

<<架构之美>>

图书基本信息

书名：<<架构之美>>

13位ISBN编号：9787564122737

10位ISBN编号：7564122730

出版时间：2010-6

出版时间：东南大学

作者：(希)斯皮内利斯//郭西奥斯

页数：404

字数：663000

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>

前言

THE IDEA FOR THE BOOK YOU'RE READING WAS CONCEIVED IN 2007 as a successor to the award-winning, best-selling Beautiful Code, a collection of essays about innovative and sometimes surprising solutions to programming problems. In Beautiful Architecture, the scope and purpose is different, but similarly focused: to get leading software designers and architects to describe a software architecture of their choice, peeling back the layers of their creations to show how they developed software that is functional, reliable, usable, efficient, maintainable, portable, and, yes, elegant. To put together this book, we contacted leading architects of well-known or less-well-known but highly innovative software projects. Many of them replied promptly and came back to us with thought-provoking ideas. Some of the contributors even caught us by surprise by proposing not to write about a specific system, but instead investigating the depth and the extent of architectural aspects in software engineering. All chapter authors were glad to hear that the work they put in their chapters is also helping a good cause, as the royalties of this book are donated to Meddcins Sans FrontiOres (Doctors Without Borders), an international humanitarian aid organization that provides emergency medical assistance to suffering people.

<<架构之美>>

内容概要

《架构之美》围绕5个主题领域来组织《架构之美》的内容：概述、企业应用、系统、最终用户应用和编程语言。

《架构之美》让最优秀的设计师和架构师来描述他们选择的软件架构，剥开架构的各层，展示他们如何让软件做到实现功能、可靠、易用、高效率、可维护、可移植和优雅。

<<架构之美>>

作者简介

编者：（希腊）斯皮内利斯（Diomidis Spinellis）（希腊）郭西奥斯（Georgios Gousios）

<<架构之美>>

书籍目录

| | | | | | | |
|-------------------|------------------------|-------------------------|-------------------|---------------------------|-------------------------------|--------------------|
| 序前言 | 第一部分 论架构 | 第1章 什么是架构 | 1.1 简介 | 1.2 创建软件架构 | 1.3 架构结构 | 1.4 |
| 好的架构 | 1.5 美丽的架构 | 1.6 致谢 | 1.7 参考文献 | 第2章 两个系统的故事：现代软件神话 | | |
| 2.1 混乱大都市 | 2.2 设计之城 | 2.3 说明什么问题 | 2.4 轮到您了 | 2.5 参考文献 | 第二部分 企 | |
| 业级应用架构 | 第3章 伸缩性架构设计 | 3.1 简介 | 3.2 背景 | 3.3 架构 | 3.4 关于架构的思考 | |
| 第4章 记忆留存 | 4.1 功能和约束 | 4.2 workflow | 4.3 架构关注点 | 4.4 用户反应 | 4.5 结论 | |
| 第5章 面向资源的架构：在Web中 | 5.1 简介 | 5.2 传统的Web服务 | 5.3 Web | 5.4 面向资源的架构 | 5.5 数据驱动的应用 | 5.6 应用面向资源的架构 |
| 5.7 结论 | 第6章 数据增长：Facebook平台的架构 | 6.1 简介 | 6.2 创建一个社会关系Web服务 | 6.3 创建社会关系数据查询服务 | 6.4 创建一个社会关系Web门户：FBML | 6.5 系统的支持功能 |
| 6.6 总结 | 第三部分 系统架构 | 第7章 Xen和虚拟化之美 | 7.1 简介 | 7.2 Xenoservers | 7.3 虚拟化的挑战 | 7.4 半虚拟化 |
| 7.5 Xen的变换形式 | 7.6 改变的硬件，改变的Xen | 7.7 经验教训 | 7.8 延伸阅读 | 第8章 Guardian：一个容错操作系统环境 | 8.1 Tandem/16，将来所有的计算机都会像这样构建 | 8.2 硬件 |
| 8.3 机械布局 | 8.4 处理器架构 | 8.5 处理器间总线 | 8.6 输入/输出 | 8.7 进程结构 | 8.8 消息系统 | 8.9 文件系统 |
| 8.10 民间传说 | 8.11 弊端 | 8.12 后继者 | 8.13 延伸阅读 | 第9章 JPC：一个纯Java的x86PC模拟程序 | 9.1 简介 | 9.2 概念验证 |
| 9.3 PC架构 | 9.4 Java性能技巧 | 9.5 把4GB放入4GB：这不起作用 | 9.6 保护模式的危险 | 9.7 从事一项毫无成功希望的斗争 | 9.8 劫持JVM | 9.9 最终灵活性 |
| 9.10 最佳安全性 | 9.11 第二次做会更好 | 第10章 元循环虚拟机的力量：JikesRVM | 10.1 背景 | 10.2 与运行时环境相关的传言 | 10.3 JikesRVM简史 | 10.4 一个自足执行的运行时自举 |
| 10.5 运行时组件 | 10.6 经验教训 | 参考文献 | 第四部分 最终用户应用架构 | 第11章 GNU Emacs：滋长的特性是其优势 | 11.1 使用中的Emacs | 11.2 Emacs的架构 |
| 11.3 滋长的特性 | 11.4 另外两个架构 | 第12章 当集市开始构建教堂 | 12.1 简介 | 12.2 KDE项目的历史和组织结构 | 12.3 Akonadi | 12.4 ThreadWeaver |
| 第五部分 语言与架构 | 第13章 软件架构：面向对象与面向功能 | 13.1 概述 | 13.2 示例 | 13.3 面向功能解决方案的模块性评价 | 13.4 面向对象视图 | 13.5 面向对象模块性的评价和改进 |
| 13.6 代理：将操作封装到对象中 | 致谢 | 参考文献 | 第14章 重读经典 | 14.1 所有东西都是对象 | 14.2 类型是隐式定义的 | 14.3 问题 |
| 14.4 砖块和灰浆建筑 | 架构参考文献跋 | | | | | |

章节摘录

插图：Finally, we would want to include architectures that have features that delight the developers and testers who use the architecture and build it and maintain it, as well as the users of the system (s) built from it. Why delight developers ?

It makes their job easier and is more likely to result in a high-quality system. Why delight testers ?

They are the ones who have to attempt to emulate what the users will do as part of the testing process. If they are delighted, it is likely that the users will be, too. Think of the chef who is unhappy with his culinary creations.

His customers, who consume those creations, are likely to be unhappy, too. Different systems and application domains offer opportunities for architectures to exhibit specific delightful features, but Conceptual Integrity is a feature that cuts across all domains and that always delights. A consistent architecture is easier and faster to learn, and once you know a little, you can begin to predict the rest. Without the need to remember and handle special cases, code is cleaner and test sets are smaller. A consistent architecture does not offer two (or more) ways to do the same thing, forcing the user to waste time choosing. As Ludwig Mies van der Rohe said of good design, "Less is more," and Albert Einstein might say that beautiful architectures are as simple as possible, but no simpler. Given these criteria, we propose some initial candidates for our "Gallery of Beautiful Architectures."

<<架构之美>>

媒体关注与评论

“ 本书的作者们在介绍软件架构的基本实践和最佳实践方面干得很漂亮，他们也同样漂亮地介绍了各式各样的现代系统。

我特别喜欢他们谈及的架构的广泛性，从Emacs到Facebook，从非常正式的系统到非常有灵气的系统。简而言之，这是一本非常及时的书，对于软件架构的艺术、科学和实践是非常有益的贡献。

”
——GradyBooch，IBM院士

<<架构之美>>

编辑推荐

《架构之美(影印版)》：健壮、优雅、灵活和易维护的软件架构是怎样炼成的?《架构之美》通过一系列优秀的文章回答了这个问题，这些文章来自于十几位当今一流的软件设计师和架构师。在每篇文章中，作者都向们展示了一个著名的软件架构，并分析了什么让其具有创新性，让其极其符合设计目标。

《架构之美》内容包括：Facebook的架构如何建立在以数据为中心的应用生态系统之上。

Xen的创新架构对操作系统未来的影响。

KDE项目的社区过程如何让软件的架构从粗略的草图演进为漂亮的系统。

不断滋长的特征如何让GNU Emacs获得从未预料到的功能。

JikesRVM自优化、自足执行的运行时环境背后的魔法。

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>