

<<使用Visual Basic和UML开>>

图书基本信息

书名：<<使用Visual Basic和UML开发应用程序>>

13位ISBN编号：9787302054658

10位ISBN编号：7302054657

出版时间：2002-6

出版时间：清华大学出版社

作者：Reed

页数：345

字数：553

译者：李博

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<使用Visual Basic和UML开>>

内容概要

本书结合作者在面向对象的客户 / 服务器系统开发方面的丰富经验，介绍了如何在Visual Basic中用UML对企业级应用程序进行建模和开发。

书中不仅对UML基础知识和Visual Basic中易被误用的面向对象属性进行了清晰阐释，还围绕一个大型案例研究，引导读者亲历项目开发周期的各个阶段：需求分析、建模、设计原型、创建体系结构到编码，让读者了解UML的设计优势，告诉读者如何把UML规范转换成Visual Basic代码，还教会读者如何结合DNA，DCOM和MTC技术及ASP等工具开发面向对象的Visual Basic应用和网络应用。

本书实用性强，适用于任何想成功创建VB应用程序并保证其长期运行的程序开发人员。

<<使用Visual Basic和UML开>>

书籍目录

第1章 项目的困境

- 1.1 目标
- 1.2 项目的困境
 - 1.2.1 迭代和增量的软件开发
 - 1.2.2 基于风险的开发
 - 1.2.3 迭代软件过程模型
 - 1.2.4 将迭代与增量结合起来：多维视图
- 1.3 Synergy过程模型
 - 1.3.1 推销软件过程的思想
- 1.4 统一建模语言
 - 1.4.1 UML和它在软件过程中的地位
 - 1.4.2 建模的本质
 - 1.4.3 UML图
 - 1.4.4 统一建模语言和"4 + 1"结构视图
 - 1.4.5 在上下文中使用UML图
- 1.5 检查点
 - 1.5.1 我们讲了什么
 - 1.5.2 我们将要讲什么

第2章 Visual Basic，面向对象和UML

- 2.1 目标
- 2.2 Visual Basic是一种有力的企业开发工具
- 2.3 Visual Basic和面向对象的概念
 - 2.3.1 Visual Basic和类
 - 2.3.2 Visual Basic和复合类型
 - 2.3.3 Visual Basic和消息传递
 - 2.3.4 Visual Basic和封装
 - 2.3.5 Visual Basic和继承
 - 2.3.6 Visual Basic和接口继承
 - 2.3.7 Visual Basic中一种实现继承的替代方法
 - 2.3.8 Visual Basic和多态
- 2.4 为什么选择UML和Visual Basic
 - 2.4.1 类图
 - 2.4.2 顺序图
 - 2.4.3 组件图
 - 2.4.4 部署图
 - 2.4.5 可视化建模工具支持
- 2.5 检查点
 - 2.5.1 我们讲了什么
 - 2.5.2 我们将要讲什么

第3章 开始项目

- 3.1 目标
- 3.2 建立项目规划
 - 3.2.1 过程模型
 - 3.2.2 项目规划的工作模板
 - 3.2.3 活动者

<<使用Visual Basic和UML开>>

3.2.4 事件列表和事件表

3.2.5 项目规划：迭代一

3.2.6 "迭代一"结束

3.3 检查点

3.3.1 我们讲了什么

3.3.2 我们将要讲什么

第4章 用树

4.1 目标

4.2 项目例子

4.3 过程模型

4.3.1 用例

4.4 寻找用例中的路径

4.4.1 找到愉快路径

4.4.2 找到可选路径

4.4.3 找到例外路径

4.5 阴影中的用例

4.6 细化愉快路径

4.7 完成处理订单用例

4.8 准备初步的体系结构

4.9 项目规划：增量和估算

4.9.1 增量

4.9.2 估算

4.10 检查点

4.10.1 我们讲了什么

4.10.2 我们将要讲什么

第5章 类

5.1 目标

5.2 细化阶段

5.3 细化路径

5.4 识别和分类商业规则

5.5 挖掘类

5.5.1 迭代1：UML类图的作用

5.5.2 什么构成一个好的类

5.5.3 应用过滤规则

5.5.4 类的类型

5.5.5 实体类

5.5.6 接口类

5.5.7 控制类

5.6 关系

5.6.1 创建关联

5.6.2 创建角色

5.6.3 创建多重性

5.6.4 高级关联

5.6.5 聚合和组合关联

5.6.6 链关联（关联类）

5.6.7 自反关联

5.6.8 限定关联

<<使用Visual Basic和UML开>>

- 5.6.9 泛化
- 5.7 创建类图
- 5.8 识别属性和操作
 - 5.8.1 属性
 - 5.8.2 操作
- 5.9 对象图
- 5.10 结束：分析模型
- 5.11 检查点
 - 5.11.1 我们讲了什么
 - 5.11.2 我们将要讲什么
- 第6章 创建一个早期原型
 - 6.1 目标
 - 6.2 创建一个早期原型
 - 6.2.1 原型
 - 6.3 收集需求信息
 - 6.3.1 用户接口原型
 - 6.3.2 活动者利用例边界
 - 6.3.3 用户接口产品
 - 6.3.4 用例耦合
 - 6.4 迭代一
 - 6.4.1 窗口结构图
 - 6.4.2 创建原型
 - 6.4.3 通过使用屏幕对话框获得用户反馈信息
 - 6.4.4 从原型中获取信息
 - 6.5 检查点
 - 6.5.1 我们讲了什么
 - 6.5.2 我们将要讲什么
- 第7章 应用程序中的动态元素
 - 7.1 目标
 - 7.2 细化阶段的下一步
 - 7.3 动态建模
 - 7.3.1 动态模型的类型
 - 7.4 顺序图
 - 7.4.1 顺序图和快乐路径
 - 7.4.2 可选路径的顺序图
 - 7.4.3 将获取知识反映到类图中
 - 7.4.4 浏览顺序图
 - 7.5 协作图
 - 7.6 状态图
 - 7.6.1 为Remulak Productions订单类的状态图建模
 - 7.6.2 另一个角度看状态图
 - 7.7 活动图
 - 7.8 选择正确的图表
 - 7.9 设计过程中的非UML内容：应用矩阵
 - 7.9.1 事件 / 频率矩阵
 - 7.9.2 对象 / 位置矩阵
 - 7.9.3 对象 / 容积矩阵

<<使用Visual Basic和UML开>>

7.10 检查点

7.10.1 我们讲了什么

7.10.2 我们将要讲什么

第8章 技术概述

8.1 目标

8.2 细化阶段的下一个阶段

8.3 分离服务

8.4 逻辑和物理层

8.5 Micorosoft的分层策略

8.5.1 六层之间的通信

8.5.2 进程间通信结构

8.5.3 层间通信结构

8.5.4 COM内通信

8.5.5 基础结构所基于的五个选项

8.6 管理应用程序中的事务作用域和 Microsoft Transaction Server

8.7 将Internet包含到解决方案中

8.8 Remulak Productions执行结核

8.9 检查点

8.9.1 我们讲了什么

8.9.2 我们将要讲什么

第9章 数据持久性：存储对象

9.1 目标

9.2 构造阶段

9.3 面向对象的概念以及转化为物理设计

9.4 将类映射到表

9.5 映射简单关联

9.6 将继承映射到关系数据库

9.7 将聚会和组合映射到关系数据库

9.8 将自反关联映射到关系数据库

9.9 键码结构和正规化

9.10 使用可视化建模工具来生成数据定义语言

9.10.1 改进可视化建模工具

9.11 存储过程和触发器以及面向对象工程

9.12 数据敏感类的 Visual Basic支持

9.13 数据转化服务和数据访问服务层

9.14 检查点

9.14.1 我们讲了什么

9.14.2 我们将要讲什么

第10章 应用基础结构

10.1 目标

10.2 构造阶段

10.2.1 synergy过程

10.2.2 组件--基础结构和所有层的通信

10.2.3 组件--探讨表示服务层

10.2.4 组件--探讨商业上下文服务层

10.2.5 组件--探讨商业规则服务层

10.2.6 组件--合作类：接口、控制和实体

<<使用Visual Basic和UML开>>

- 10.2.7 组件--层通信
- 10.2.8 组件--实现基础结构
- 10.2.9 组件--回顾UML类图来改进操作特征
- 10.3 检查点
 - 10.3.1 我们讲了什么
 - 10.3.2 我们将要讲什么
- 第11章 从UML类图生成代码（第1部分）
 - 11.1 目标
 - 11.2 构造阶段
 - 11.2.1 Synergy过程
 - 11.2.2 可视化建模--可视化建模工具在项目中的任务
 - 11.2.3 可视化建模--可视化建模工具在程序代码生成方面的任务
 - 11.2.4 回顾有关准备生成程序代码的安装问题
 - 11.2.5 修改代码生成参数
 - 11.2.6 为组件指定类
 - 11.2.7 从可视化建模工具生成第一段代码
 - 11.2.8 从可视化建模工具生成其余代码--数据转化服务
 - 11.2.9 从可视化建模工具生成其余代码--商业规则服务
 - 11.2.10 从可视化建模工具生成其余的代码--表示服务
 - 11.2.11 回顾在代码生成完成之后需要注意的事项
 - 11.2.12 探讨如何将程序代码逆向工程到可视化模块图中
 - 11.3 添加代码来实现一条用例路径
 - 11.3.1 要支持简单的从头到尾的事务所必须添加的代码
 - 11.4 数据访问服务层：DASVC组件
 - 11.4.1 连接到数据源并执行选择查询
 - 11.4.2 关闭与数据源的连接
 - 11.4.3 连接到数据源并执行插入、更新或删除查询
 - 11.5 数据转化服务层：DTSVC组件
 - 11.5.1 建立要由数据访问服务层执行的SQL
 - 11.6 商业规则服务层：BRSVC组件
 - 11.6.1 建立控制过程的规则
 - 11.7 表示服务层：UISVC组件
 - 11.7.1 用户看到什么：将用户接口与商业规则服务层连结
 - 11.8 为将来创建代码块
 - 11.9 检查点
 - 11.9.1 我们讲了什么
 - 11.9.2 我们将要讲什么
- 第12章 由UML类图生成代码（第2部分）
 - 12.1 目标
 - 12.2 构造阶段
 - 12.2.1 增强顾客查询以及介绍浅对象和扩充对象的概念
 - 12.2.2 对顾客关系查询所做的代码修改
 - 12.2.3 为了支持扩展对象所做的代码修改
 - 12.2.4 使用户接口更加简单：用户定义类型
 - 12.2.5 客户端对象和非客户端对象
 - 12.2.6 伴随分布式应用的出现而产生的干扰趋势
 - 12.2.7 从用户接口获取信息时后端进行更新

<<使用Visual Basic和UML开>>

12.2.8 保持对象

12.3 检查点

12.3.1 我们讲了什么

12.3.2 我们将要讲什么

第13章 创建一个分布式应用：DCOM和MTS

13.1 目标

13.2 构造阶段

13.2.1 Synergy过程模型

13.2.2 构造--分布式应用程序：好还是坏

13.2.3 构造--Remulak Productions分布策略--偿付时间

13.3 远程解决方案--分布式组件对象模型

13.3.1 构造--为DCOM分布准备组件

13.3.2 构造--分布服务器组件

13.3.3 构造--在服务器上安装组件

13.3.4 构造--客户端准备测试DCOM安装

13.3.5 构造--创建客户端安装包

13.4 远程解决方案--Microsoft Transaction Server

13.4.1 构造--获得接口

13.4.2 构造--事务类型

13.4.3 构造--Remulak Productions事务类型

13.4.4 构造--MTS管理

13.4.5 构造--修改Remulak Productions来使用 MTS事务管理

13.4.6 构造--支持投票权力

13.4.7 构造--Remulak修改：给与取

13.4.8 MTS--第一轮修改

13.4.9 MTS--第二轮修改

13.4.10 MTS--第三轮修改

13.4.11 构造--事务管理

13.5 检查点

13.5.1 我们讲了什么

13.5.2 我们将要讲什么

第14章 可选接口：Internet

14.1 目标

14.2 构造阶段

14.2.1 Synergy过程模型

14.2.2 Web的角色

14.2.3 Web技术

14.2.4 组件重配置

14.2.5 完成Web组件

14.2.6 订单查询HTML表单

14.2.7 活动服务器网页 (ASP)

14.2.8 使用JavaScript的更加动态的客户端

14.2.9 其他可能性

14.3 检查点

14.3.1 我们讲了什么

附录A 使用用倒来估算项目

附录B 为Rational Rose数据定义语言添加补充功能

<<使用Visual Basic和UML开>>

附录C 面向对象入门

附录D 组件对象模型和COM+

附录E 面向对象的项目计划

附录F 项目输出示例

参考书目

推荐读物

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>