复杂系统的面向对象建模、分析与设计

范玉顺　曹军威　　编著

清华大学出版社　施普林格出版社

2000 年 10 月

序

面向对象方法在经历了 30 年的发展之后，已经成为软件系统分析、设计与实现的最重要的方法。无论从提高软件系统的可重用性方面，还是从降低软件系统的复杂性方面，面向对象技术作为软件开发的新方法都表现出了明显的优势。其增量式的循环开发过程更符合复杂软件系统实际需求和开发过程。因而，它也就理所当然受到广泛的重视和欢迎。将面向对象技术应用于复杂系统的设计与开发可以大大提高软件的规范化程度和开发效率。

目前，国内外已经有许多面向对象技术方面的专著，但大多数是介绍面向对象的编程技术，以介绍面向对象的编程语言为主，如 C＋＋， Visual C++等。这些著作对于普及面向对象的概念有重要意义，在很大程度上可以帮助软件开发人员提高软件开发水平。但是，面向对象编程（Object-Oriented Programming，简称 OOP）仅是面向对象技术的组成部分之一。对于复杂软件系统的设计和开发，仅仅会使用面向对象编程语言是远远不够的。从某种程度上讲，如果软件开发人员仅仅掌握了面向对象的编程语言而没有对面向对象技术有全面的了解，他们开发的软件系统在本质上可能并不是真正的面向对象的软件系统，而仅仅是使用了对象、类、封装等概念，是一个使用了面向对象编程语言开发的一个似是而非的面向对象软件系统。

广大的软件开发人员，特别是组织复杂系统开发的项目负责人，迫切需要了解面向对象技术的全部内涵，除了要掌握面向对象的编程技术外，还需要了解面向对象的分析（Object-Oriented Analysis，简称 OOA）技术和面向对象的设计（Object-Oriented Design，简称 OOD）技术。在此基础上还要学会使用面向对象的设计工具及面向对象的文档生成与管理工具。有了理论基础并掌握了先进的工具，同时充分借鉴他人的开发经验，成功地开发复杂软件系统就有了可靠的保证。

为了向读者全面介绍面向对象技术以及使用该技术开发复杂软件系统的方法，作者结合自己在使用面向对象方法和工具设计开发 863/CIMS 主题重大关键技术攻关项目“制造业 CIMS 应用集成平台原型系统开发”中的实际经验，通过对面向对象方法、面向对象设计开发过程、面向对象设计工具及文档规范生成与管理工具的介绍，全面地为读者提供一整套实用的面向对象技术。通过应用实例介绍，使读者深入了解面向对象技术的优势和应用过程，为读者学习和使用面向对象方法提供了生动而具体的参考。

本书全文由范玉顺教授负责编著，曹军威提供了本书第 5，7，8 章的初稿，由范玉顺完成修改校定。在本书的写作过程中，得到了清华大学计算中心俞盘祥教授的大力支持和指导，俞教授不仅鼓励和支持作者完成本书的写作，还悉心审阅了全书，并提出了许多建设性的意见，在此表示衷心的感谢。作者还特别要感谢清华大学自动化系吴澄院士，吴澄院士不仅是 863/CIMS 主题重大关键技术攻关项目“制造业 CIMS 应用集成平台原型系统开发”的倡导者和规划者，而且在作者负责该项目期间，从总体设计到关键技术攻关都给予了全面的指导和支持，该项目的成功是本书得以完成的基础。衷心感谢中国太平洋软件公司为本书提供了面向对象设计工具 Rose4.0 并授权将此软件的测试版作为本书的附件提供给读者。书中许多内容均是作者研究开发工作的体会。由于作者水平有限，书中的缺点和错误在所难免，欢迎广大读者批评指正。

范玉顺

1999 年 5 月于清华园

目　录

序

第 1 章　引言
1.1 软件工程的基本概念
1.2 软件需求分析和开发的结构化方法
1.2.1 结构化分析与设计方法
1.2.2 Jackson 系统开发方法
1.3 结构化方法的特点与不足
1.3.1 结构化方法的特点
1.3.2 结构化方法存在的不足
1.4 面向对象技术
1.4.1 面向对象的分析、设计与编程方法
1.4.2 面向对象软件工程
1.4.3 OMT 方法
1.4.4 一体化建模语言
1.5 Booch 方法简介

第 2 章　对象模型
2.1 对象模型的基本元素
2.1.1 抽象
2.1.2 封装
2.1.3 模块化
2.1.4 层次性
2.1.5 类型
2.1.6 并发性
2.1.7 持久性
2.2 面向对象方法与对象模型的优点

第 3 章　对象和类
3.1 对象
3.1.1 对象的状态
3.1.2 对象的行为
3.1.3 对象的标识
3.1.4 对象的生命周期
3.2 对象间的关系
3.2.1 连接
3.2.2 聚合
3.3 类
3.4 对象与类之间的关系及不同作用

第 4 章 对象与类的抽象机制及表示方法
4.1 类与对象抽象机制与方法
4.2 如何建造高质量的类和对象
4.2.1 检验抽象的质量
4.2.2 选择合适的操作
4.2.3 选择合适的关系
4.2.4 选择合适的实施方法
4.3 对象与类的基本表示形式
4.3.1 类图及类范畴图
4.3.2 其它类图表示形式
4.3.3 形式化定义
4.4 状态转换图
4.4.1 状态
4.4.2 态转换的深入的概念
4.5 对象图
4.6 交互图
4.7 模块图
4.8 处理图

第 5 章 面向对象的设计工具
5.1 概述
5.1.1 面向对象的设计工具的特点
5.1.2 面向对象方法的系列设计开发工具
5.2 面向对象的设计工具 Rational Rose/C++简介
5.3 Rose/C++使用方法
5.4 使用 ROSE/C++建立系统模型
5.4.1 Rose 的安装与启动
5.4.2 类图的建立
5.4.3 面向对象设计中其它模型的建立
5.5 ROSE/C++对面向对象设计实施提供的其它支持功能
5.5.1 Use-Case 建模工具
5.5.2 UML 建模功能
5.5.3 Rose 对代码生成、DDL 和 IDL 接口的支持
5.5.4 方便的用户联机帮助和自学指导

第 6 章 面向对象设计开发过程
6.1 面向对象设计开发过程概述
6.2 微过程方法
6.2.1 确定类和对象
6.2.2 确定类和对象语义
6.2.3 确定类和对象之间的关系
6.2.4 实施类和对象
6.3 面向对象设计开发主流程
6.3.1 概念建立

6.3.2 系统分析
6.3.3 系统设计
6.3.4 系统实施
6.3.5 运行维护

第 7 章 面向对象软件开发规范与文档生成工具
7.1 面向对象软件开发规范
7.2 面向对象软件设计开发文档细则
7.2.1 文档组成
7.2.2 文档细则
7.3 面向对象文档生成工具 SoDA for Word
7.3.1 SoDA 简述
7.3.2 SoDA 入门
7.3.3 SoDA 模板定制
7.3.4 模板示例

第 8 章 应用实例研究
8.1 应用实例背景介绍
8.1.1 集成平台的产生背景
8.1.2 CIMS 应用集成平台的基本概念
8.1.3 制造业 CIMS 应用集成平台的体系结构与功能
8.1.4 运控代理模型
8.1.5 控制代理
8.2 应用实例的面向对象设计分析
8.2.1 概念建立阶段文档
8.2.2 系统分析阶段文档
8.2.3 系统设计阶段文档
8.2.4 系统实施阶段文档
8.2.5 运行维护阶段文档
8.3 结论

第 9 章 软件设计开发技术展望
9.1 CORBA 规范及其应用简介
9.1.1 CORBA 的产生
9.1.2 对象管理参考模型及 CORBA 体系结构
9.1.3 CORBA 的应用状况
9.2 软件代理技术简介
9.2.1 为什么软件需要代理
9.2.2 什么是软件代理
9.2.3 代理通信语言
9.2.4 代理实现的典型应用

参考文献
附录:BOOCH 表示方法快速参考图例