

<<统一建模语言UML>>

图书基本信息

书名：<<统一建模语言UML>>

13位ISBN编号：9787302194385

10位ISBN编号：7302194386

出版时间：2009-5

出版时间：清华大学

作者：袁涛//孔蕾蕾

页数：107

字数：181000

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>

<<统一建模语言UML>>

内容概要

本书是学习面向软件分析与设计的和应用UML 2.0的教材。

它不仅详细阐述了UML在建模活动中的基本应用方法，而且对UML的建模图示在软件生命周期中的应用进行了分类。

在介绍UML

10种最为重要的语言图示时，强调该UML图示在软件建模活动中的建模目的，并根据UML各种图示的语法结构详细解释该图示在实际建模中的不同表示形式和语法，最后应用UML图示对一个贯穿全书的真实软件工程项目实例进行建模示范，使读者在深入理解UML语义、语法和图示法的同时，能牢牢把握住学习该UML图示的目的和意义。

本书可作为高等学校计算机、电子、通信等专业高年级学生及研究生课程教学用书，同时对软件研究者和开发人员亦颇具参考价值。

<<统一建模语言UML>>

书籍目录

第1章 导言

- 1.1 模型
- 1.2 开发软件为什么需要模型
- 1.3 什么是统一建模语言
- 1.4 UML的发展史
- 1.5 UML的建模对象
 - 1.5.1 UML的结构模型
 - 1.5.2 UML的行为模型
- 1.6 总结

第2章 用例图

- 2.1 基于用例的系统行为建模
- 2.2 用例图
- 2.3 用例图的表示方法
 - 2.3.1 参与者
 - 2.3.2 用例
 - 2.3.3 用例之间的关系
- 2.4 总结

第3章 对象图

- 3.1 基于对象的系统瞬间状态建模
- 3.2 对象图
- 3.3 对象图的表示方法
 - 3.3.1 对象
 - 3.3.2 链
- 3.4 总结

第4章 顺序图

- 4.1 基于交互的对象行为建模：交互时的行为顺序
- 4.2 顺序图
- 4.3 顺序图的表示方法
 - 4.3.1 生命线
 - 4.3.2 活动条
 - 4.3.3 消息
 - 4.3.4 交互框
- 4.4 案例分析
- 4.5 总结

第5章 通信图

- 5.1 基于交互的对象行为建模：交互时的对象结构
- 5.2 通信图
- 5.3 通信图的表示方法
 - 5.3.1 交互的参与者
 - 5.3.2 链接
 - 5.3.3 消息
- 5.4 案例分析
- 5.5 总结

第6章 类图

- 6.1 基于类的系统结构建模

<<统一建模语言UML>>

6.2 类图

6.3 类图的表示方法

6.3.1 表示类

6.3.2 类的关系

6.4 总结

第7章 状态图

7.1 基于状态的对象行为建模

7.2 状态图

7.3 状态图的表示方法

7.3.1 状态

7.3.2 迁移

7.4 案例分析

7.5 总结

第8章 活动图

8.1 基于活动的系统行为建模

8.2 活动图

8.3 活动图的表示方法

8.3.1 活动和动作

8.3.2 活动边

8.3.3 活动节点

8.3.4 活动划分或泳道

8.3.5 调用其他活动

8.4 案例分析

8.5 总结

第9章 包图

9.1 基于包的系统静止状态下的结构建模

9.2 包图

9.3 包图的表示方法

9.3.1 包

9.3.2 包中元素的可见性

9.3.3 包之间的关系

9.4 总结

第10章 构件图

10.1 基于构件的系统静止状态下的结构建模

10.2 构件和构件图

10.2.1 构件

10.2.2 构件图

10.3 构件图的表示方法

10.3.1 构件

10.3.2 供接口和需接口

10.3.3 构件间的关系

10.3.4 实现构件的类

10.3.5 外部接口——端口

10.3.6 连接器

10.3.7 显示构件的内部结构

10.4 总结

第11章 部署图

<<统一建模语言UML>>

11.1 基于物理环境的系统执行状态下的结构建模

11.2 部署图

11.3 部署图的表示方法

11.3.1 制品

11.3.2 节点

11.3.3 部署

11.3.4 部署规约

11.3.5 通信路径

11.4 总结与强调

附录A UML的扩展机制

附录B PPS项目的部分主要用例的用例规约

术语英汉对照表

参考文献

章节摘录

第1章 引言 1.2 开发软件为什么需要模型 在开发软件的过程中，开发者在动手编写程序之前需要研究和分析软件的诸多复杂和纷乱的问题。

例如，用户需求的准确描述问题、功能与功能之间的关系问题、软件的质量和性能问题、软件的结构组成问题、建立几十个甚至几百个程序或组件之间的关联问题等。

所以，软件系统的开发是一个非常复杂的过程，它们的复杂程度不比任何一项大型的复杂土木建设工程逊色。

但是，在这个复杂的开发过程中，我们最关注的还是开发者之间的交流问题。

软件开发中消除技术人员与非技术人员（用户）之间、使用不同技术的开发人员之间、不同功能使用者之间等交流障碍是软件开发成功的关键。

直观的软件模型将有助于软件工程师与他们进行有效地交流。

在软件的需求分析中，用户和系统所属领域的专家更熟悉将要构建系统的功能，我们称他们为领域专家（Domain Expert）。

他们提出软件系统在这个领域中所需要具有的功能。

所以，软件设计者可以通过建立需求模型来实现技术人员与非技术人员（用户）之间的交流。

在软件的设计中，设计人员首先要把描述系统功能需求的自然语言形式转化为软件程序的形式，在这个转化过程中，设计人员要借助许多模型来完成最终的程序设计模型。

这些中间辅助模型包括系统的行为模型、对象的状态和行为模型等。

如果这些模型都是严格遵循统一建模语言标准而建立的，那么，无论开发人员具有多么不同的开发条件和技能，他们都可以理解软件设计，并且进行有效交流。

在软件的实施、测试和部署中，模型为不同领域的技术人员在软件和硬件的实施、测试和部署中提供有效的交流平台。

最后，要强调的是，在各种各样的软件中，软件模型是最有效的软件文档保存形式，软件模型在开发团队人员的培训、学习和知识的传递与传播等方面起着非常重要的作用。

所以，软件开发中需要建立需求（Requirement）模型、问题域（Domain）模型、设计（Design）模型、实施（Implementation）模型、测试（Test）模型和部署（Deployment）模型。

可见，在系统开发生命周期中，需要从多角度来建立模型才能全面、准确地分析和设计软件系统。

<<统一建模语言UML>>

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>