

<<UML嵌入式设计>>

图书基本信息

书名：<<UML嵌入式设计>>

13位ISBN编号：9787302163107

10位ISBN编号：7302163103

出版时间：2008-1

出版时间：清华大学

作者：高焕堂 编

页数：463

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嵌入式设计>>

前言

序言当全球快速迈向数字家电、汽车电子化、M化生活之际，软硬件整合设计(HW-SW Co-Design)蔚为风潮。

像IPod和WII等数码产品都是此波软硬整合设计的代表作。

由于软硬整合设计的强烈需求，当前嵌入式系统开发已成为主流。

上述软硬件整合设计的基本目标是： 软硬整合设计； 软硬分开生产； 软硬独立成长。

虽然软硬整合设计的概念已经存在多年了，但是大多偏向从硬件看软件的观点，认为软件比较软，像树叶一般；而硬件较硬，像树干一般。

其并未兼顾软硬的平衡感，导致软件对于硬件组件的演进和调换性并未做出应有贡献。

于是有必要增添另一个新观点。

新观点的特色是：从改变(Change)上看，树叶长大之后就不易改变了，所以每年都必须蜕变换新，就像硬件的迅速更换；而树干则是柔软地、不断地、局部地持续改变与成长，就像软件一样，容易局部修改、持续成长。

总而言之，新观点就是：“软件是树干，硬件是树叶。

”基于此软硬平衡的新观点，让软硬件整合设计确实达成整体系统的和谐感。

本书深入浅出地阐述以上观点，并以平易近人的实例说明应用途径。

盼望您在本书的陪伴之下，进行Learn by doing，祝福您的创意和能力都能因而突飞猛进，自由翱翔，无远弗届。

高焕堂 谨识于台北2007.5.15

<<UML嵌入式设计>>

内容概要

UML是当今最流行的标准建模语言，它与嵌入式系统设计愈来愈密切。

本书深入浅出地介绍了两者之间的密切关系，并通过精彩的实例讲解，让读者迅速掌握UML建模技能，并能迅速进入当今科技行业最热门的领域。

本书共分7章，第1章首先介绍了嵌入式系统建模的特点及其效益，然后介绍如何将UML迅速而有效地应用于嵌入式建模上；第2章是建立系统模型的基础技术；第3~6章详细介绍了建模的高级技术，并以详尽的实例解说来提升读者建模的技能及其应用能力；第7章以翔实的案例深入讲解如何将专家的设计模式应用于各式各样的嵌入式和实时系统开发上。

本书着重于传授作者多年的实践经验，使读者能够运用最流行的UML技术进行嵌入式系统设计工作。

因此，本书可作为高等院校UML。

课程的教材，对初中级嵌入式/实时系统程序员以及准备学习嵌入式系统设计与建模的读者具有较高的参考价值。

<<UML嵌入式设计>>

作者简介

高焕堂，台湾软件架构设计大师，从事IT行业近30年，台湾oo技术的代表人物。
现任MISOO软件开发与管理顾问公司首席架构师，编著过十余本软件技术相关书籍。
多年来一直与UML China密切联系，与大陆软件人员分享软件“设计”的观点、技术和哲理。

<<UML嵌入式设计>>

书籍目录

第1章 嵌入式软件新途径——目件化设计 1.1 迎接软、硬件整合设计时代的来临 1.2 嵌入式系统设计：组件化新思维 1.3 重视组件的依赖 (Dependency) 1.3.1 组件依赖至关重要 1.3.2 管理软件组件的依赖 1.4 认识软件组件 1.4.1 何谓组件 1.4.2 软件组件的特性 1.5 组件化软件的目标：有机次序 1.6 组件化软件的手段：对象 (OO) 技术 1.7 认识对象模型 (Object Model) 1.7.1 什么是模型？ 1.7.2 为什么需要对象模型？ 1.7.3 全方位的对象模型：以EA工具为例 1.8 对象建模语言——UML 1.8.1 设计师共通的表达方式——建模语言 1.8.2 UML的发展 1.8.3 UML的基本元素 1.8.4 UML的图示 1.9 简介UML开发工具 1.9.1 Sparx公司的EA (Enterprise Architecture) 1.9.2 免费的ArgoUML 1.9.3 I-Logix公司的Rhapsody 1.9.4 IMB公司的RSA (Rational Software Architect) 1.10 从UML到C：以电冰箱为例 1.11 从UML到C++：以电冰箱为例 1.12 从UML到Java：以电冰箱为例 1.13 以C编写面向对象程序 1.13.1 对象的行为 (Behaviour) 1.13.2 消息与运算 1.13.3 着手编写OOPC：程序 1.13.4 对象的分类 1.13.5 定义继承关系 1.14 关于本书的范例程序第2章 UML如何表达组件的沟通 2.1 “数据交换”沟通方法 2.2 “消息传递”沟通方法 2.3 练习“消息传递”表示法 2.4 设计实例探讨 2.4.1 需求分析 (Analysis) 2.4.2 软件组件规划 2.4.3 类设计 2.4.4 定义对象 2.4.5 分析Use Case 2.4.6 安排对象的协作 2.4.7 模拟设计 2.5 C程序实现示范 2.6 C++程序实现示范 2.7 Java程序实现示范第3章 UML如何表现Whole-Part结构 3.1 组件化与Whole-Part关系 3.2 整体/部分 (Whole-Part) 结构关系 3.2.1 组合/部分关系 3.2.2 容器/内容关系 3.2.3 集合/成员关系 3.3 UML如何表达Whole-Part关系 3.3.1 Aggregation与Composition 图示 3.3.2 Odell对Whole-Part关系的分类 3.4 软件Whole-Part结构设计 3.4.1 简单的Whole-Part关系 3.4.2 递归式Whole-Part关系 3.4.3 Whole-Part结构的类设计 3.5 设计实例探讨 3.5.1 需求分析 (Analysis) 3.5.2 软件组件规划 3.5.3 类设计 3.5.4 定义对象 3.5.5 分析Use case 3.5.6 安排对象的协作 3.5.7 模拟设计 3.6 C程序实现示范 3.7 C++程序实现示范 3.8 Java程序实现示范第4章 UML如何表示组件状态的变化 4.1 组件的行为 (Behavior) 4.2 状态 (State) 与行为 (Behavior) 4.3 组件状态的转换 4.4 状态转换触发的行为 4.5 如何设计状态 4.6 设计实例探讨 (一) 4.6.1 C程序实现示范 4.6.2 C++程序实现示范 4.6.3 Java程序实现示范 4.7 设计实例探讨 (二) 4.7.1 状态分析与设计 4.7.2 C程序实现示范 4.7.3 C++程序实现示范 4.7.4 Java程序实现示范 4.7.5 状态图设计练习：交通信号灯的控制第5章 UML如何实现Use Case 5.1 Use Case：收益与成本的交会点 5.2 从3个角度看Use Case 5.2.1 从User角度看 5.2.2 从User与系统的交互角度看 5.2.3 从系统内部组件的协作角度看 5.3 如何分解Use Case 5.4 Use Case模型的角色 5.5 如何编写Use Case描述 5.6 以Use Case表达创新功能 5.7 表达Use Case之间的关系 5.8 设计实例探讨 5.8.1 需求分析 (Analysis) 5.8.2 组件分解 5.8.3 类设计 5.8.4 内观组件协作 5.8.5 状态 (State) 设计 5.9 C程序实现示范 5.10 C++程序实现示范 5.11 Java程序实现示范第6章 UML如何表达组件的活动 6.1 Why活动图 (Activity Diagram) 6.2 活动分配 (Assignment) 6.3 细说活动图 6.4 设计实例探讨 6.4.1 需求分析 (Analysis) 6.4.2 软件组件规划 6.4.3 分析Use Case 6.4.4 活动分配 (即类设计) 6.4.5 设计序列图 6.5 C程序实现示范 6.6 C++程序实现示范 6.7 Java程序实现示范第7章 UML如何表达设计模式 7.1 何谓模式 7.2 认识设计模式 7.3 设计模式的起源 7.4 简介3个软件模式 7.4.1 Facade模式 7.4.2 Chain Of Responsibility模式 7.4.3 State模式 7.5 设计实例探讨 7.5.1 需求分析 (Analysis) 7.5.2 运用State模式 7.5.3 GUI设计 7.6 C程序实现示范 7.7 C++程序实现示范 7.8 Java程序实现示范参考资料

<<UML嵌入式设计>>

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>