



SEI软件工程译丛

软件构架实践

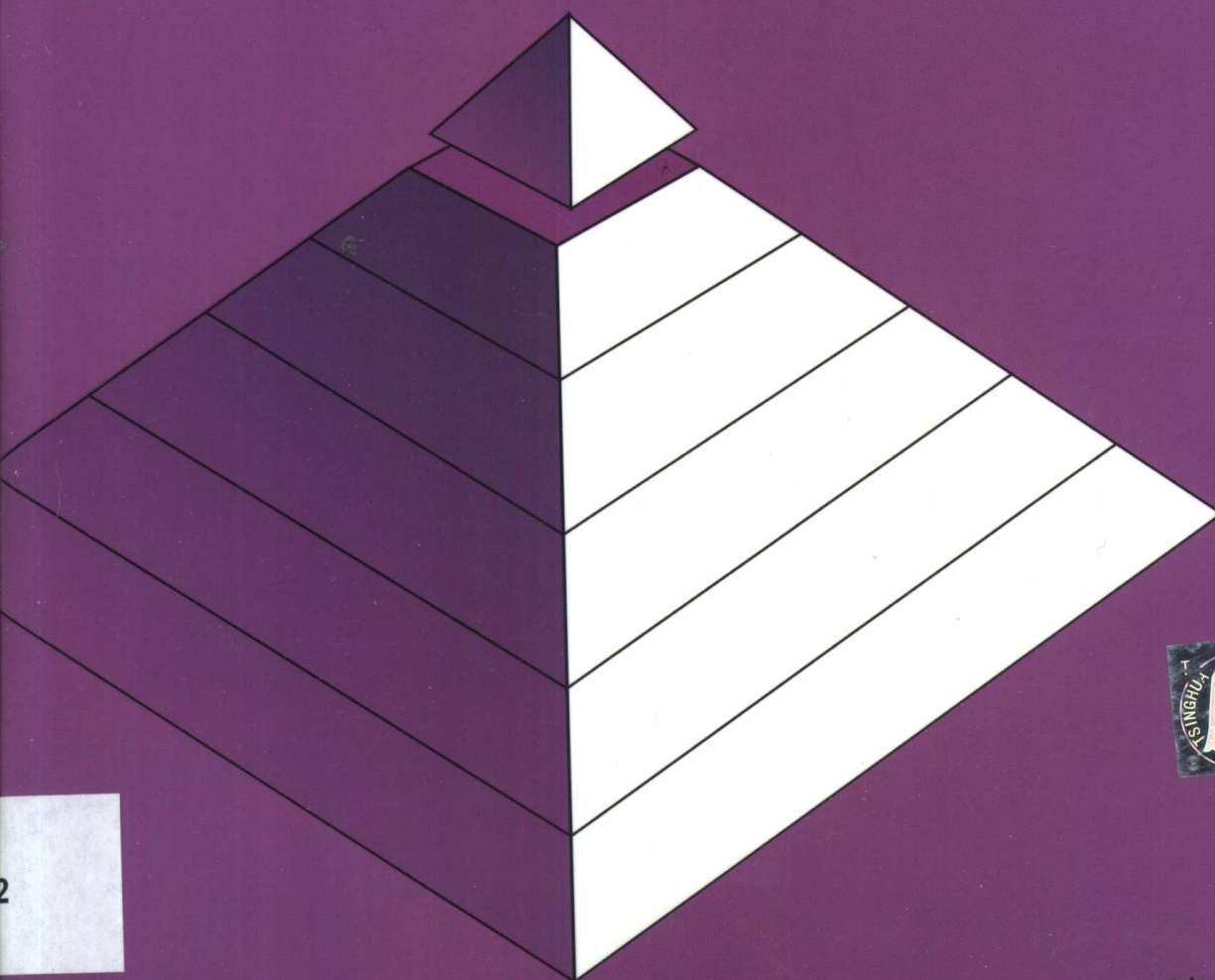
Software Architecture in Practice

林·巴斯 [Len Bass]

[美] 保罗·克莱门茨 [Paul Clements] 著

瑞克·凯兹曼 [Rick Kazman]

孙学涛 杜学绘 刘冬萍 译



清华大学出版社



SEI软件工程译丛

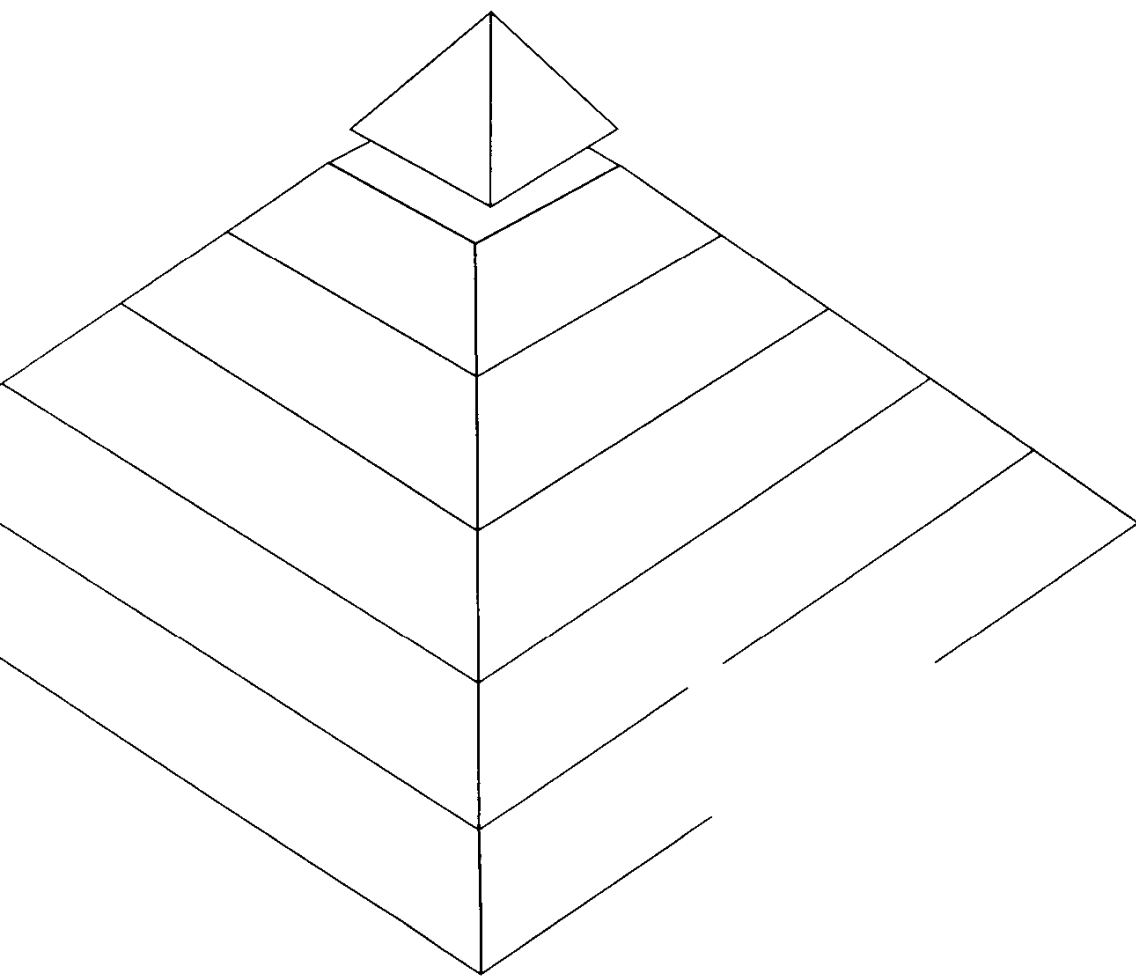
软件构架实践

林·巴斯 [Len Bass]

[美] 保罗·克莱门茨 [Paul Clements] 著

瑞克·凯兹曼 [Rick Kazman]

孙学涛 杜学绘 刘冬萍 译



清华大学出版

(京)新登字 158 号

内 容 简 介

本书根据软件生命周期的特点,分为软件构架基础、软件构架的创建与分析、从构架到实际系统和构架重用 4 部分进行阐述。具体内容包括:软件构架及最终系统的开发、如何通过构架来保证质量属性的实现、通过构架来保证质量属性的案例与分析、可重用资产的构架、基于组件的系统和商业基础结构、商业问题及实时系统的构架等。本书还精选了一些现实系统的案例进行了分析,以帮助读者更好地理解软件构架如何适应商业需求。

本书既可以作为计算机科学或软件工程专业的教材,也适合软件专业人员及其管理者阅读。

Software Architecture in Practice

Len Bass, Paul Clements, Rick Kazman

Copyright © 1998 by Addison Wesley Longman, Inc.

Original English language edition published by Addison Wesley Longman, Inc.

All right reserved.

No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher. For sale in the People's Republic of China Only.

本书中文简体版由 Addison Wesley Longman, Inc. 授权清华大学出版社出版发行, 未经出版者书面许可, 不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号: 图字 01-2002-4425 号

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

软件构架实践/[美]巴斯,[美]克莱门茨,[美]凯兹曼著;孙学涛,杜学绘,刘冬萍译.

—北京:清华大学出版社,2002

(SEI 软件工程译丛)

书名原文: Software Architecture in Practice

ISBN 7-302-05883-0

I. 软... II. ①巴... ②克... ③凯... ④孙... ⑤杜... ⑥刘... III. 软件开发 IV. TP311.52

中国版本图书馆 CIP 数据核字(2002)第 071714 号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 车立红

印 刷 者: 北京牛山世兴印刷厂

发 行 者: 新华书店总店北京发行所

开 本: 787×960 1/16 印张: 29.75 彩插页: 2 字数: 518 千字

版 次: 2002 年 10 月第 1 版 2002 年 10 月第 1 次印刷

书 号: ISBN 7-302-05883-0/TP·3489

印 数: 0001~3500

定 价: 69.00 元

总 序

——为清华大学出版社出版“SEI 软件工程译丛”而作

美国卡内基·梅隆大学软件工程研究所(CMU/SEI)是美国联邦政府资助构建的研究单位,由美国国防部主管。他们确认,为了保证软件开发工作的成功,由软件开发人员、软件采办人员和软件用户组成的集成化团队必须具有必要的软件工程知识和技能,以保证能按时向用户交付正确的软件。所谓“正确的”就是指在功能、性能和成本几个方面都能满足用户要求且无缺陷;所谓“无缺陷”就是指在编码后对软件系统进行了彻底的穷举测试并修复了所有的缺陷,或保证所编写的代码本身不存在缺陷。

CMU/SEI 为了达到这个目的,提出了创造、应用和推广的战略。这里的“创造”是指与软件工程研究社团一起,共同创造新的实践或改进原有的实践,而不墨守成规。这里的“应用”是指与一线开发人员共同工作,以应用、改进和确认这些新的或改进的实践,强调理论联系实际。这里的“推广”是指与整个社团一起,共同鼓励和支持这些经过验证和确认的、新的或改进的实践在世界范围内的应用,通过实践进行进一步的检验和提高。如此循环,往复无穷。

他们把所获得的成就归纳为两个主要领域。一个是倡导软件工程管理的实践,使软件组织在采办、构建和改进软件系统时,具有预测的能力与控制质量、进度、成本、开发周期和生产效率的能力。另一个是改进软件工程技术的实践,使软件工程师具有分析、预测和控制软件系统属性的能力,其中包括在采办、构建和改进软件系统时,能进行恰当的权衡,作出正确的判断和决策。CMU/SEI 通过出版软件工程丛书,总结他们的研究成果和实践经验,是推广这两个领域经验的重大举措。

SEI 软件工程丛书由 CMU/SEI 和 Addison-Wesley 公司共同组织出版,共分 4 个部分:计算机和网络安全(已出版了 2 本著作),工程实践(已出版了 8 本著作),过程改进和过程管理(已出版了 11 本著作),团队软件过程和个体软件

出版说明

1984年,美国国防部出资在卡内基·梅隆大学设立软件工程研究所(Software Engineering Institute,简称SEI)。SEI于1986年开始研究软件过程能力成熟度模型(Capability Maturity Model,简称CMM),1991年正式推出了CMM 1.0版,1993年推出CMM 1.1版。此后,SEI还完成了能力成熟度模型集成(Capability Maturity Model Integration,简称CMMI)。目前,CMM 2.0版已经推出。

CMM自问世以来备受关注,在一些发达国家和地区得到了广泛应用,成为衡量软件公司软件开发管理水平的重要参考因素,并成为软件过程改进的事实标准。CMM目前代表着软件发展的一种思路,一种提高软件过程能力的途径。它为软件行业的发展提供了一个良好的框架,是软件过程能力提高的有用工具。

SEI十几年的研究过程和成果,都浓缩在由SEI资深专家亲自撰写的SEI软件工程丛书(SEI Series In Software Engineering)中。为增强我国软件企业的竞争力,提高国产软件的水平,经清华大学出版社和三联四方工作室共同策划,全面引进了这套丛书,分批影印和翻译出版。这套丛书采取开放式出版,不断改进,不断出版,旨在满足国内软件界人士学习原版软件工程高级教程的愿望。



总 序

——为清华大学出版社出版“SEI 软件工程译丛”而作

美国卡内基·梅隆大学软件工程研究所(CMU/SEI)是美国联邦政府资助构建的研究单位,由美国国防部主管。他们确认,为了保证软件开发工作的成功,由软件开发人员、软件采办人员和软件用户组成的集成化团队必须具有必要的软件工程知识和技能,以保证能按时向用户交付正确的软件。所谓“正确的”就是指在功能、性能和成本几个方面都能满足用户要求且无缺陷;所谓“无缺陷”就是指在编码后对软件系统进行了彻底的穷举测试并修复了所有的缺陷,或保证所编写的代码本身不存在缺陷。

CMU/SEI 为了达到这个目的,提出了创造、应用和推广的战略。这里的“创造”是指与软件工程研究社团一起,共同创造新的实践或改进原有的实践,而不墨守成规。这里的“应用”是指与一线开发人员共同工作,以应用、改进和确认这些新的或改进的实践,强调理论联系实际。这里的“推广”是指与整个社团一起,共同鼓励和支持这些经过验证和确认的、新的或改进的实践在世界范围内的应用,通过实践进行进一步的检验和提高。如此循环,往复无穷。

他们把所获得的成就归纳为两个主要领域。一个是倡导软件工程管理的实践,使软件组织在采办、构建和改进软件系统时,具有预测的能力与控制质量、进度、成本、开发周期和生产效率的能力。另一个是改进软件工程技术的实践,使软件工程师具有分析、预测和控制软件系统属性的能力,其中包括在采办、构建和改进软件系统时,能进行恰当的权衡,作出正确的判断和决策。CMU/SEI 通过出版软件工程丛书,总结他们的研究成果和实践经验,是推广这两个领域经验的重大举措。

SEI 软件工程丛书由 CMU/SEI 和 Addison-Wesley 公司共同组织出版,共分 4 个部分:计算机和网络安全(已出版了 2 本著作),工程实践(已出版了 8 本著作),过程改进和过程管理(已出版了 11 本著作),团队软件过程和个体软件

过程(已出版了3本著作)。前两者属于软件工程技术实践,后两者属于软件工程管理实践。目前这4个部分共出版了24本著作,以向软件工程实践人员和学生方便地提供最新的软件工程信息。这些著作凝聚了全世界软件工程界上百位开拓者和成千上万实践者的创造性劳动,蕴含了大量的宝贵经验和沉痛教训,很值得我们学习。

清华大学出版社邀请我和郑人杰教授共同组织 SEI 软件工程译丛编委会。清华社计划首先影印 6 本著作,翻译出版 15 本著作。据我所知,在 Addison-Wesley 公司出版的 SEI 软件工程丛书中,人民邮电出版社已经翻译出版了《个体软件过程》和《团队软件过程》,还拟影印出版《个体软件过程》和《软件工程规范》;电子工业出版社已经翻译出版了《净室软件工程的技术与过程》、《能力成熟度模型 CMM 1.1 指南》、《能力成熟度模型集成 CMMI》和《软件项目管理》;北京航空航天大学出版社已经翻译出版了《统计过程控制》。这些出版社共计影印 2 本著作,翻译出版 7 本著作。这样,可以预期我国在今年年底共可影印 8 本著作,翻译出版 22 本著作。各个出版社的有远见的辛勤劳动,为我们创造了“引进、消化、吸收、创新”的机遇。我们应该结合各自的实践,认真学习国外的先进经验,以大大提高我国软件工程的理论和实践水平。

在这套丛书中,特别值得一提的是,在过程工程领域被誉为软件过程之父的 Humphrey 先生所撰写的《软件过程管理》、《技术人员管理》、《软件工程规范》、《个体软件过程》、《团队软件过程》和《软件制胜之道》等 6 本著作,将于今年年内全部翻译出版,其中《软件过程管理》、《技术人员管理》、《软件工程规范》、《个体软件过程》和《软件制胜之道》这 5 本著作亦已经或将于今年年内影印出版。

《软件过程管理》是软件过程领域的开创性著作,是为软件公司经理和软件项目经理撰写的。用这本书提出的原理来指导软件开发,可以有效地按照预定进度得到高质量的软件,同时还可了解如何持续进行过程改进。美国 CMU/SEI 按照这本书提出的原理开发了能力成熟度模型,在国际上得到绝大多数国家的认可和广泛采用,是改进软件过程能力的有力武器。在信息技术迅速发展和企业激烈竞争的今天,能否持续改进过程往往决定企业的命运。

作为一个软件经理,在改进组织的能力之前,首先必须明确绝大多数软件问题是由管理不善所引起的。因此,要改进组织的性能,首先需要改进自己的管理模式。同时还要认识到软件开发是一项智力劳动,需要拥有掌握高技能和忘我工作的技术人员。因此,有效的软件管理需要充分注意技术人员的管理。

《技术人员管理》这本著作就是为达到这个目的而撰写的。高质量的技术

工作要求没有差错,这就要求人们高度专心和高度献身。因此要求人们对他所从事的工作不仅具有高度的责任感,而且具有浓厚的兴趣和高度的热忱。在当前知识经济群龙相争的今天,一个能激励人们进行创造性工作的领导群体,是众多竞争因素中最重要的因素。本书提供了大量的实用指南,可用来有效地改进工程人员、经理和组织的性能。

Humphrey 先生还认为这本书特别适合于在我国工作的软件经理。我国是一个人口大国,拥有大量能干的知识分子,而且信息领域的劳动力价格比国际市场的价格要低,因此吸引了许多国家到我国来投资。但若不提高人员的素质,不在产品质量和进度方面也狠下功夫,就不能在这方面持续保持优势。

《软件工程规范》是为编程人员撰写的。它精辟地阐述了个体软件过程(PSP)的基本原理,详尽地描述了人们如何来控制自己的工作,如何与管理方协商各项安排。在软件工程界,这本著作被誉为是软件工程由定性进入定量的标志。目前在世界范围内,有成千上万的软件工程技术人员在接受有关 PSP 的培训,以便正确地遵循 PSP 的实践、开发和管理工作计划,在他们承诺的进度范围内,交付高质量的产品。

《软件制胜之道》这本著作描述了团队软件过程的基本原理,详尽地阐述了在软件组织中如何应用 PSP 和 TSP 的原理以及它所能带来的效益。此外,虽然 CMM 同样适用于小型组织,但在其他著作中都没有描述如何应用 CMM 于个体或小型团队,这本书填补了这个空白。应该指出,如果一个组织正在按照 CMM 改进过程,则 PSP 和 TSP 是和 CMM 完全相容的。如果一个组织还没有按照 CMM 改进过程,则有关 PSP 和 TSP 的训练,可以为未来的 CMM 实践奠定坚实的基础。

在软件工程技术实践方面目前共出版了 10 本著作,其中《用商业组件构建系统》、《软件构架实践》和《软件构架评估——方法和案例研究》等 3 本著作详尽地阐述了软件构架的构建、实践和评估。鉴于是否有一个稳定的软件构架,对软件的质量和成本影响很大,因此如何获得一个良好的构架就成为当今软件界研究的重点。我相信这几本著作的出版,将对我国软件构架领域的研究与实践有重要的参考价值。此外,众所周知,计算机与网络的安全问题对信息系统的可靠使用关系极大,《CERT 安全指南——系统与网络安全实践》的出版将会对我国在这一领域的研究和实践起积极的促进作用。《风险管理——软件系统开发方法》、《软件采办管理——开放系统和 COTS 产品》、《项目管理原理》、《软件产品线——实践和模式》和《系统工程——基于信息的设计方法》等 5 本著作,分别从风险管理、软件采办、项目管理、软件产品线以及信息系统设计方法

等几个方面阐述了大型、复杂软件系统的开发问题,是有关发展软件产业的重要领域,很值得我国软件产业界借鉴。目前我们所处的时代是信息化时代,是人类进入能够综合利用物质、能量和信息三种资源的时代。千百年来以传统的物质产品的生产、流通、消费为基本特征的物质型经济,将逐步进入以信息产品的生产、流通、利用和消费为基本特征的知识型经济。在这个历史任务中,建造和广泛应用各类计算机应用系统是其公共特征。计算机软件是计算机应用系统的灵魂,没有先进的软件产业,不可能有先进的信息产业,从而也不可能建成现代化的知识型经济。

我们应该看到,在软件领域中我国在总体上离世界先进水平还有相当大的差距。但是,我们不能跟随他国的脚印,走他人的老路。我们应该抓住机遇,直接针对未来的目标,在软件工程技术和软件工程管理两个方面,注意研究 SEI 软件工程丛中倡导的原理和方法,联系实际,认真实践,并充分利用我国丰富优秀的人力资源和尊重教育的优良传统,大力培养各个层次的高质量的软件工程师人员,使其具有开发各类大型、复杂软件系统的能力。我衷心地预祝清华大学出版社影印和翻译出版这套丛书,在把我国建设成为一个真正现代化的软件产业大国的历史任务中起到推波助澜的作用,并请读者在阅读这些译著时,对这套丛书的选题、译文和编排等方面都提出批评和建议。

周伯生

于北京

2002 年 8 月 18 日

前言

软件构架是软件业的一个重要研究领域,正引起越来越多的关注,其地位也日益明显地体现出来。但就我们所知,无论是从技术角度还是从管理角度,目前针对实际软件开发商的、有关如何管理软件构架的实用指导文献都还十分缺乏。我们认为,目前很少有关于如何把软件构架和行业或组织的实际情况结合起来的探讨。本书就是基于这样一种认识而编写的。

在设计和评价大型的、复杂的以软件为主的系统方面的经验,使我们认识到软件行业和开发商对这些系统的设计及最终成败所起的作用。构建系统的目的是为了满足不同组织的需求(或者是预先假设的需求),这些需求决定了一个系统必须达到什么样的性能、应具有的可操作性程度、应如何与其他系统实现互操作、应有多长的生命期等。系统的这些属性受软件构架的制约,或者说,上述属性的实现影响着软件设计师的设计选择。

本书通过对一些现实系统的案例分析,阐述了如何把软件构架与行业或组织的实际情况结合起来的问题。这些案例包括:

- 第7章阐述了在保证最低限度的集中控制下,在组织内部,如何快速、方便地把共享文档的设想,最终转化为万维网软件构架。
- 第11章讨论了在空中交通管制中,对安全性的极高要求如何使软件开发商围绕获得极高可用性的构架构建系统。
- 第14章描述了分散各地的不同开发商开发的飞行模拟器各子系统如何连接成一个构架,以便于各子系统的集成。
- 第16章解释了如何为满足产品同时交付的需求而促使(或者说强迫)软件开发商采用某个适当的构架,从而使该公司经济地构建出一组复杂而又相互联系的软件系统。

上述及其他一些案例分析向读者展示了如何根据公司和其业务模型的需求、设计师的经验或主流设计方式来构建合适的软件构架。

另外,我们也讨论了软件构架本身对上述诸因素的影响。一个或一组成功的产品能够影响其他产品的开发,对万维网软件构架的案例分析就是一个很好

的例子。在该系统诞生之前,考虑网络需求的软件很少。当时很少考虑数据的可访问性,只有少数几家组织(一般是金融组织或政府部门)考虑安全问题。

本书的读者对象是软件专业人员——即设计和实现大型的以软件为主的系统人士——和软件专业人员的管理人员。本书不包括诸如对使用某个软件构架的详尽财务分析、如何进行早期的构架分析、如何投资产品线来开发软件之类的问题。尽管我们坚信做这些工作都是值得的,但书中仅提供趣闻一样的案例来支持这样的观点,对此没有进行深入阐述。

作为一种开发产品,软件构架在质量、进度和成本方面具有极高的投资回报。这是因为构架出现于产品生命周期的早期,合理的构架为系统生命周期的其他阶段——开发、集成、测试、更改——奠定良好的基础。相反,如果构架选择不当,就意味着系统的整体结构有误,而且无法通过细小的修补或删除某些部分得到更正,即经常导致要全部拆解重建。另外,与其他开发活动相比,构架分析的投入较低。由于在构架方面所做的决策对以后阶段的开发有着重大影响,而且构架的检查和修正代价相对较低,因此,软件构架能够实现极高的投资回报。

另外,我们认为只有构架良好,可重用的组件才会发挥应有的作用。组件也并非惟一能够重用的部分。构架的重用有利于类似产品的开发,而这又将产生新的组织结构和新的商机。

本书用大量篇幅讲述了一些用于解决组织的现实问题的具体构架。精选这些案例分析的内容旨在说明设计师必须根据具体的质量目标做出各种选择,以及组织的目标对最终系统的影响。

除上述案例分析外,本书还讲述了设计、构建和评价软件构架的若干技巧。我们从理解对构架的实际质量要求和构建满足这些要求的构架的角度来阐述这些技巧,把构架描述语言看作是描述和评价软件构架的手段。我们是从分析和评价某个构架与其系统目标的适应程度来讨论这些技巧的。书中所有技巧都来自我们自己在和 SEI 工作的同事对各种各样的软件系统进行分析的经验。我们所分析的一些系统长达数百万行代码,是由大软件开发商历经数年开发出来的。

我们还提供一种可视化语言,用于描述软件构架。该语言有足够多的表达方式,能够描述过程的和面向对象的系统,并且具有很高的通用性,能够从各种角度(功能的划分、软件结构的集合、硬件结构的集合或这三方的任意组合)描述系统。尽管可视化语言并不是构架的文档,但它是编写这些文档不可缺少的部分。软件构架描述的模糊性是我们对现今该领域研究不满意的原因之一。

希望这里提出的可视化语言能够提高软件构架文档的有效性,从而促进该领域的发展。

本书的读者对象是软件专业人员、具备软件工程知识或经验的学生。预计本书的读者主要有以下三类人员:

(1) 从事软件开发而又希望理解软件构架技术基础和可以对他们的工作产生影响的公司或组织需求作用的软件工程师。

(2) 希望理解软件构架以帮助他们更有效地监管软件系统的开发、提高工作效率的技术经理。

(3) 计算机科学或软件工程专业的学生,他们可将本书作为第一门或第二门软件工程课程的补充读物。

尽管有关商业问题的讨论贯穿全书(如软件构架会如何影响企业的市场竞争力,产品系列的构架将如何影响产品上市时间),我们并没有深入讨论企业经营的问题,也没有使用企业经营方面的专业术语。毕竟,我们是软件工程师,因此,对技术进行了更加深入的讨论。这些技术方面的内容代表了当前软件构架研究的现状——即当前该领域研究与实践相结合的程度,这也是全书的理论基础。书中的案例分析进一步阐明了这些基础,并向读者展示了这些理论知识如何应用于实践。但是,我们在撰写这些案例分析时也做了一定的处理,使得读者无需过多了解相关领域的专业知识就可理解这些案例。例如,书中的空中交通管制系统和飞行模拟案例都不要求读者具备飞行员的专业知识,但要求读者具备相应的计算机科学、软件工程或相关专业的基础知识,以能够从本书的案例分析中真正获益。

对本书的内容安排还要再说明的是,本书并不是为软件构架的设计直接给出切实可行的良方。实际上,我们认为也不可能给出这样一个令人满意的良方。在任何一次设计过程中都必然包含若干权衡:可更改性影响性能、安全性影响可更改性、可扩展性影响可靠性,所有的一切都影响系统的成本。而任何一个“良方”都会显式或隐式地假设某些质量属性相对更为重要,并按这些观点指导用户。这种方法对某个特定的领域可能是合适的,但不可能通用。不同的组织在不同的时间对软件质量都会有不同的要求。

因此,与上述做法相反,本书提供了一种工具箱设计方法。我们认为不同的构架工具和技巧适用于不同的情形、不同的质量目标。任何一个单一的技巧都不可能满足各种需要。所以,我们提供了若干种不同的构架工具(分层、多视图、模式、黑板等)和技巧(分析方法、集成策略、工程原则),并结合不同的行业和技术分别进行了阐述。

考虑到上述因素,在大多数案例分析中都使用多种工具或技巧也就不足为奇了。选择这些案例就是要向读者说明软件构架是决定系统成败与否的重要基础。这些案例之所以是成功的,就是因为使用了恰当的工具并运用了恰当的技巧。缺少其中的任何一种工具或技巧都可能导致系统的失败,这也是我们要通过本书让读者信服的观点。

阅读指南

本书共分 4 个部分。这种划分大致是按照软件生命周期来进行的,我们把这一周期称为体现软件构架如何适应商业需求的构架商业周期。4 个部分的划分如下:

- (1) 软件构架基础:第 1~3 章
- (2) 构架的创建与分析:第 4~11 章
- (3) 从构架到实际系统:第 12~14 章
- (4) 构架重用:第 15~19 章

其中第 3,7,8,11,14,16 和 18 章是案例分析,在章的标题中都已明确标出。下面对各章节所讲述的内容进行概要的介绍。

第 I 部分 软件构架基础

第 1 章-构架商业周期 贯穿全书的主题是“构架不是自然形成的,而是某个周期的一部分”。构架是实现某个目的的手段,它受到来自客户和开发组织的功能及质量目标的影响。构架也受到设计师的素质和经验及技术环境的影响。反过来,构架也影响着所要开发的系统,但它是影响开发组织的一种潜在核心资产。系统对开发组织、构架和潜在的技术环境也都有影响。这影响着系统及其开发组织的未来目标。围绕着构架的这些影响和反馈循环构成构架商业周期(Architecture Business Cycle,简称 ABC)。

第 2 章-什么是软件构架 构架是对系统各结构的一种描述。这里所说的系统各结构有多种类型(数据流结构、模块结构、进程结构等)。构架是能够用以分析确定系统质量属性的实现程度的第一个开发成果,可以用作项目的蓝图。构架也是对组件及连接件之间关系的描述。这就是我们用“构架”这一术语所表达的意义。

第 3 章-A-7E 案例分析:各种构架结构的运用 A-7E 航空电子系统项目的

开发特别重视构架的 3 个不同结构的设计与规范,并以此保证开发工作的简化和系统对未来更改的适应性。本章讲述了这些结构的设计及文档编制工作。

第II部分 构架的创建与分析

第 4 章-质量属性 每个构架都要实现某些特定的软件质量。本章讨论了软件的各种质量属性及其影响。有些质量属性可以通过构架手段来实现,有的在构架上只能部分实现,还有的则与构架无关。本章是从软件构架的角度来阐述质量属性的。

第 5 章-从质量属性到构架:构架样式 在明确了系统所要实现的质量属性之后,还要解决如何设计出某个构架以实现这些质量属性的问题。本章讨论了开发组织的目标、系统的质量属性和构架的结构之间的关系,给出了可用以实现运行时间和开发时间质量属性的许多技巧。

第 6 章-单元操作 单元操作是一组设计操作,构架设计师可以利用这组操作实现从一种构架到另一种构架的变换。本章介绍了多种单元操作,并讲述了在从 20 世纪 80 年代开始的用户界面参考模型历史演变中是如何运用这些单元操作的。我们从单元操作的角度对这一演变过程进行了诠释。

第 7 章-可互操作性案例分析:万维网 万维网起源于某个组织想使其研究人员能够相互交流信息的愿望,现在的万维网早就超出了最初的目标。本章讨论了 Web 基础软件的构架、该构架如何演变以适应 Web 的发展、Web 的发展反过来又怎样给使用它的组织带来极大的影响等内容。

第 8 章-CORBA:对工业标准计算基础结构的案例分析 对象管理小组的创建是为了解决来自各个软件厂商的软件产品的互操作性问题。特别地,通用对象请求代理构架(Common Object Request Broker Architecture,简称 CORBA)就是用以实现这一目标的构架。本章讨论了当初创建对象管理小组的商业环境,并对该小组所创建的各种构架、标准和参考模型(包括 CORBA 和对象管理构架)如何反映这些目标进行了讨论。

第 9 章-在构架层次上分析开发质量:软件构架分析方法 对构架进行评估,确定该构架能否实现所期望的质量属性是可能的。本章给出了对系统的可更改性进行此类评估的一种方法,并给出了几个例子来对该方法的各个步骤进行说明。

第 10 章-构架评审 在构架层次对系统进行评审不仅仅意味着对系统进行

检查。评审时机的选择、合同的签署、输入/输出的确定、正确过程的采用等都是在进行构架评审时需要解决的问题。本章对进行构架评审的“最佳行业实践”做了概括介绍。

第 11 章-空中交通管制系统:高可用性设计案例分析 为空中交通管制设计的系统需要具备几乎始终可用的特性。这一目标导致了若干个设计决策,在本章中一一做了介绍。除了讨论由诸如高可用性和性能这样的质量属性引发的构架决策外,本章还对构架的评估进行了讨论。

第III部分 从构架到实际系统

第 12 章-构架描述语言 构架的描述一般都是非形式化的。在过去的几年中,出现了多种构架描述语言,以期实现对构架的形式化描述。这类语言中,许多是由研究机构开发出来的,有几个已经作为商业产品出现在市场上。将来,这种语言会发挥更重要的作用。本章讨论了如何对构架描述语言的适宜性进行评估,并用一个例子说明了采用这种语言的好处。

第 13 章-基于构架的开发 构架是系统开发的蓝图,系统的构建必须依据这一蓝图而展开。从构架到可执行系统的过程中要创建任务分解结构,并确定它与构架的关系;要以构架为基础创建系统的某个子集版本;要在构架的结构中运用多种模式;要把重用的组件加入到系统中;要保证最终系统与构架相符。

第 14 章-飞行模拟:构架可集成性案例分析 本章描述了一个飞行模拟系统的构架。该案例向我们展示了在复杂领域内充分注意软件构架可以有效地促进一组具有苛刻的功能和逼真性要求的大型系统的构建,并可使这些大型系统的要求为各个软件工程师所充分理解,集成难度降低,并能够经受住下游更改的考验。

第IV部分 构架重用

第 15 章-产品线:在组织内部重用构架资产 构架是各种不同类型的重用的基础。按照某个特定构架的要求开发的组件可以重用,构架本身也可以重用。本章对基于组件的开发和产品线的开发做了讨论。

第 16 章-CelsiusTech 公司:产品线开发案例分析 CelsiusTech 公司在某一

构架上成功地进行了产品线的开发。本章对这一产品线的构架做了描述,并讨论了为什么构架对公司的成功开发意义重大。如果不采用这一方法,CelsiusTech公司就不可能开发出这么多的系统——该公司没有足够的人力。本章也讨论了产品线方法对组织的组织结构的影响、对业务的要求与限制等内容。

第 17 章-构架资产在同行业内的重用 用软件产品构建系统的客户对软件产品符合标准的情况很感兴趣。本章不仅从用户角度对符合某些标准的系统的使用进行了讨论,而且也对标准化过程本身做了阐述。

第 18 章-气象测报站系统案例分析 本章介绍了采用万维网和 CORBA 技术快速经济地开发出来的一个系统。在确定系统所涉及的领域时,务必要让最终用户参与决定在后续阶段应先实施哪个功能,以及该功能应以什么方式呈现给用户。

第 19 章-未来的软件构架 本章再度审视构架商业周期,并指出了关于软件构架需要进一步研究的问题。

图形符号

本书将使用如图 0.1 和图 0.2 所示图形符号来表示软件和系统的构架。我们把这些图形符号看作是一种符号语言,因此它们有与软件设计过程相类似的内在逻辑。在软件设计过程中,我们首先是对功能性有一较为模糊的认识,然后逐步做出一系列的设计决策,使功能性逐步具体化,最终得到实际的系统。书中我们用从圆角框到直角框的变化来反映这一过程。我们用椭圆来表示功能划分时的功能,用圆角矩形来表示软件组件,用切掉四角的矩形来表示过程,用矩形表示硬件组件。

另外,本书的插图中统一用点线来表示数据,用实线表示控制信息。对组件或连接件都使用这样的点线或实线。例如,完成运算功能的组件(如某一过程)是一个软件,所以我们用圆角矩形来表示。但这一圆角矩形有实线和点线两层边,这是因为其中既有数据部分又有控制部分。与之相对的是,被动的数据组件(如文件)是用单层点线边表示的,因为文件里面只有数据。类似地,数据连接(如套接字连接)是用点线表示的,而控制连接(如进程的衍生)则是用实线表示的。