

<<UML精粹>>

图书基本信息

书名：<<UML精粹>>

13位ISBN编号：9787121170492

10位ISBN编号：7121170493

出版时间：2012-6

出版时间：电子工业出版社

作者：福勒

页数：207

字数：397000

译者：潘加宇

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>

<<UML精粹>>

内容概要

在紧迫的时间压力下，程序员很难通过紧跟技术创新的脚步来享受软件工程领域的最新成果。正因如此，殿堂级大师Martin Fowler这本旨在帮助专业人士提升效率的图书一问世，立即给软件工程社区带来巨大震撼。其前两版经年畅销，因行文精炼自然、见解透彻而广受称赞。被誉为以软件设计为生者必备资料的第3版，更是带来有效面向对象设计的最佳思想及愈发便利的教学体例，引进交互概述图、时间图、组合结构等新图型，对类图、顺序图、状态图、活动图等也有较多更新。

本书适合作为计算机、电子、通信等专业本科及研究生课程教材，对软件开发人员及专业研究者也极具参考价值。

<<UML精粹>>

作者简介

作者：（美国）福勒（Martin Fowler）译者：潘加宇

书籍目录

插图目录

第3版前言

第1版前言

序

为什么要花时间学UML

本书结构

第3版的变化

致谢

第1章 简介

1.1 UML是什么

1.2 使用UML的方式

1.3 UML诞生史

1.4 表示法和元模型

1.5 UML图

1.6 什么是合法的UML

1.7 UML的含义

1.8 仅有UML是不够的

1.9 何处开始UML

1.10 更多资料

第2章 开发过程

2.1 迭代和瀑布过程

2.2 预测性和自适应计划

2.3 敏捷过程

2.4 Rational统一过程

2.5 为项目裁剪过程

2.6 为过程裁剪UML

2.6.1 需求分析

2.6.2 设计

2.6.3 文档

2.6.4 理解遗留代码

2.7 选择开发过程

2.8 更多资料

第3章 类图：基础

3.1 性质

3.1.1 属性

3.2.2 关联

3.2 多重性

3.3 性质的编程解释

3.4 双向关联

3.5 操作

3.6 泛化

3.7 注解符和注释

3.8 依赖

3.9 约束规则

3.10 何时使用类图

<<UML精粹>>

3.11 更多资料

第4章 序列图

4.1 创建和删除参与者

4.2 循环、条件等

4.3 同步和异步调用

4.4 何时使用序列图

第5章 类图：进阶概念

5.1 关键词

5.2 责任

5.3 静态操作和属性

5.4 聚合和组合

5.5 派生性质

5.6 接口和抽象类

5.7 只读和冻结

5.8 引用对象和值对象

5.9 限定关联

5.10 分类和泛化

5.11 多重和动态分类

5.12 关联类

5.13 模板（参数化）类

5.14 枚举

5.15 主动类

5.16 可见性

5.17 消息

第6章 对象图

6.1 何时使用对象图

第7章 包图

7.1 包和依赖

7.2 包的分解

7.3 实现包

7.4 何时使用包图

7.5 更多资料

第8章 部署图

8.1 何时使用部署图

第9章 用例

9.1 用例的内容

9.2 用例图

9.3 用例的级别

9.4 用例和特性（或故事）

9.5 何时使用用例

9.6 更多资料

第10章 状态机图

10.1 内部活动

10.2 活动状态

10.3 超状态

10.4 并发状态

10.5 实现状态图

<<UML精粹>>

- 10.6 何时使用状态图
- 10.7 更多资料
- 第11章 活动图
 - 11.1 分解一个动作
 - 11.2 分区
 - 11.3 信号
 - 11.4 令牌
 - 11.5 流和边
 - 11.6 针脚和变换
 - 11.7 扩展区域
 - 11.8 流结束
 - 11.9 结合规格
 - 11.10 其他更多内容
 - 11.11 何时使用活动图
 - 11.12 更多资料
- 第12章 通信图
 - 12.1 何时使用通信图
- 第13章 组合结构
 - 13.1 何时使用组合结构
- 第14章 组件图
 - 14.1 何时使用组件图
- 第15章 协作
 - 15.1 何时使用协作
- 第16章 交互概述图
 - 16.1 何时使用交互概述图
- 第17章 时间图
 - 17.1 何时使用时间图
- 附录A UML版本之间的变化
 - A.1 对UML的修订
 - A.2 《UML精粹》中的变化
 - A.3 从UML1.0到1.1的变化
 - A.3.1 类型和实现类
 - A.3.2 完整和不完整的鉴别器约束
 - A.3.3 组合
 - A.3.4 不可变和冻结
 - A.3.5 序列图上的返回
 - A.3.6 术语“角色”的使用
 - A.4 从UML 1.2 (和1.1) 到1.3 (和1.5) 的变化
 - A.4.1 用例
 - A.4.2 活动图
 - A.5 从UML 1.3到1.4的变化
 - A.6 从UML 1.4到1.5的变化
 - A.7 从UML 1.x到UML 2.0
 - A.7.1 类图：基础 (第3章)
 - A.7.2 序列图 (第4章)
 - A.7.3 类图：概念 (第5章)
 - A.7.4 状态机图 (第10章)

<<UML精粹>>

A.7.5 活动图 (第11章)

参考文献

索引

插图目录

图1.1 UML元模型的小片段

图1.2 UML图形类型分类

图1.3 Wiki的一部分的非正式屏幕流图 (<http://c2.com/cgi/wiki>)

图3.1 一个简单的类图

图3.2 展示订单的性质为属性

图3.3 展示订单的性质为关联

图3.4 双向关联

图3.5 使用动词短语命名关联

图3.6 注解符用做一个或多个图形元素的注释

图3.7 依赖的例子

图4.1 一张中央控制的序列图

图4.2 分布控制的序列图

图4.3 创建和删除参与者

图4.4 交互框

图4.5 旧习惯表达控制逻辑

图4.6 CRC卡样例

图5.1 在类图中展示责任

图5.2 静态表示法

图5.3 聚合

图5.4 组合

图5.5 time period内的派生属性

图5.6 接口和抽象类的Java例子

图5.7 小球-球窝表示法

图5.8 老的带依赖的棒棒糖表示法

图5.9 使用棒棒糖表示法在序列图中展示多态

图5.10 限定关联

图5.11 多重分类

图5.12 关联类

图5.13 提升关联类为完整的类

图5.14 关联类的细微差别 (角色不应该是关联类)

图5.15 用一个类表达瞬时关系

图5.16 为关联使用 ?temporal? 关键词

图5.17 模板类

图5.18 绑定元素 (版本1)

图5.19 绑定元素 (版本2)

图5.20 枚举

图5.21 主动类

图5.22 带消息的类

图6.1 Party组合结构的类图

图6.2 展示Party实例的对象图

图7.1 在图上展示包的方式

图7.2 企业应用包图

图7.3 把图7.2分离成两个部分

<<UML精粹>>

- 图7.4 被其他包实现的包
- 图7.5 在客户包中定义一个需求接口
- 图8.1 部署图实例
- 图9.1 用例文本实例
- 图9.2 用例图
- 图10.1 一个简单的状态机图
- 图10.2 一个文本域的typing (输入中) 状态的内部事件展示
- 图10.3 带活动的状态
- 图10.4 带有嵌套子状态的超状态
- 图10.5 并发正交状态
- 图10.6 处理图10.1状态转换的C#嵌套switch
- 图10.7 图10.1的状态模式实现
- 图11.1 一个简单的活动图
- 图11.2 一张次级的活动图
- 图11.3 修改图11.1的活动为调用图11.2的活动
- 图11.4 活动图上的分区
- 图11.5 活动图上的信号
- 图11.6 发送和接收信号
- 图11.7 展示边的4种方式
- 图11.8 流上的变换
- 图11.9 扩展区域
- 图11.10 扩展区域中的单个动作的速记
- 图11.11 活动中的流结束
- 图11.12 结合规格
- 图12.1 中央控制的通信图
- 图12.2 嵌套小数编号的通信图
- 图13.1 展示TV Viewer及其接口的两种方式
- 图13.2 组件的内部视图 (来自Jim Rumbaugh的例子)
- 图13.3 带多个端口的组件
- 图14.1 组件的表示法
- 图14.2 组件图例子
- 图15.1 带角色类图的协作
- 图15.2 拍卖协作的序列图
- 图15.3 协作发生
- 图15.4 展示JUnit(junit.org)中模式使用的非标准方式
- 图16.1 交互概述图
- 图17.1 展示状态为线的时间图
- 图17.2 展示状态为区域的时间图

章节摘录

版权页：插图：认识到奥卡姆剃刀的价值后，当我在做架构和阅读时，我会寻找遵从吝啬定律的项目和书。

因此，我为你正在阅读的这本书喝彩。

你一开始可能会发现我刚才的评价令人吃惊。

我经常被人与内容多而密的定义统一建模语言（UML）的规则联系起来。

这些规则允许工具厂商实现UML，允许方法学家应用UML。

7年以来，我已经作为主席主持大型国际标准化团队制定UML1.1和UML2.0规则，以及若干版本之间小的修订。

在这个时期，UML在表现力和准确性上已经成熟；但作为标准化过程的一个结果，也添加了不必要的复杂性。

遗憾的是，标准化过程更倾向于妥协，以符合委员会设计的结果，而不是倾向于优雅的吝啬。

作为一名UML专家，我已经熟知规则中隐密的细枝末节，还能从Martin对UML 2.0的提炼中学到东西吗？

和你一样，可以学到很多。

一开始，Martin熟练地把一门复杂的大型语言削减为一个实用的子集，这个子集经他的实践证明是高效的。

他拒绝走更容易的路线：在他的书的上一个版本上加上附页。

因为语言发展了，Martin要忠于他的目标：寻求“最有用的UML组成部分”，而且就告诉你这部分。

他给出的这部分就是UML中那神秘的20%（能帮助你完成80%的工作）。

捕获和驯服UML这头善于躲避的野兽，是了不起的成就！

更让人印象深刻的是，Martin以极其吸引人的口语化风格达成了这个目标。

通过和我们分享他的观点和轶事，使得这本书读起来很有趣，并提醒我们架构和设计系统应该是创造性和生产性的。

如果我们追寻吝啬心法的完整意图，我们应该会发现UML建模项目就像我们在初中时上指画班一样令人愉快。

UML应该是我们吸收创造性闪电的一根避雷针，同时也是精确详述系统蓝图的一束激光，这样，第三方可以投标和建造那些系统，这是判断一门语言是否是真正的蓝图语言的酸性测试。

因此，虽然这是一本很薄的书，但绝不平凡。

你既可以学到Martin的建模方法，也可以学到他对UML 2.0的解释。

我很享受和Martin一起工作，挑出这个修订版本要解释的UML 2.0语言的特性并修正以前的错误。

我们需要记住，所有现存的语言，包括自然语言和人造语言，必须不断进化，要不就消亡了。

Martin对新特性的选择，以及你和其他从业者的偏好，是UML修订过程中的关键部分。

你们让这门语言保持生命力，帮助它通过市场的自然选择进化。

<<UML精粹>>

编辑推荐

《UML精粹:标准对象建模语言简明指南(第3版)》已经向成千上万的开发人员介绍了UML, 激发了他们进一步探索用现在这种标准建模语言建模的诸多好处。
不管您是第一次对UML感兴趣, 还是希望了解UML在开发过程中扮演的关键角色。

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>