

(德国)Klaus Pohl & Günter Böckle 著
(荷兰)Frank van der Linden
张佳骥 李彦平 译 陈 勇 校

软件产品 线工程

Software Product
Line Engineering



國防工業出版社
National Defense Industry Press

TP344.7

软件产品线工程

Software Product Line Engineering

(德国) Klaus Pohl

(德国) Günter Böckle 著

(荷兰) Frank van der Linden

张佳骥 李彦平 译

陈 勇 校

国防工业出版社

· 北京 ·

著作权合同登记 图字:军—2010—016 号

图书在版编目(CIP)数据

软件产品线工程/(德)普尔(Pohl,K.)等著;张
佳骥,李彦平译. —北京:国防工业出版社,2010.7

书名原文:Software Product Line Engineering

ISBN 978 - 7 - 118 - 06898 - 6

I. ①软... II. ①普...②张...③李... III. ①软件工
程 IV. ①TP311.5

中国版本图书馆 CIP 数据核字(2010)第 122263 号

Translation from the English language edition:

Software Product Line Engineering. Foundations, Principles and Techniques

by Klaus Pohl, Günter Böckle, Frank van der Linden

Copyright © 2005 Springer-Verlag Berlin Heidelberg

Springer is a part of Springer Science + Business Media

All Rights Reserved

※

国防工业出版社出版发行

(北京市海淀区紫竹院南路 23 号 邮政编码 100048)

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787 × 1092 1/16 印张 20 字数 436 千字

2010 年 7 月第 1 版第 1 次印刷 印数 1—3000 册 定价 50.00 元

(本书如有印装错误,我社负责调换)

国防书店:(010)68428422

发行邮购:(010)68414474

发行传真:(010)68411535

发行业务:(010)68472764

译者序

如何提高软件复用率是软件开发所面临的至关重要的问题。实践证明,面向特定领域的软件开发是最有可能被广泛运用的行之有效的提高软件复用率的方法,而软件产品线工程即属其中之一。

本书以特定领域的软件产品或软件产品簇为研究对象,对软件产品线基本理论、主要方法、开发过程、开发组织形态和相关的技术进行了全面的论述。书中以智能住宅系统为领域实例,详细讲述软件产品线技术在特定领域中的应用。书中特别强调了正交变化模型,并由此阐述了软件产品线中的重要概念,即共性、可变性和可追踪性,讨论了如何在软件产品线的各个环节处理可变性问题。

本书讲述的软件产品线工程是当前世界上先进的软件开发理念与方法,学习与理解、运用其原理与方法,对于培养各层次的高质量软件工程人员,对于提高软件复用率,提高软件开发效率与质量将大有裨益。

在本书的翻译与出版过程中,鲍建光对本书译稿作了文字润色。本书的翻译与出版工作得到了中国电子科技集团公司第54研究所有关领导、专家与第54研究所信息处理与控制事业部领导的大力支持,本书的出版受益于总装备部装备科技译著出版基金,并得到国防工业出版社的大力支持,在此一并表示感谢!

本书面向的读者主要是复杂或大型软件的研究、设计人员和管理者,也可作为大专院校研究生、本科生学习的参考材料。

本书涉及到的知识而甚广,尽管译者尽了很大的努力,但难免仍会出现不妥或错误之处,欢迎读者给予批评和建议。信息反馈邮箱:jiajizhang@yahoo.com(张佳骥), 123451yp@163.com(李彦平)。

前 言

1. 软件产品线工程

你对以低成本、快速生产出高质量的软件产品或软件密集型的系统感兴趣吗？如果是，那你手里拿的正是这样的一本书。

软件产品线工程已经被证明是以低成本，短时间，生产高质量、多样性的软件产品或软件密集型系统的一种方法。大量的报告显示了软件业在引入软件产品线后所取得的巨大的成就和宝贵经验。本书第 21 章总结了某些案例。

关于术语，“软件产品家族”和“软件产品线”的意义相同。然而在欧洲，前者使用得更多；在北美，后者用得较多。关于这一点，在两个会议的名称上也得到反映：其一是 2000 年始于美国的软件产品线年会，另一个是 1996 年始于欧洲的软件产品家族系列年会（PFE）；后来，于 2004 年合并为著名的软件产品线（SPLC）年会。

在本书中，我们使用软件产品线这一术语。

2. 本书的读者

本书是为对软件产品线工程原则感兴趣的人准备的，它详细阐述了软件产品线工程的基础，提供了基于经验的关于两个主要过程（领域工程、应用工程）的知识，定义了可变性与可变性的管理。

本书是为产品线实践者、研究者和学生而写的。

3. 主要内容

本书是根据我们的软件产品线工程框架组织的，该框架是基于过去 8 年产品线工程经验开发的。它强调了软件产品线工程与单一系统开发的主要区别。

（1）两个不同开发过程（领域工程和应用工程）的必要性。领域工程过程的目的是定义和实现软件产品线的通用性和可变性；应用工程的目的是通过利用软件产品线的可变性定制特定应用。

（2）清晰定义和管理需要的可变性：在领域工程中，可变性存在于所有的领域工程工件中（需求、架构、构件、测试用例等）；在应用工程阶段根据用户的不同需求利用可变性定制特定应用。

本书还提供了以下问题的答案：

- （1）如何降低开发成本、缩短开发时间，同时提高软件产品的质量？
- （2）如何在软件开发中做到为重用而开发？
- （3）软件产品线的可变性是什么？
- （4）领域和应用工程过程的主要活动和目的是什么？
- （5）如何描述和管理软件产品线的可变性？

(6) 如何保证不同开发工件中,如需求、架构和测试用例中,定义的可变性的一致性?

(7) 如何在应用工程过程中利用可变性,从通用核心平台获得特定产品?

本书分为6个部分:

第一部分:引言

第二部分:产品线可变性

第三部分:领域工程

第四部分:应用工程

第五部分:组织方面

第六部分:经验和下一步的研究

第一部分:引言,讲述软件产品线工程的动机,介绍我们的软件产品线工程框架,对本书通篇使用的范例领域进行了介绍。

第1章:介绍了软件产品线工程的基本原则,它们起源于传统的工程学。

第2章:介绍了软件产品线工程框架,定义了领域工程和应用工程的关键过程和在这些过程中使用和产生的工件。

第3章对智能住宅领域进行了简单的介绍。在全书中,将通过这个例子对主要的原则和概念加以解释。

第二部分:产品线可变性,定义了软件产品线的变化原则,并介绍了如何在各种软件开发工件中表示可变性。

第4章定义了软件产品线工程可变性的原则并介绍了正交变化模型。通过使用这一模型,将对本书中各种软件开发工件的可变性加以清晰与无歧义性的表述。

第5章定义了如何在需求工件,如文档化需求、特征、场景、用例、状态图和类图中,描述可变性。

第6章定义了如何在架构工件中描述可变性,如在软件架构的开发视图、进程视图、代码视图中描述可变性。

第7章定义了如何描述构件接口的可变性和构件内部结构的可变性。

第8章定义了如何在测试工件中定义可变性,如在测试用例、测试用例场景和测试用例场景步骤中描述可变性。

第三部分:领域工程,定义了领域工程过程中的关键子过程。对每个子过程描述了如何构建通用(不变)的产品线工件,定义了软件产品线的可变性。

第9章介绍了领域工程过程中产品管理子过程的原则,这个子过程主要与经济方面的问题相关,尤其是与产品目录管理相关。

第10章定义了需求工程子过程的原则。它定义并描述了在软件产品线中如何识别和描述通用和可变特征以及需求。

第11章定义了软件产品线的参考架构,描述了如何将产品线的通用部分和可变部分包含在参考架构中。

第12章解决的是进行可重用软件构件的详细设计问题,定义了如何将参考架构中定义的通用部分和可变部分映射到构件。

第 13 章讨论了可变性对不同的测试级别的影响,提出并分析了测试的策略以及如何
在软件产品线中应用测试策略。重点是在软件产品线测试中建立对测试工件的全面
重用。

第 14 章提出了在考虑软件产品线可变性的情况下,选择商用(COTS)构件的技术。
在这里,主要考虑能提供部分重要功能的构件,即所谓的高层构件。

第四部分:应用工程,定义了应用工程过程的关键子过程。描述了在应用工程阶段,
如何使用领域工程阶段建立的正交变化模型绑定可变性,从而实现有计划的重用。

第 15 章定义了应用需求工程子过程,解决在定义一个应用时如何利用软件产品线中
的通用构件工件和可变构件工件的问题。展示了正交变化模型如何在应用需求工程中支
持软件产品线工件的重用。

第 16 章讨论应用设计子过程,它从参考架构中获得应用架构。根据应用需求绑定可
变性,选择需要的变量并加入到应用架构中。该子过程也将对应用的特殊需求进行设计。

第 17 章是关于软件产品线上特定应用的实现。理想情况下,应用是通过将可重用领
域构件与特定应用构件进行配置而实现的。在此过程中,利用了构件及其接口的共性点
与可变性。

第 18 章是关于应用的测试。描述了领域测试工件中引入的可变性如何在应用工
程中支持测试用例设计重用,从而大大减少为产品线不同应用开发测试用例的工作
量。

第五部分:组织方面,介绍了将软件产品线引入到组织时需要解决的两个关键问
题——组织结构和转化过程。

第 19 章讨论了软件产品线工程中不同的组织结构的优缺点。

第 20 章描述了如何实现从单一系统开发向软件产品线开发的转化。指出了在何时
采用何种转化策略取决于组织目前的状况。

第六部分:经验和下一步的研究,介绍了在实施软件产品线方面的经验,简要描述了
下一步研究的几个主要内容。

第 21 章总结了将软件产品线工程方法引入到几个组织中所取得的经验的报道。同
时列出了若干参考文献,用于进一步的阅读。

第 22 章介绍了在下一步软件产品线工程领域研究中面临的主要挑战。

此外,作为本书的结尾,我们还提供了作者的有关信息、全书中涉及的参考文献、软
件产品线工程术语与书中的索引。

4. 分享你的经验!

如果你有改进建议,或者发现了书中的错误,或者发现遗漏了重要议题,请立即与作
者联系,我们将对你的反馈十分感谢。

联系地址:

SPLE-Book@ software-productline. com

或者访问本书的英文网页:

www. software-productline. com/SPLE-Book

5. 致谢

感谢 Eureka/ITEA, BMBF (Germany), NOVEM/SENER (Netherlands), 以及资助 ES-APS (1999—2001), CAFÉ (2001—2003) 与 FAMILIES (2003—2005) 项目的领导者, 本书中大多数研究成果是在这些项目中得出并验证的。特别要感谢 Ernst Sikora, 感谢他为改善本书的一致性与版面所作出的努力。也感谢 Silja Recknagel 对本书英文版的付印所作的支持。感谢 Springer Verlag, Heidelberg 的 Ralf Gerstner, Ronan Nugent 与 Ulrike Stricker 为本书出版所作的努力。

最后, 感谢国内及国际上工业界及学术界的合作, 没有他们对于某些问题的深入探讨, 本书的写作是不可能完成的。

Klaus Pohl

University of Duisburg—Essen, Germany

Günter Böckle

Siemens Corporate Technology, Germany

Frank van der Linden

Philips Medical Systems, The Netherlands

May, 2005

教学指导

许多公司采用软件产品线工程(SPLE)开发嵌入式软件和信息系统,了解 SPLE 对于员工的发展意义重大。本书提供了 14 周的基本的 SPLE 阅读资料和作为高级软件工程课程的部分材料。SPLE 课程可以按照我们提出的 SPLE 框架为基础来组织,SPLE 框架强调了 SPLE 和单一系统开发的主要差别,即区分两个开发过程(领域工程和应用工程)的必要性和显式定义与管理可变性的必要性。

假设一节课需要两个小时,本书覆盖的内容可以划分为下面的课程和模块:

(1) 十四节关于 SPLE 的课程,几乎覆盖本书所有的内容(第一部分)。

(2) 两节关于 SPLE 的高级软件工程课程模块,介绍 SPLE 的主要概念(第二部分)。

(3) 四节关于 SPLE 的高级软件工程课程模块,介绍 SPLE 的主要概念,并说明其在领域工程和应用工程某一特定过程阶段的应用。例如,用于领域和应用需求子过程,或者领域和应用设计子工程(第三部分)。

(4) 一节课程是关于 COTS 系统的选择,是软件工程过程的高级课程,该课程介绍了选择 COTS(商用)构件或系统时所面临的挑战和技术(第四部分)。

(5) 三节关于 SPLE 中测试的课程模块,它们是软件质量保证课程的一个组成部分,或者是高级软件工程课程。该模块介绍了 SPLE 的主要概念,讲授领域测试工件和应用测试工件的基本解决方案和面临的挑战(第五部分)。

另外,建议另外一个三节的课程模块:

(1) 软件产品线需求工程,它是高级需求工程课程的一部分;

(2) 软件产品线设计,它是高级软件架构或设计课程的一部分。

1. 十四节关于 SPLE 的课程

本部分课程提供 SPLE 的总体介绍。该课程讲述关于 SPLE 的各个重要方面,按照本书提出的产品线工程框架组织有关材料。参加本课程的学生应该有软件工程的基础知识。学完本课程,将会熟知软件产品线工程(SPLE)的有关原则:可变性管理、领域及应用工程。

建议十四节课程的结构如下:

两节介绍 SPLE 框架原则的课程:

(1) SPLE 原则(第 1 章),SPLE 经验(第 21 章);

(2) 单一系统工程和 SPLE 在开发上的区别;SPLE 框架(第 2 章)。

四节关于可变性原则和如何描述可变性的课程,重点要强调在不同的开发工件中可变性的一致性定义:

(3) 通过例子介绍产品线可变性原则与基本概念(第 4 章);

- (4) 通过例子介绍正交变化元模型的概念(第4章),描述需求工件(第5章);
 - (5) 利用一个完整的例子说明需求工件中的可变性(第5章),以及其与设计的内在关系(第5章和第6章);
 - (6) 基于正交变化模型在设计、实现和测试工件中描述可变性(第6章~第8章)。
- 五节关于领域工程子过程的课程,着重强调 SPLE 和单一系统开发的区别:
- (7) 介绍产品管理、产品目录定义、产品范围定义(第9章);
 - (8) 介绍领域需求工程子过程,定义可变性、通用性和可变性分析、需求可变性建模(第10章);
 - (9) 领域设计子过程,将需求的可变性精化为设计的可变性,定义参考架构/平台(第11章);
 - (10) 领域实现,将设计可变性映射到构件(第12章);在高层设计中选择 COTS, COTS 选择技术(第14章);
 - (11) 领域测试,领域测试的策略,根据需求、设计和实现的可变性定义测试工件中的可变性(第13章)。
- 两节应用工程的课程,介绍如何从领域工件通过绑定预定义的可变性得到应用工件:
- (12) 应用需求工程和应用设计子过程,定义产品线应用的需求,绑定领域需求和设计工件中定义的可变性,得到应用设计(第15章和第16章);
 - (13) 应用实现和应用测试子过程,根据应用需求工程和应用设计中建立的可变性绑定,从领域测试工件中获得应用测试工件(第17章和第18章)。
- 一节关于组织方面的课程,重点介绍当组织内引入 SPLE 时,在组织结构上产生的变化:

(14) 组织结构对 SPLE 的影响,评估和确定 SPLE 的 ROI(投入产出比),在组织中引入 SPLE 时的基本转变策略(第19章与第20章)。

课程大纲的另外一种选择是根据领域和应用工程子过程的另一种顺序来组织的,在每个领域工程子过程之后是对应的应用工程子过程。例如,领域需求工程、应用需求工程、领域设计、应用设计等。按照这种方式安排课程的优势是强调领域工程和应用工程的区别和其联系。不足之处是无法着重强调领域工程子过程之间的关系、应用工程子过程之间的关系。

2. 两节关于 SPLE 的课程

两节适用于高级软件工程课程模块。在这两节课中,学生将学习在组织内引入 SPLE 的关键动机,以及与单一系统开发的主要区别。建议两节课的内容如下:

(1) SPLE 的动机和原理(第1章),再包括一两个实践经验的例子(第21章);SPLE 和单一系统开发的主要区别(第2章)。

(2) SPLE 框架(第2章);软件产品线可变性原理;可变性建模概览(第4章)。

3. 四节关于 SPLE 的课程

四节 SPLE 课程模块,全面介绍 SPLE 的基本原理,以及与单一系统开发的区别。如果时间允许,我们建议如下四节关于 SPLE 介绍的课程:

(1) SPLE 的动机和原理(第 1 章),再包括一两个实际经验的例子(第 21 章);SPLE 与单一系统开发的主要区别(第 2 章)。

(2) SPLE 框架(第 2 章);软件产品线可变性的原理;可变性建模概览(第 4 章)。

(3) 正交变化元模型(第 4 章);在需求或设计工件中描述可变性的例子(依赖于下一节要讲的内容)。

(4) 领域和应用工程过程的具体内容,建议选择领域和应用工程的特定阶段阐述两个过程的具体内容,同时建议选取领域和应用需求工程子过程(第 10 章和第 5 章),或者选取领域和应用设计子过程(第 11 章和第 16 章)。

4. 一节关于 COTS 选择的课程模块

本课程介绍在高层设计阶段选择产品线 COTS 构件面临的挑战。此外,关于如何选择 COTS 构件的技术可以作为该课程的一部分。这一技术综合考虑了领域需求、领域架构与产品线的可变性(第 14 章)。

5. 三节关于 SPLE 测试的课程模块

这三节课的课程提供对 SPLE 的简单介绍,重点是领域工程和应用工程子过程中测试方面的具体内容。

(1) SPLE 的动机和原则(第 1 章);SPLE 和单一系统开发的主要区别(第 2 章);测试面临的挑战(第 13 章)。

(2) 软件产品线可变性的原则;正交变化元模型(第 4 章);在测试用例工件中描述可变性(第 8 章)。

(3) 测试策略与评估准则,在测试用例设计中保留可变性(第 13 章);从领域测试用例得到应用测试用例,测试工件重用(第 18 章)。

6. 练习

我们建议两种类型的练习。一种是随堂练习,目的是使学生加深对 SPLE 原则、可变性建模和领域与应用过程知识的了解;第二种是课后练习,除了第一种形式练习外,增加扩展软件产品线能力的作业。建议可以这样做:增加功能和非功能特征,并获得一个具体的产品线应用。

1) 书面练习

为理解 SPLE 的原理,得到正交变化模型的经验并在领域和应用工程中使用的各种工件模型,建议进行下面的练习。

(1) 产品线原则:学生应该熟悉两三个产品线工程实践经验的报告(基于第 21 章提到的参考文献)。应该总结和比较报告中的经验,将正面或反面的经验用产品线的原则进行归纳。结果是 20 分钟的演讲与/或三四页的书面报告。

(2) 定义可变性:基于必须集成到现有软件产品线的可变性的自然语言描述,学生应该扩展正交变化模型(不需考虑开发工件)。

(3) 在需求工件中建模可变性:给学生一份已有的需求文档和需要集成到变化模型和需求工件中的一个新可变性。他们必须在集成中考虑可变性依赖和约束。例子的规模应该较小。

(4) 将需求工件中的可变性映射到设计工件中:给定学生一个设计文档,需求文档和产品线的正交变化模型。在正交变化模型和需求模型中,标记了新的扩展点。学生应该能够将这些扩展点映射到产品线的软件架构,并同时考虑可变性依赖和约束及设计约束。例子的规模要小,为避免不需要的关注点,要用与练习(3)不同的例子。

可选部分:此外,学生可以承担将内部设计可变性集成到设计中的任务。例如,应该使架构是灵活的,以便当使用不同的中间件时,可以在不同的中间件供应商中作出选择。

(5) 将设计工件中的可变性映射到实现工件:给定学生一个实现文档、设计文档、需求文档和产品线正交变化模型。在正交变化模型和设计模型中,标示新的扩展点。学生应该能够将这些扩展点映射到软件构件和其接口,并考虑可变性依赖和约束以及实现约束。为了避免不必要的关注点,例子应该小,要不同于练习(3)和练习(4)的例子。

(6) 为系统测试定义领域测试用例:基于需求文档、设计文档和正交变化模型,学生应该为部分需求规格说明设计一组测试用例。领域工件的可变性应该保留下来,即应该完全引入到测试用例的设计中去。

(7) 将应用需求映射到领域需求工件:基于一系列应用需求工件(如特征和场景),学生必须识别出领域需求工件中对应的共性和可变性。此外,必须对领域需求工件中定义的可变性给出合适的绑定,以实现给定的应用需求。

(8) 获得应用设计:根据应用需求中的绑定(正交变化模型和应用需求工件中描述),学生应该得出应用设计,它根据应用需求绑定领域设计的可变性。

(9) 获得应用的系统测试用例设计:根据正交变化模型中绑定的可变性,学生应该从领域测试用例得到应用的系统测试用例,以此测试给定应用的功能和质量特征。

(10) 期望的工具支持:根据在练习中获得的经验,学生应该定义一个所需的实施软件产品线的支持工具。每个学生应该着眼于对一个具体的领域和应用工程子过程的支持,如领域和应用需求工程工具;或者跨越子过程的工具,如可变性建模工具或配置支持工具。

(11) 扩展正交变化元模型:学生根据经验,建议对正交变化模型的以下方面扩展:①变化点和变量的特化;②与领域工件的关系以及领域工件之间的关系;③与应用工件的关系和应用工件之间的关系。建议对做出的扩展模型要进行讨论。

此外,课程可以加入适当的基于问题的练习。

2) 扩展产品线和获得应用

练习的前提是已有软件产品线的(部分)实现,包括最新的软件产品线特征、需求、设计、构件和测试用例的描述。示例产品线可以以本书的住宅自动化例子(第3章)为基础。

本练习有双重目的。第一阶段通过练习课程类型V中定义的练习,加深关于SPLE的认识;第二阶段进行领域工程和应用工程实践,在现有的产品线中增加新的特征并获得产品线应用,这包括对第一阶段建立的模型的调整和扩展。还应该产生在两个过程中每一步骤间的反馈,以及应用工程向领域工程的反馈。如果时间允许,还要基于反馈内容主动地重做上述过程。与书面练习相反,这是一个更具实践性的练习,即对产品线的扩展是

基于支持工具和一个(部分)实现了的产品线进行的。

在第一阶段,建议采用书面练习对学生进行培训,如利用练习(2)、(3)、(4)、(7)和(8)。

第二阶段的练习包括两个部分。

(1) 新特征的集成:学生应该增加一个或多个功能或/和非功能特征或需求,所有这些刻意选取的特征或需求对现有产品线的可变性会产生影响。如果时间允许,建议:

① 选择一个新的导致产品线可变性扩展的特征,让学生感觉到可变性很容易集成。

② 选择一个特征,它需要引入一个新的变化点,结果将比产品线原则中的任务要复杂的多。

两种情况下的主要挑战是新特征在所有产品线工件中的一致集成,更重要的是需要考虑集成对领域资产以及现有应用的影响。学生将体会到如果扩展影响到现有可变性时,追踪关系和正交变化模型起到重要的作用。

(2) 获得产品线应用:给定新应用的一组需求工件(特征、需求、用例、用例场景),由此获得新应用面临双重挑战。第一,学生应该将应用需求向产品线需求映射,确定实现应用需求需要绑定哪些可变性;第二,学生获得如何利用工具从需求中识别出来的可变性绑定获取应用的实践经验,即利用正交变化模型支持将应用需求中绑定的可变性映射到应用设计、实现和测试用例设计的经验。这个练习可以通过以下方式进行扩展,如为学生设定一个应用需求,它包括对应用需求工件(不应该与领域工件的扩展相混淆)面向具体应用的扩展。

目 录

第一部分 引 言

第 1 章 软件产品线工程介绍	2
1.1 产品线工程的原则	2
1.1.1 大规模定制	2
1.1.2 平台	3
1.1.3 将基于平台的开发和大规模定制相结合	4
1.2 定制产品的工程化	4
1.2.1 创建平台	4
1.2.2 引入灵活性	5
1.2.3 公司的重新组织	5
1.3 产品线工程的动机	6
1.3.1 降低开发成本	6
1.3.2 提高质量	6
1.3.3 缩短上市时间	7
1.3.4 其他动机	7
1.4 软件产品线工程	8
1.4.1 定义	9
1.4.2 软件平台	9
1.4.3 前提条件	10
第 2 章 软件产品线工程框架	12
2.1 引言	12
2.2 两个开发过程	12
2.3 过程框架概述	13
2.4 领域工程	14
2.4.1 产品管理	15
2.4.2 领域需求工程	16
2.4.3 领域设计	16
2.4.4 领域实现	16

2.4.5	领域测试	17
2.4.6	其他软件质量保证技术	17
2.5	领域工件	17
2.5.1	产品路线图	17
2.5.2	领域变化模型	18
2.5.3	领域需求	18
2.5.4	领域架构	18
2.5.5	领域实现工件	18
2.5.6	领域测试工件	19
2.6	应用工程	19
2.6.1	应用需求工程	20
2.6.2	应用设计	20
2.6.3	应用实现	20
2.6.4	应用测试	21
2.7	应用工件	21
2.7.1	应用变化模型	21
2.7.2	应用需求	22
2.7.3	应用架构	22
2.7.4	应用实现工件	22
2.7.5	应用测试工件	22
2.8	在本书中框架的角色	22
第3章	住宅自动化领域的例子	25
3.1	智能住宅基础设施	25
3.1.1	目标	25
3.1.2	利益相关者	26
3.1.3	智能住宅和传统住宅的区别	26
3.2	住宅自动化系统的构建模块	27
3.2.1	传感器和激励源	27
3.2.2	智能控制设备	27
3.2.3	住宅网关	28
3.2.4	网络	29
3.2.5	住宅自动化领域的标准	29
3.3	例子	29
3.3.1	系统功能	29
3.3.2	一个简单的系统配置	31
3.3.3	系统构件交互	31
3.4	智能住宅应用的软件可变性	32
3.4.1	可变性的例子	32

3.4.2 可变性的原因	33
3.5 本书中住宅自动化领域的角色	33

第二部分 产品线可变性

第4章 可变性原则	36
4.1 引言	36
4.2 变化主题和变化对象	37
4.3 软件产品线工程中的可变性	38
4.3.1 变化点	38
4.3.2 变量	39
4.3.3 定义变化点和变量	39
4.3.4 软件产品线的可变性	40
4.4 时间可变性和空间可变性的对比	41
4.5 内部可变性和外部可变性	42
4.5.1 存在外部可变性的原因	43
4.5.2 存在内部可变性的原因	44
4.5.3 内部可变性和外部可变性的判定	44
4.5.4 可变性金字塔	44
4.6 正交变化模型	45
4.6.1 可变性的清晰描述	45
4.6.2 正交可变性定义	46
4.6.3 变化点、变量和可变性依赖	47
4.6.4 可替代选择	48
4.6.5 可变性约束	49
4.6.6 变化模型和其他开发工件之间的追踪关系	52
4.6.7 图形标记	53
4.6.8 例子	53
4.6.9 术语的使用	54
4.7 处理变化模型中的复杂性	55
4.8 与单一系统工程的差别	56
4.9 总结	56
第5章 需求工件的可变性描述	57
5.1 引言	57
5.2 描述需求	58
5.2.1 基于模型的和文本格式的需求描述	58

5.2.2	需求工件	58
5.2.3	目标和特征	59
5.2.4	用例和场景	59
5.2.5	传统的需求模型	60
5.3	文本格式的需求中的可变性	61
5.3.1	在文本格式的需求中定义可变性	61
5.3.2	用 XML 描述可变性	62
5.4	需求模型中的可变性	63
5.4.1	特征模型中的可变性	63
5.4.2	用例模型中的可变性	66
5.4.3	传统需求模型中的可变性	68
5.5	变化模型和需求工件之间的追踪	71
5.6	与单一系统工程的区别	72
5.7	总结	73
第 6 章	设计工件的可变性描述	75
6.1	引言	75
6.2	架构工件	76
6.3	参考架构	80
6.4	开发视图中的可变性	80
6.4.1	子系统和层	80
6.4.2	构件	82
6.4.3	接口的作用	83
6.4.4	配置	83
6.5	处理视图中的可变性	84
6.6	代码视图中的可变性	86
6.7	与单一系统工程的差别	87
6.8	总结	87
第 7 章	实现工件可变性描述	88
7.1	引言	88
7.2	详细设计工件	89
7.3	构件接口可变性	90
7.3.1	算法和协议的可变性	91
7.3.2	资源的可变性	92
7.3.3	应用配置的可变性	92
7.3.4	多个构件提供接口	93
7.4	内在的构件可变性	94
7.5	与单一系统工程的区别	96