动画效果	69
3.4 程序清单	71
3.4.1 文件 GLView.h	71
3.4.2 文件 GLView.cpp	72
本章相关程序	78

第 4 章 封装 OpenGL 功能的 C++ 类的设计

4.1 封装 OpenGL 的 C++ 类的设计	79
4.2 照相机类 GCamera 的设计	80

4.2.1 视点坐标系和视图变换	80	(类 GCamera)	141
4.2.2 投影变换与视景体	82	5.6.2 文件 Camera.cpp	
4.2.3 视口变换	83	(类 GCamera)	142
4.2.4 设计照相机类 GCamera	85	5.6.3 文件 OpenGLDC.h	
4.3 类 COpenGLDC	88	(类 COpenGLDC、	
4.4 修改类 CGLView	93	CGLView)	147
4.5 运行应用程序	96	5.6.4 文件 OpenGLDC.cpp	
4.6 源程序清单	96	(类 OpenGLDC)	151
4.6.1 类 GCamera 的声明代码	96	5.6.5 文件 GLView.cpp	
4.6.2 类 GCamera 的实现代码	97	(类 CGLView)	160
4.6.3 类 COpenGLDC 的声明代码	99	本章相关程序	163
4.6.4 类 COpenGLDC 的实现		第 6 章 CAD 应用程序的几何内核	
代码	100	模块的设计	164
4.6.5 类 CGLView 的声明代码	103	6.1 几何对象类的设计	164
4.6.6 类 CGLView 的实现代码	104	6.1.1 类的层次设计	164
本章相关程序	106	6.1.2 几何对象基本类 CEntity	167
第 5 章 基于 OpenGL 的 CAD 图形		6.1.3 三角面片对象类 CTriChip	170
工具库的设计	107	6.1.4 STL 几何模型类	
5.1 创建动态链接库		CSTLModel	171
glContext.dll	107	6.1.5 高级几何模型类 CPart	176
5.2 类 GCamera 的功能增强	108	6.2 串行化(Serialize)实现文档	
5.2.1 选择典型的观察视图	110	存取功能	179
5.2.2 景物平移	113	6.2.1 为什么要使用串行化	179
5.2.3 景物缩放	114	6.2.2 CArchive 类	179
5.2.4 使用 OpenGL 的选择		6.2.3 串行化类的设计步骤	180
模式	115	6.2.4 CObArray 的 Serialize()函数	181
5.3 类 COpenGLDC 功能的		6.2.5 应用程序的文档串行化	
增强	116	实例剖析	181
5.3.1 实现和 Windows 窗口的		6.3 虚拟函数	184
关联	120	6.3.1 虚拟函数与多态性	184
5.3.2 定义光源	120	6.3.2 纯虚拟函数	186
5.3.3 定义颜色	126	6.3.3 实现 CPart 模型的 OpenGL	
5.3.4 图形绘制函数	131	显示	186
5.3.5 选择模式	136	6.4 建立几何内核库	
5.4 增加类 CGLView 中的		GeomKernel.dll	188
功能	137	6.5 程序清单	189
5.5 glContext 类的输出和调用	140	6.5.1 文件 Entity.h	189
5.6 源程序清单	141	6.5.2 文件 Entity.cpp	192
5.6.1 文件 Camera.h		本章相关程序	200

第7章 CAD 应用程序 STLViewer

的模块化实现	201
7.1 STLViewer 的模块结构	201
7.2 创建应用程序框架	202
7.3 修改应用程序框架	205
7.3.1 增加界面资源	205
7.3.2 修改框架类 CMainFrame ...	206
7.3.3 修改文档类 CSTLViewerDoc	207
7.3.4 修改视图类 CSTLViewerView	209
7.4 运行 STLViewer.exe	215
7.5 源程序清单	216
7.5.1 文件 MainFrm.h	216
7.5.2 文件 MainFrm.cpp	217
7.5.3 文件 STLViewerDoc.h	219
7.5.4 文件 STLViewerDoc.cpp	220
7.5.5 文件 STLViewerView.h	222
7.5.6 文件 STLViewerView.cpp ...	224
本章相关程序	228

第8章 增强 CAD 应用程序的

界面功能	229
8.1 STLViewer 的界面增强	229
8.2 工具栏的排列	230
8.3 使用快捷菜单	232
8.4 创建类似 Visual Studio 风格 的浮动窗口	232
8.4.1 控制条与停靠栏	233
8.4.2 开发具有 Visual Studio 风格的 浮动窗口	235
8.4.3 CTabCtrl 控件的功能增强 ...	237
8.4.4 建立界面工具库 DockTool.dll	243

8.5 使用树型视图 CTreeView

显示和管理文档数据	243
8.5.1 树型视图与树型控件	244
8.5.2 在 STLViewer 中创建 CPartTreeView	244
8.5.3 树型视图 CPartTreeView 与 文档的关联/文档多视图 ...	246
8.5.4 在树型控件中使用图标 ...	248
8.5.5 使用树型视图控件显示文档中 几何模型的结构和属性	249
8.5.6 通过树型视图控件对文档 数据进行操作	251
本章相关程序	253

第9章 基于 OpenGL 的 CAD 软件

拾取功能的实现	255
9.1 使用 OpenGL 选择模式	256
9.1.1 OpenGL 的三种操作模式 ...	256
9.1.2 使用选择模式	256
9.2 一个 OpenGL 选择模式的应 用程序	262
9.3 OpenGL 的选择功能与 CAD 应用程序的集成	266
9.3.1 定义选择视景物/修改类 CCamera	266
9.3.2 对选择过程的操作/修改类 COpenGLDC	267
9.3.3 自动给对象命名/对类 CSTLModel 的修改	269
9.3.4 在 STLViewer 中调用拾取 功能	270
9.3.5 运行程序	271
本章相关程序	271

第 1 章 基于 MFC 的三维 CAD 应用程序 框架结构分析

本章要点:

- 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。
- 动态链接库的创建与使用。

在 Visual C++ 2.0 以后的版本中, Microsoft 公司推出了 MFC (Microsoft Foundation Class) 类库。MFC 类库是用来编写 Windows 程序的 C++ 类集。使用 MFC 类库, 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。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”。应用程序框架设计得好坏与否, 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 (CAD/CAM/CAE: Computer Aided Design/ Manufacturing/ Engineering) 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 (RPM: Rapid Prototyping Manufacturing) 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, 并贯穿全书, 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, 本章也在示例中讲解了基于 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, 通过循序渐进的介绍, 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。

1.1 MFC 应用程序的文档/视图结构

1.1.1 文档/视图结构概述

MFC 提供了一个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导引步骤（MFC AppWizard）就可以创建一个基于文档/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数据；而视图用于显示文档中的数据，并处理与用户的交互信息。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。

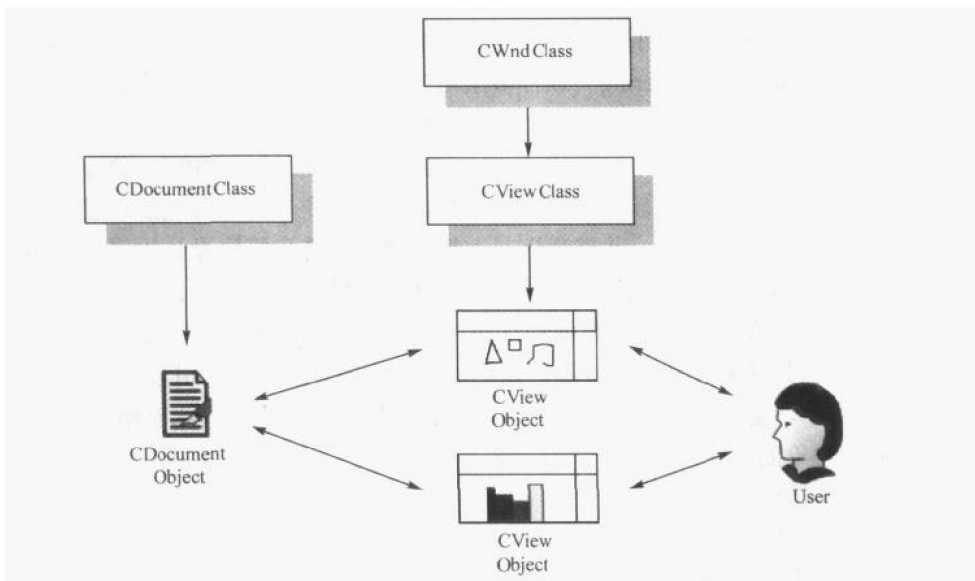


图 1-1 文档与视图的关系

1.1.2 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或一对多，即一个文档可以关联一个或一个以上的视图，也就是可以用多个视图来显示和操作文档中的数据。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。如图 1-1 所示，一个文档对象就同时关联了多个视图对象。当在一个视图类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对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，图形显示给用户直观的感觉，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 应用程序中也较常用。在 CAD 程序中，使用一个大的视图窗口用于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，如使用一个树型结构来显示模型的组成结构和属性信息等。这样的多视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（GUI: Graphic User Interface）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 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 8 章还将进一步介绍单文档多视图关系的实现及有关内容。

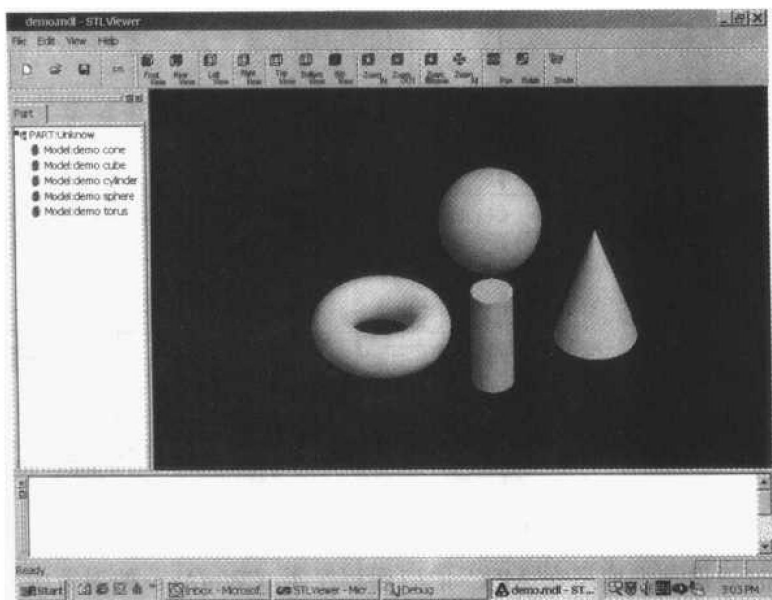


图 1-2 STLViewer 的主界面

1.1.3 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 类库提供有两种文档模板类，即用于单文档（SDI）应用程序的 `CSingleDocTemplate` 和用于多文档（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、图 1-4 分别显示了基于 MFC 单文档和多文档的模板结构。

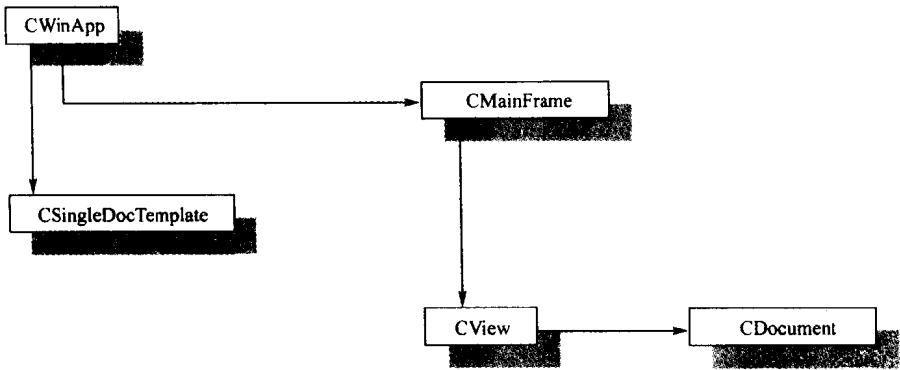


图 1-3 MFC 的单文档应用程序结构

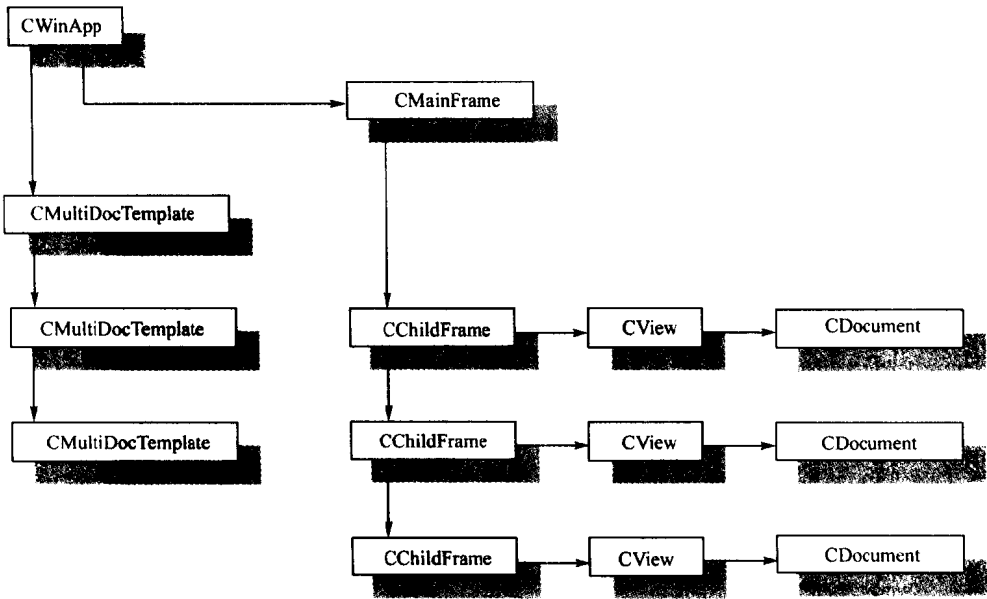


图 1-4 MFC 的多文档应用程序结构

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。

```
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实现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，就可以实现软件的专业功能。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 提供的程序基类 `CWinApp` 所派生，负责从总体上实现对程序的管理。例如初始化程序以及进行最后的程序清除工作。应用程序类管理从程序开始到结束的全过程。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。每个 MFC 应用程序都必须生成一个 `CWinApp` 派生类的对象。

(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，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、菜单以及工具栏命令等。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，用于读取和操作文档中的数据，从而实现对文档数据的修改。如前面所提及的，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。

需要特别说明的是，`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 1-3、图 1-4 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。

主框架窗口还负责管理用户界面对象，如菜单、工具条、状态栏等。每个框架窗口对应管理一个可选的加速键表，改变加速键表的内容就可以自动转换键盘的加速键，这样可以方便地定义调用菜单命令的加速键。

主框架窗口同时管理所有的视图窗口，并跟踪当前活动的视图。当框架窗口含有多个视图窗口时，当前视图就是最近使用的视图。当视图活动改变时，边框窗口通过调用 `CView` 的成员函数 `OnActiveView()` 来通知当前视图。例如，在如图 1-2 所示的应用程序界面中，主框架窗口就同时管理了三个视图，分别是左边的树型视图、右边的 OpenGL 图形显示视图和下边的信息输出视图。

1.2 实例分析——三维 CAD 示例软件 STLViewer

本书将结合一个三维 CAD 软件 STLViewer 的设计与开发，讲述在 MFC 环境下使用面向对象的方法设计三维 CAD 软件的一些技术。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、如何在 MFC 环境下使用 OpenGL 进行图形绘制、为 CAD 应用程序开发 OpenGL 的通用绘图类、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、软件的界面设计等。

这里，首先介绍一下 STLViewer 的主要功能。图 1-2 是 STLViewer 的主界面。作为一个简化了的三维 CAD 软件，STLViewer 可以接受输入的 STL 几何模型，并将 STL 文件转化为系统自定义的几何模型，修改并增加属性，还可以采用串行化（Serialize）方法存储和装载系统自定义的几何模型（*.mdl 文件）。在 STLViewer 中，使用 OpenGL 对几何模型进行了三维真实感渲染，并对模型进行视角变换、显示缩放、光照设置以及鼠标拾取等。在界面中，设计了具有 Visual Studio 界面风格的浮动窗口和信息输出框。浮动窗口采用树型控件，用于显示几何模型结构、修改模型的附加信息（颜色、名称等）和操作选取几何模型。信息输出框用于输出系统提示信息，类似于 Visual Studio 开发环境中的 Output 窗口。

STL 文件，即立体光造型文件（STL File, Stereo Lithographic File），是描述三维几何形状的标准文件格式之一，它采用一系列离散的三角片来描述三维曲面形状。目前，大多数 CAD 软件都具有了 STL 文件输出接口，可以将三维几何模型输出成 STL 文件。本书所附带的 STL 文件（ASCII 格式）均是从 AutoCAD R14 中输出的。STL 文件在工业领域有较多的应用，如快速原型制造（RPM, Rapid Prototyping Manufacturing）就使用 STL 文件作为加工模型的几何输入接口。贯穿本书的示例程序 STLViewer 就是作者对参与开发的一个快速原型制造软件作了大量简化后的结果。

1.3 STLViewer 的程序框架

图 1-5 显示了 STLViewer 的 MFC 文档/视图结构框架。STLViewer 采用 MFC 提供的单文档模板，其中包含四个主要的应用程序类：

- 文档类 `CSTLViewerDoc`。
- 视图类 `CSTLViewerView`。
- 主框架类 `CMainFrame`。

● 应用程序类 CSTLViewerApp。

它们所对应的源文件分别是：STLViewerDoc.h，STLViewerDoc.cpp，STLViewerView.h，STLViewerView.cpp，MainFrm.h，MainFrm.cpp，STLViewerApp.h，STLViewerApp.cpp。所有本书提及的源程序 Visual C++ 代码都在随书附带的光盘内。

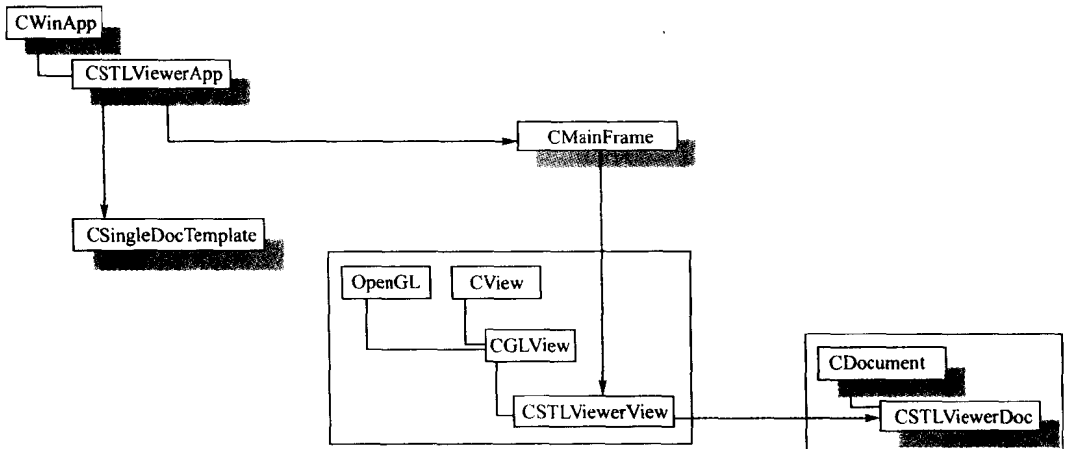


图 1-5 STLViewer 的文档/视图结构

下面将对这四个应用程序类分别加以分析。

(1) 文档类 CSTLViewerDoc。文档类负责管理应用程序的数据。在 STLViewer 中，CSTLViewerDoc 由 MFC 的文档基类 CDocument 派生，并插入了存放几何模型的对象 m_Part。m_Part 由一个自定义的几何类 CPart 定义，用于存储和管理全部的几何模型。

CSTLViewerDoc 的定义如下（详见本书附带光盘内的 STLViewerDoc.h 文件）：

```

class CSTLViewerDoc : public CDocument
{
protected: // create from serialization only
    CSTLViewerDoc();
    DECLARE_DYNCREATE(CSTLViewerDoc)

// Attributes
public:
    CPart m_Part;           //几何模型对象

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{AFX_VIRTUAL(CSTLViewerDoc)
    public:
        virtual BOOL OnNewDocument();

```

```

        virtual void Serialize(CArchive& ar);
        //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CSTLViewerDoc();
#ifdef _DEBUG
    virtual void AssertValid()const;
    virtual void Dump(CDumpContext& dc)const;
#endif

protected:
// Generated message map functions
protected:
    //{AFX_MSG(CSTLViewerDoc)
    afx_msg void OnStdFilein();
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()
};

```

对 `m_Part` 进行串行化存储和读取操作由 `CSTLViewerDoc` 的串行化函数 `Serialize()` 实现。用 `m_Part` 代表的几何模型可以以二进制文件的形式存储在磁盘上，或从一个磁盘文件中读取出来并创建该对象。

`CPart` 由一个自行开发的动态链接库 `GeomKernel.dll` 输出。关于 `CPart` 类的描述，将在本书第 6 章开发几何内核库时详细介绍。

(2) 视图类 `CSTLViewerView`。视图类负责文档类中数据的显示，以及负责处理用户与图形窗口之间的交互操作。`CSTLViewerView` 在 `CGLView` 基础上派生，用于在 `OpenGL` 的三维环境下绘制和操作几何模型对象 `m_Part`。通常，应用程序的视图类都是由 `MFC` 提供的基类 `CView` 派生得来。但由于 `CView` 作为一个视图窗口的基础类，没有直接调用 `OpenGL` 绘制三维几何图形的功能。在本书开发的 `CAD` 图形显示动态链接库 `glContext.dll` 中，专门开发了支持 `OpenGL` 的视图类 `CGLView`，它由 `CView` 类派生，并采用面向对象技术封装了 `OpenGL` 的图形绘制功能。`CGLView` 的设计将在以后的章节中详细论述。由于在类 `CGLView` 中已经实现了主要的模型显示和操作功能，类 `CSTLViewerView` 所承担的任务更多的是接受用户输入信息，并调用 `CGLView` 中的相关函数处理用户输入。

`CSTLViewerView` 的定义如下（详见本书附带光盘内的 `STLViewerView.h` 文件）：

```

class CSTLViewerView : public CGLView
{
protected: // create from serialization only
    CSTLViewerView();
    DECLARE_DYNCREATE(CSTLViewerView)

// Attributes

```

```

public:
    CSTLViewerDoc* GetDocument();

    virtual void RenderScene(COpenGLDC* pDC); // 用 OpenGL 显示几何体

// Operations
public:

// Overrides
    // ClassWizard generated virtual function overrides
    //{ {AFX_VIRTUAL(CSTLViewerView)
    public:
        virtual BOOL PreCreateWindow(CREATESTRUCT& cs);
    protected:
    //}}AFX_VIRTUAL

// Implementation
public:
    virtual ~CSTLViewerView();
#ifdef _DEBUG
    virtual void AssertValid()const;
    virtual void Dump(CDumpContext& dc)const;
#endif

protected:

// Generated message map functions
protected:
    //{ {AFX_MSG(CSTLViewerView)
    //}}AFX_MSG
    DECLARE_MESSAGE_MAP()

    virtual BOOL GetBox(double& x0,double& y0,double& z0,
        double& x1,double& y1,double& z1); // 获取视图中显示模型的最大包围盒

};

```

视图类中的虚拟函数 `CSTLViewerView::RenderScene()` 在基类 `CGLView::OnDraw()` 中被调用，用于 OpenGL 的场景绘制。

(3) 主框架类 `CMainFrame`。主框架类提供了文档界面的主窗口，并创建和管理系统菜单、浮动工具条和状态条等界面对象。程序 `STLViewer` 中，在 MFC AppWizard 生成的界面基础上进一步增强了用户界面的功能，即在主框架类 `CMainFrame` 中插入了一个具有 Visual Studio 界面风格的浮动窗口（图 1-2 所示界面中左边的包含树型视图的窗口）对象和一个信息输出框对象（图 1-2 所示界面中底部的信息输出窗口），分别用于几何模型的浏览操作和提示信息的输出。程序示例中给出了具体的程序实现。浮动窗口和信息输出框对象是分别由