

《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

出版社：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以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.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嵌入式设计》特色：《UML嵌入式设计(附光盘1张)》针对目前嵌入式开发人员的广泛需求，将UML的应用与嵌入式相结合进行深入剖析。着重于传授作者多年的实战经验，教会读者运用最流行的UML技术，并通过多个翔实的开发案例来帮助读者做好嵌入式系统的设计工作。

《UML嵌入式设计(附光盘1张)》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。

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 (Behavior)	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
---------------------	---------------------	--------------------	--------------------------	----------------	-----------------	------------	------------	---------------	-------------------	-------------------------	---------------------------	--------------	------------------	------------------------	-----------------	------------------------	--------------	----------------	--------------	---------------	--	------------------	--------------------------	---	--------------------	----------------------	-----------------------	-----------------	-------------------------	--------------	--------------------	--------------	---------------	----------------	------------------	----------------	----------------	-----------------	------------	-----------------------	--------------	-----------	------------	------------------	---------------	------------	-------------	---------------	----------------	-------------------------	----------------------	-----------------------------	---------------	---------------	---------------	-------------------------	---------------------------------	-----------------------------	----------------------	-----------------------	-----------------------	------------------------	------------	-----------------------	--------------	-----------	------------	------------------	---------------	------------	-------------	---------------	----------------	--------------------	----------------------	-------------------------------	-------------	---------------	------------	----------------	---------------	-----------------	------------------	----------------	---------------	---------------	-----------------	------------------	------------------------	---------------------	------------------------	--------------------	----------------	----------------------	---------------------	------------------	-------------------	--------------------	---------------------	---------------------	------------	-----------------------	------------	-----------	--------------	---------------------	-------------	----------------	-----------------	------------------	-------------------------------	-----------------------	-----------	------------	-----------------------	--------------	------------------	-------------------	-------------	-------------	---------------	----------------	-----------------	----------	------------	-------------	--------------	----------------	---------------------------------	---------------	------------	-----------------------	-----------------	-------------	-------------	---------------	--------------------

精彩短评

1、优点：一步步地介绍一些OOP的思想。缺点：既然是UML，书中的实例用的是EA但在却没有将其实例的UML图放到光盘中，让初学者或对EA不熟悉的人来说多少有点遗憾。书中至少四分之一的页数是代码，本可以放到随书送的光盘中，感觉这样做浪费了资源不说还给人不太道德的感觉。

2、对嵌入式建模来说，非常具有指导意义，本人非常喜欢，该书很通俗易懂，非常好！

3、其实在买这本书之前看到评论说不怎么样，但是仍然是怀着试试看的心情买了一本。因为从目录中来看，书中有OOPC相关的东西，最重要的是，UML和嵌入式的结合非常的诱人。

先说说书的客观情况吧。书本的印刷质量不错，捧在手里的感觉也还行，纸张比较光滑，做笔记的时候不会透墨。

用了一天左右的时间将这本书第一次拿下。不说深刻的，但从第一感觉来说，这本书不值得买。原因如下：· 印刷时行间距比较大，虽然书看起来比较厚，但是内容不是很多。· 虽然是介绍UML的书，但是书中对UML的介绍很少，如果使用UML作图以应用场景都没有详细的介绍。· 代码冗余现象严重。书附带的光盘中已经有了完整的代码示例，但是作者还是把他们印了出来。一章假设有十页的话，代码至少占4页。同一个实现采用了C C++ JAVA三种语言实现，我估计作者的真正意图是凑页数。· 很多人都吹捧作者，但是看完之后非常的方案。就从个人来看，也许作者在台湾，在UML界很有名，但是在书中没有看出一个大师的水平。书中举了众多的有中国特色的例子，但是没有一条主线，基本都是左一句，右一句。但是读完的感觉就是：哦，是这么回事儿，但是一点作用都没有。

当然，才读了一遍，可能没有发现书中的精华，等时间充足的话我也许会看第二遍。但是，就目前来看，不推荐大家买这本书。

4、这本书找了好长时间,正是我想要的。

5、台湾作家写的，值很一看，不像大陆拼写数

6、大师级作品。作者对运用UML开发嵌入式系统的进行了深入浅出的叙述，见解独到，案例翔实，可操作性强。只是每个案例给出三种语言，书中附有大量的源代码，有凑数之嫌。选一种语言，其他放在光盘上即可。

7、但是还没来得及仔细看，UML要用起来并不是简单的事，虽然本身很简单

8、深入浅出，值得收藏

9、内容比较基础，介绍的也很详细通过实例介绍了一个简单项目的设计流程和注意事项只是书中源代码太多了，更雷人的是竟然贴上了三种开发语言的代码！！既然有送光盘，何必在书上copy这么完整的代码，重点在哪？难道连作者自己也不知道？？

10、学习UML的嵌入式人员，提升个人能力的好书

11、浅显易懂，很容易入手

12、大师在软件设计中，哲学的运用让我非常敬佩，就是本书的代码太占地地方了。

13、用这本书来做的uml入门

14、满眼全是代码,那要光盘是干啥的?

15、概念解释的很透彻

16、这本书写得真的不怎么样

17、看完后，很失望。这本叫作《UML嵌入式设计》的书既没有UML，也没有嵌入式设计。此类作者远离实际开发太久了，满篇空话。书中的例子翻来覆去就是红绿灯、手电筒。这种水平又怎么好意思出书，浪费别人时间

18、印刷质量好，使用价值高

19、觉得一般，也可能是我理解力不够吧。不建议初学者买

《UML嵌入式设计》

精彩书评

- 1、个人觉得这本书把问讲得很简单易懂，但有点显得过于啰嗦，同一个问题说来说去却缺少点精之笔。读完后给你感觉书本来用百多两百可以说得很清楚了，但却硬是用代码（无注释，无风格的代码）撑到463页。代码本可以放到光盘中或网上的。没觉得不可思意的是，书中大量的UML图在光盘中却没有配上原稿。
- 2、虽然合适的例子能起到“深入浅出”的效果，但是作者举的例子未必太简单了。用了粗略的说明UML是干什么的也许是可行的，但是想达到叫人体会UML在嵌入式（请注意与一般纯软件不同）设计上的特点和优点，显然是不够的。对于学习UML的人来说，或者保守点，就是对于我来说，拿一些非常简单的，比如手电筒、买肯德基这样的例子，我肯定也能分析的头头是道，但是我对如何用UML做一个真正的嵌入式系统设计，还是不知如何下手（这也正是我现在没有在自己负责的固件开发项目中使用UML设计的缘故）。我觉得作者应该拿一些真实案例进行剖析（可忽略细节，突出抽象本质），那样才会予人启迪，使人信服。另外，大段的代码不仅仅是增加篇幅，而且几乎没有体现嵌入式系统的特点。我没有怎么看Java的代码，不好妄自评论。我只看了C语言部分，没有看出这些代码跟桌面机Windows下编程有什么区别，也许有一点可以算得上，那就是一点注释都没有（这一点，有位读者已经说了，并认为是缺点，其实还真算不上，因为基于作者这么简陋的例子，确实没有必要注释了）。我不知道作者是否真的做过嵌入式研发，如果没有独特的心得，那么还不如直接改成为“UML设计入门”，那样，这本书也许还真的不错。
- 3、这是我看过的，有关uml写的最好的一本书，简单扼要，非常清楚。唯一不足的地方，就是代码完全可以放到光盘里，没必要用来增加书本的厚度，赚取稿费！

《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:www.tushu000.com