

软件复用与构件技术系列

软件复用

结构、过程和组织

Software Reuse

Architecture, Process and
Organization for Business Success

Ivar Jacobson
Martin Griss 著
Patrik Jonsson

韩柯 译



机械工业出版社
China Machine Press

52
软件工程技术丛书

68
软件复用与构件技术系列

软件复用

结构、过程和组织

Software Reuse
Architecture, Process and
Organization for Business Success

Ivar Jacobson
Martin Griss 著
Patrik Jonsson

韩柯 译



机械工业出版社
China Machine Press

本书是一本关于面向对象软件工程实质性复用的著作,主要讲述了复用驱动的软件工程的业务框架如何系统地解决关键业务、过程、体系结构和组织的问题,并通过讲解系统化软件复用的理论和实践,以及系统化的基于模型面向对象的软件开发和业务工程理论实践,帮助读者开发出软件复用的新方法。本书把作者致力多年的系统化方法、体系结构、与领域有关的软件工程以及复用适配等领域的最佳思想融入其中,向读者充分展示了面向对象基于组件的软件工程的未来。

Ivar Jacobson, Martin Griss, Patrik Jonsson: Software Reuse: Architecture, Process and Organization for Business Success (ISBN 0-201-92476-5).

Original edition copyright © 1997 by Pearson Education Limited. All rights reserved.

Chinese edition is published by arrangement with Pearson Education Limited.

All rights reserved.

本书中文简体字版由美国 Pearson Education 培生教育出版集团授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号: 图字: 01-2002-1565

图书在版编目 (CIP) 数据

软件复用: 结构、过程和组织/雅各布森 (Jacobson, I.) 等著; 韩柯译. - 北京: 机械工业出版社, 2003.1

(软件工程技术丛书 软件复用与构件技术系列)

书名原文: Software Reuse: Architecture, Process and Organization for Business Success

ISBN 7-111-11033-1

I. 软… II. ①雅…②韩… III. 软件工程 IV. ①31.55

中国版本图书馆 CIP 数据核字 (2002) 第 7701 号

机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑: 何健辉 武恩玉

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2003 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·25 印张

印数: 0 001-5 000 册

定价: 55.00 元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

序 言

Björn Svedberg

在标志着当今业界特征的快节奏全球业务活动的有效运作中，信息技术起着前所未有的重要作用。在开发这些信息技术的过程中，软件设计对于关键任务业务系统的创建至关重要。这些系统使得信息可以快速流动，并支持整个企业的决策制订。

企业的成功常常取决于能否很好地管理其软件资产。高效的软件确实是一种竞争手段，并且已经有事实证明，软件的有效性是很多公司成功地跨到业界前列，而另一些企业却销声匿迹的原因。作为爱立信公司的主席和 Skandinaviska Enskilda Banken 银行的总裁，我已经看到有效软件战略对电信公司和银行产生的重要影响。

今天，银行和电信公司都在为快速开发和部署新业务而激烈竞争。没有设计良好、管理良好的大量软件组件，是不可能实现这一目的的。

20 世纪 70 年代中期，我们爱立信公司开发了新一代交换系统 AXE。那时我负责开发工作，充分看到了软件的重要性。

在激烈的全球化竞争中，爱立信必须为所有电信运营公司提供量身定制的产品。这意味着我们向开发部门提出的首要要求是开发具有高度可适应性的产品。我们设计了可以在全球复用的核心系统，并努力工作，以尽可能减少打开每个新市场时修改和扩展 AXE 所需要的工作。

多亏采用了由 Ivar Jacobson 领导的核心团队开发的基于组件的设计方法，才使我们能够以比竞争对手更超前的客户视角来开发产品。AXE 系统具有独特的软件体系结构，它是由不同层次上的组件构成的，通过定义完备的接口连接起来。这种结构能够使我们在系统的功能需求增强时引入新的组件。通过从组件库中选择一组特殊的组件，并添加一些新的与客户有关的组件，就可以配置特定的客户系统。

毫无疑问，这种组件复用方法对于爱立信的成功是至关重要的。

像 Skandinaviska Enskilda Banken 这样的现代银行也非常依赖软件。信息的快速流动绝对是至关重要的。随着业务节奏的不断加快和世界的快速变化，客户不断需要新的功能。新的账户类型、新的业务、新的贸易策略以及在线银行业务都是一些简单的例子。这些新功能几乎全是软件领域的问题。为了提供新的价值并进行有效竞争，银行必须在很短的时间内设计、开发并部署这些新的业务，完成这些工作往往只有几个月的时间。软件开发必须快，但是不能以牺牲质量和大幅度提高成本为代价。

同时，我们必须对新技术作出反应，平稳地替代老软件中的巨大投资，从大型主机计算环境转向基于因特网的分布式计算环境，并通过因特网与客户保持联系。为了迎接这些挑战并满足客户的需求，必须采用一种有效、灵活的系统体系结构和组件软件策略。

基于本书的背景以及 Ivar 在爱立信和其他公司软件开发上发挥的作用，我深信，本书所介绍的经验教训对于所有依赖软件的公司都具有十分重要的价值。

在编写本书的过程中，Ivar 与惠普公司的顶级复用专家 Martin Griss 进行了合作。惠普公司是平台、度量系统和制造方面的世界领导者。爱立信公司与惠普公司基于组件设计与复用经验的联合与补充，使我真正相信本书是对软件业的经典贡献。

从斯德哥尔摩的皇家技术学院电子工程系毕业后，Bjorn Svedberg 博士在爱立信公司开始了自己的职业生涯。他在公司的职务迅速得到提升，责任越来越重，所有职务都有很强的技术要求：存储器设计、计算机体系结构、软件设计与实现、软件体系结构、开发过程等。

Svedberg 博士 40 岁时成为爱立信公司这个电信业世界领导者的总裁，他在这个位置上工作了 10 年以上。1990 年，Svedberg 博士被任命为爱立信公司董事长。几乎同时，他又接受了瑞典最大银行之一 Skandinaviska Enskilda Banken 的总裁职务。

Svedberg 博士 1984 年被授予 Lund 技术学院荣誉博士学位。

序 言

Joel Birnbaum

应邀为本书作序，我感到非常荣幸。每一位负责软件使用或开发的高级经理都应该阅读此书。对于软件工程以及理论与实践的发展，本书无疑是一部经典之作，这是因为，它会在很多层面上使读者受益：对所面临的各种挑战及今天所拥有的最佳解决方案性质的总的理解，有关准备可复用代码技术的详细说明，以及用于定制处理过程的体系结构框架，这个过程对于当前这个依赖信息技术的年代是至关重要的竞争要素。

参加工作后三十多年来，我首先担任的是研究员，后来是工业及大学研究的经理和赞助商，这些研究主要解决软件设计、开发和修改方面的问题，以将这些工作提高到与信息工业在硬件系统上取得巨大进步的工程原理相当的高度上。我先在 IBM 研究所工作了 15 年，然后又在惠普公司工作了 16 年，在这期间，我一直在考虑目前还不能将软件开发工作限制在可以预测、可以重复的过程中的问题，这是将计算转换为一种真正普遍深入技术的惟一最大障碍。例如对于惠普公司，很多人都认为它主要是计算硬件平台、度量仪器和计算机外设制造商，但几年前进行的一次非正式调查显示，我们超过 70% 的研发开支是花在软件和固件上。据保守估计，在 1997 年 30 亿美元的研发开销中，软件和固件在总开支中所占的比例几乎上升到了 80%，这是因为我们的收入越来越多地依赖于提供完整解决方案、系统集成和客户服务。

作为惠普公司的一名高级技术人员，我每天都要关注能够为公司的成功带来实质性长期变化的思想、方法、过程和工具，Martin Griss 及其同事在软件复用方面的工作是促使惠普公司出现这种变化的开始。作为公司研发部门的负责人，工作性质使我要接触来自不同行业的很多客户，以尽量理解他们的需求和所面临的主要困难。我还从来没有遇到过不把软件列在需求清单的首位或接近首位的客户。

Martin Griss 1993 年加入惠普公司，从一开始我就认为他具有适合领导这个领域的研究工作的才能、经验和兴趣。尤其是他在物理学方面接受过严格培训，能够很好地把握如何利用自己的科学研究经验。很多经过理论培训的计算机科学家和软件经理缺少设计、执行以及统计分析研究结果的经验，而这种统计分析要求必须系统地隔离可变因素并查找不明显的关联。作为犹他大学的计算机科学教授，Martin 已经获得了很大成功，除了完成很多其他工作之外，他还以开创性语言设计者著名，他领导的一个研究小组开发了“可移植标准 Lisp”语言，这在当时是一种创新性语言。在惠普公司实验室工作时，他在自己已经很广的知识基础上，又增强了组织和管理技能。我一直在找一位具有创造性，能够把广博的知识、远见和坚韧与宽广的专业知识结合在一起的领导，如程序语言设计、软件工程开发工具、软件过程设计与建模以及机构设计与改革等。Martin 具备这些素质，他接受了挑战，准备在未来 10 年中专心致力于追求这一目标。我们两个人都认为这不是一件容易的事，也不那么吸引人，不会有单一的工具，也不会有杀手锏。我们还知道，具有硬件工作背景经理，既不会理解也不会容忍有条不紊但却很缓

慢的进步。我们义无反顾地承担了这一任务。今天，我们公司的所有部门都实践了系统性的复用，取得的成果不断接近最初设想的目标。

软件是至关重要的因素，这不仅对我们是这样，对于我们绝大多数的客户也是如此。最初，只有少数人能够理解 Martin 的工作，现在，它已经在公司内外得到普遍接受和认同。很多人开始认识到，复用体系结构是大幅度降低开发成本并缩短时间的最有效的途径。现在，绝大多数程序开发人员仍然在修改他人编写的代码，因此，高效快捷地添加基于软件的功能的能力常常是评价软件开发能力的关键因素。

为了完成本书所介绍的重要工作，Martin 与杰出的面向对象方法论专家和业界软件工程师 Ivar Jacobson 进行了协作。在软件界，尤其是在爱立信，Ivar Jacobson 在软件设计和开发的理论与实践方面具有很大影响。具有开创性的 AXE 电信系统的成功，在很大程度上要归功于成功地运用了 Ivar 的思想。Svedberg 博士在其序言中已经阐述了 Ivar 的远见卓识和魄力在这个通信业里程碑系统中的关键作用以及在管理上给予的一贯支持。通过 Ivar 和 Martin 的通力合作，我们才有了软件复用的“梦之队”。正如本书作者所说，基于组件的软件工程的成功实践，要求高级管理层的承诺，高级经理需要了解使用设计良好、结构合理的软件组件构造复杂软件系统的基本原理和实践。吸收本书的精华是向这个方向努力的良好开始。

Joel Birnbaum 博士是惠普公司的高级副总裁和研发主管。

前言

本书面向的对像为希望显著改进软件开发工作的软件工程实践人员和他们的经理。对于很多工业和商业企业来说，达到关键业务目标，例如满足客户要求，通过产品和服务抢先占领市场或控制成本，都会直接影响选择什么样的方式开发和使用信息技术和软件系统以获得竞争优势。在大多数情况下，对象、基于组件的开发以及软件复用是其软件工程战略的关键部分。达到业界水准的面向对象软件工程的成功，要求将大规模软件复用付诸实践。本书将提供成功的程序框架。

系统的软件复用

自从 1968 年 Doug McIlroy 第一次提出共享组件库以来，软件复用一直被认为是具有明显回报的很有吸引力的思想。通过以前开发的高质量的组件构建软件系统，可以节省重复工作所耗费的成本和时间，并能够改进系统。很多年以来，获得高水平的复用一直是梦想。很多不同的技术、过程和组织问题影响了进展。但是尽管人们不断寻求各种其他“杀手锏”来改进软件开发，大家还是发现，系统的软件复用和基于组件的开发仍然是显著改进软件开发过程的最有效的途径之一。

大多数软件开发机构都开始采用对象技术，因为工程经理相信，对象技术会导致显著的复用。不幸的是，如果没有明确的复用程序和系统化的复用思想指导下的软件过程，这类采用对象技术的大部分工作并不能导致大规模的复用。

为什么使用“复用”这个术语呢？其他工程领域都不使用这个术语。相反，标准组件的系统化设计和使用是被普遍接受的实践；从汽车和齿轮到芯片，很多硬件组件手册每年都要出版，每天都被设计工程师研究。尽管 Brad Cox 使用了术语“软件集成电路”，微软公司和其他公司也鼓励基于组件的开发，但是广泛联合使用组件的软件组件行业仍然处于初级阶段。未来，基于组件的软件工程将会作为软件工程课程的标准部分讲授，“复用”这个词将会被废弃。今天，我们使用术语“复用”描述这种还在完善过程中的软件工程领域的目标。

在过去的 10 年里，软件复用和软件工程界已经对基于组件的软件工程有了更好的了解。在几乎所有的成功复用案例中，关键是管理层的支持、系统与组件体系结构、专门的组件开发团队、稳定的应用系统领域、标准以及组织支持。很多软件工程专著和会议都重点讨论了系统的软件复用和对象技术。很多专著都提到了复用、体系结构以及与过程和领域有关的应用系统开发，但是所提出的复用方式不同。有些对象技术专著，包括 Jacobson 等人 1992 年出版的有关“面向对象的软件工程（OOSE）”的专著，都直接论述了一些对象复用问题，或以相当大的篇幅讨论了复用。本书附录 B 提供了带有注释的有关参考文献。

本书的主题

本书将带领读者走进未来，未来面向对象基于组件的软件工程将成为规范。人们逐渐认识到，系统的大规模复用，结合对象技术，是从根本上改进软件开发过程的惟一途径。根据我们在惠普公司和 Objectory 公司（现在叫做 Rational 公司）进行复用研究，以及与很多客户接触的经验，我们相信，只有从根本上改变传统软件体系结构和开发过程才可以做到实质性的复用。首先，我们确信实质性的复用要求软件必须是考虑了复用的结构规划、设计、打包和支持。这里的体系结构表示系统和组件必须是结构化的，以便支持组件和系统的独立开发，以及稍后的集成和进化。其次，软件工程过程，包括具体的软件开发步骤、支持管理和组织结构、甚至与客户的交互模式，都必须能够使机构有效地运用这些可复用的组件。最后，必须把过程和工具集成到基础设施中，以支持关键活动。

将有效的复用引入软件工程领域，要求管理层和软件开发人员的一致和系统的努力，以克服业务、过程、组织和技术等方面的障碍，这些障碍常常阻碍有效复用。此外还必须尽早使客户、用户和维护人员参与其中。尽管这些问题和可能的解决方案已经被复用界充分认识，但是对于那些采用对象技术并期望它可以自动导致复用的人来说，仍然没有想到这些问题和解决方案。如果没有采用明确的复用程序和系统化的方法进行设计和处理，就不可能得到所期望的对象复用水平。

本书将开发一种一致的模型和一套确保大规模面向对象复用获得成功的方针。我们提出的称之为“复用驱动软件工程业务（简称为‘复用业务’）”的框架用于系统地解决这些关键业务、过程、体系结构和组织问题。

我们相信，系统化软件复用的理论和实践，以及系统化的基于模型面向对象的软件开发和业务工程的理论和实践已经足够成熟，能够帮助我们开发这种新的一致方法。我们的一致解决方案，吸收了我们致力多年的系统化方法、体系结构、与领域有关的软件工程以及复用适配等领域的最佳思想。我们的工作基于 Jacobson 的用例驱动的体系结构和过程建模基本原理，即“面向对象的软件工程（OOSE）（Jacobson 等，1992）”、《对象的优势（TOA）》（Jacobson 等，1994）所描述的“面向对象业务工程”和“统一建模语言（UML）”（Booch 等，1997）。

什么人需要复用驱动的软件工程业务

任何生产软件的机构都可以通过本书提出的思想受益。多年以来，很多大型机构把对象技术作为有意思但是仍然还在发展的一种技术密切地观察，现在已经开始增加对对象的承诺，并在业务再造和重新实现其支持信息系统方面投入大量资源。对于很多企业来说，有效的软件开发战略是其使用信息技术达到关键业务目标的基本组成部分，例如资源的有效使用、缩短的上市时间以及对市场变化的灵活响应。他们希望对象和复用可以根据商业和其自己的应用框架、可复用业务组件以及分布式面向对象中间件，为其提供经济高效的实现。

为了帮助这些企业达到其战略软件目标，本书论述了大规模结构化的对象复用如何帮助改进软件过程。本书提出的复用业务模型描述了这些软件机构如何向复用驱动的业务方向转变。软件机构只有在满足以下几点时才可以获得全面的收益：

- 生产作为某个产品线或产品系列成员的相关应用系统（或重要的子系统）
- 愿意在建立可复用体系结构、组件、过程和工具方面进行巨大投资
- 愿意进行一定的过程和组织变更

这种机构通常都会感到在更短的时间内交付更多应用系统，以满足更复杂的客户需求的不断增长的压力。

“复用业务”方法主要面向开发关键任务信息系统和软件产品是其成功的关键的机构。但是，本书也将论述“复用业务”如何用于能够从本书提出的系统化方法很多方面受益的机构，即使这些机构不能或没有选择完全使用本书提出的模型也是如此。在理想情况下，软件开发机构以软件工程业务方式运行。这里所说的业务，既指信息技术和软件工程目标是实现企业业务目标的关键，而且作为一种结果，软件机构本身又作为一种业务运营，具有明确的客户和财务目标。作为复用驱动的软件机构，要生产多个相关的应用系统（产品线或产品系列），重点是优化组件的生产和复用。作为一种业务，这种机构必须理解其客户并且为满足其需求服务，同时又要有效地达到赢利和支出目标。这种业务上的折中全面考虑要使用经济、产品与过程手段进行管理。

我们的经历

1993年以来，本书的几位主要作者在熟悉了彼此的工作之后，开始在研究小组、会议、研讨会以及共同研究中进行密切合作，以综合我们的技能和经验，形成本书所介绍的共同观点。我们在技能方面各有所长，相互补充。Ivar Jacobson带来了很强的体系结构方面的观点和在爱立信开发 AXE 电信交换系统所获得的经验，此外，他还有 OOSE 开发和 Objectory 公司客户服务方面的经验。Martin Griss 在复用组织、适配以及技术方面拥有广博的经验。两人在系统化的软件过程定义与开发方面，都具有丰富的经验。Patrik Jonsson 于 1994 年 5 月参加了进来，并为项目组带来了他的过程经验，最终我们决定共同努力形成本书所介绍的正式协作。

在协作中，我们每个人扮演了不同角色，在不同时间完成了不同的工作。大家一致同意作者的排列顺序，认为这种顺序最好地反映了我们对本书技术内容的贡献情况，这是衡量这种工作的最重要的标准。

Ivar Jacobson 是 OOSE 方法的发明者，是瑞典 Objectory AB 公司的创始人。他目前是 Rational 软件公司业务工程部的副总裁，在这之前是 Objectory 公司技术部的副总裁。他是面向对象界的领导者，以开创性的工作和 20 多年使用对象方法进行大型实时系统设计而著称。他在爱立信公司工作了 25 年，致力于 AXE 交换系统的研发。在爱立信公司工作期间，他开发了一种体系结构和软件工程过程，以支持大规模复用。他早期基于对象的设计技术促进了国际 CCITT/SDL 电信标准的产生。他是两部很有影响专著的主要作者，即《面向对象的软件工程——一种用例驱动方法》和《对象的优势——采用对象技术的业务流程再造》，以及很多对象技术方面的被广为引用的论文的主要作者。他在用例工程方面的工作，影响了今天使用的几乎所有面向对象方法。他还为 OOPSLA、ECOOP 和 TOOLS 程序委员会工作。

Martin L. Griss 是位于加利福尼亚州帕洛阿尔托的惠普实验室的高级实验科学家。在该实验室工作的 14 年里，他一直从事软件工程过程和系统、系统化的软件复用、面向对象的复用

以及度量系统包的研究。他是惠普公司专业服务组织部的高级复用问题顾问，在“面向对象解决方案中心”工作。作为惠普公司的“复用先生”，他领导了软件复用过程、工具和软件工厂的研究，创建了惠普公司的复用程序，并将软件复用系统化地引入惠普公司的各个部门。他是惠普公司实验室的软件技术实验室主任，在软件工程研究方面具有 25 年以上的经验。他以前是犹他大学的计算机科学副教授，现在是那里的客座教授。他发表了大量有关软件工程和复用的论文和报告，为《对象杂志》撰写复用专栏文章，并积极参与多个复用程序委员会的工作。

Patrik Jonsson 是第一批加入 Objectory AB 公司的人员之一，与 Ivar 一起工作了很多年。他曾经参加了多种不同的活动，包括客户项目、过程和方法的开发，为 Objectory CASE 工具定义需求并进行教学。他是 Addison-Wesley 出版公司的畅销书《面向对象的软件工程——一种用例驱动方法》的作者之一。他目前是 Rational 公司的高级顾问。Patrik 参加了 Ivar 和 Martin 的工作，帮助整理和表达他们的思想。他是将复用驱动的软件工程业务概念和 OOSE 软件工程框架结合在一起的主要推动者，并在其中增加了很多自己的思想。

本书的内容组织

本书由四个部分组成。

第 1 部分：复用驱动的软件工程业务的引入将介绍有关系统化的复用驱动方法，简称复用业务的动机、背景和概况。第 1 章将回顾软件复用的发展情况以及促使我们采用复用业务方法的关键的管理、体系结构、过程和组织原则。第 2 章要描述复用业务的整体概念和目标。这一章要定义应用系统和组件系统，描述“应用系统工程”和“组件系统工程”的关键过程，前者使用组件系统构建应用系统，后者创建可复用的组件系统。

第 2 部分：体系结构风格将介绍复用业务背后的体系结构概念和标记。这些体系结构构件块、连接器和合成规则使开发人员可以描述各种体系结构。第 3 章将介绍“面向对象的软件工程 (OOSE)”和“统一建模语言 (UML)”，包括模型、执行者、用例、对象、类型、类和系统。第 4 章介绍应用系统、组件和组件系统。这里把正面 (facade) 的概念定义为一般化的接口，并描述提供可管理而又非常灵活的复用的机制。第 5 章和第 6 章详细论述用例和对象组件。第 7 章介绍互操作系统的分层体系结构和系统，以便支持大规模可控复用。

第 3 部分：过程将讨论系统化地创建、使用和管理第 2 部分介绍的体系结构要素所需要的面向复用的软件工程过程。第 8 章将介绍“面向对象的业务工程”，包括业务系统、业务过程和工作人员的建模。业务模型要与信息系统和人力资源关联。“面向对象的业务工程”和 OOSE 可以结合使用，以开发并精确描述过程、组织模型以及复用业务的下层工具。第 9 章将介绍复用业务的高层业务模型，这个模型为后续各章介绍的关键过程和组织的详细处理提供一种框架。这里定义并把“组件系统工程”、“应用系统工程”、“应用系列工程”与复用管理结合起来。第 10 章将业务工程运用于客户业务，以定义支持其业务过程的应用系统套件。然后使用“应用系列工程”分析这些应用系统，最终形成整体分层体系结构，并分解为可复用的组件系统。此外这一章还讨论了结合老系统的问题。第 11 章将讨论“组件系统工程”，解释如何分析成套需求以产生高质量的组件及其正面。第 12 章讨论“应用系统工程”，解释如何通过选择、定制和复用来自一个或多个组件系统的组件，构建应用系统。

第4部分：组织复用业务为建立符合复用业务模型的具体复用业务提出建议。第13章将介绍向复用业务的系统化转移，将业务工程技术、变更管理和与具体复用有关的方针结合起来。这将导致渐进式的适配过程，同时要改变一些过程和组织。基本复用业务模型可以用于各种软件开发环境。业务工程技术用于划分复用角色和部门，以为具体复用业务创建组织结构。有一些工具和技术可以帮助管理过程以及创建和打包组件系统。第14章将介绍如何创建和管理各种过程的实例，使用哪些技术与经济、过程以及产品度量手段管理全面折中和过程流，需要哪些组织和技术基础设施支持复用业务。第15章将对关键体系结构、过程和组织原则以及高级管理层在指导读者下一步的工作中所起的重要作用进行小结。

本书的附录将提供详尽的参考材料。附录A提供了完整的术语解释，附录B列出并解释了主要参考读物和在线资源，附录C归纳了统一建模语言标记和扩展使用，附录D列出了本书的所有参考文献。

本书的主要内容

这是一本关于业界面向对象软件工程实质性复用的专著，主要目标是介绍在构建复用驱动的软件工程时会遇到的各种关键问题。读者理解本书不需要具备任何对象或复用方面的经验。

本书并不是一本完备的手册，因此没有提供实现并运行复用业务需要的所有详细信息。更详细的过程指南、培训以及实际经验，对于确保成功都是非常必要的。就像OOSE和面向对象的业务工程是对更详细、更精确的Rational Objectory SE和BE过程的简化一样，本书描述的工作是对更复杂的过程的简化。

作为复用业务潜在参与者的所有读者来说，深刻理解第1章和第2章所介绍的关键概念，以及第1到第4部分的引言部分是非常重要的。此外：

- 负责批准和资助向复用业务转换的上层管理人员，还应该认真阅读第13章、第14章和第15章。
- 负责复用业务日常运行的复用管理人员应该阅读所有章节。
- 项目管理人员应该阅读第2部分中的第3章、第4章和第7章，以及第3部分介绍的过程。
- 软件工程师、系统结构设计师、组件开发人员、复用开发人员以及组件系统维护人员都应该熟悉第2部分的所有内容，以及在第3部分中所介绍的最适合自己的特定过程。

这些读者可以有以下收获：

高层管理人员需要理解经济、政治和组织方面的后果，以及投入市场的时间、质量和成本效益。

复用管理人员需要建立、运行和改进大规模复用机构。

项目管理人员需要学习如何运作应用系统项目组，为了利用复用需要对哪些过程进行修改。

系统结构设计师需要设计适合实质性复用和进化的体系结构。

组件开发人员需要学习如何设计和构建可复用组件系统。

复用开发人员需要学习如何使用组件系统构建应用系统。

组件系统维护人员需要打包和支持组件系统，并提供相关的服务。

虽然本书的主要读者对象是没有多少面向对象软件或系统化软件复用实际经验的经理和实践者，但是具有其中一个或两个领域经验的读者仍然会学到很多有价值的东西。特别是：

- 已经熟悉系统化软件复用概念的读者，例如领域工程，以及创建或使用过面向复用的过程和机构的读者，会通过本书了解到如何将这些概念集成到重要的面向对象方法、OOSE 中，以及如何充分利用面向对象的业务工程。
- 已经具有面向对象分析和设计方法经验的读者，会通过本书了解到如何适配和扩展 OOSE 体系结构和过程，以协调大规模系统化复用议程。
- 已经了解 OOSE 的读者，会通过本书了解到如何扩充 OOSE，以充分利用新的统一建模语言（UML）、软件体系结构和复用过程与技术。

当我们开始编写本书的时候，曾经设想要根据我们多年的经验，直接合并我们每个人所了解的有关面向对象的软件工程、业务工程和系统化的软件复用等方面的知识。但是随着工作的展开，我们发现还需要进行很多技术上的创新、综合、扩展，并提出很多思想。这些创新包括：

- 使用统一建模语言（UML）重新解释 OOSE。当我们进行这项工作时，对 UML 设计作了多处体系结构、过程和复用扩充。
- 扩展并开发了 UML、OOSE 和面向对象的业务工程，以解决支持系统化软件复用的关键技术、过程和渐进式适配等方面问题，并为组件、正面、分层体系结构和可变要素设计了新的结构。
- 发现了如何使用和扩充 OOSE 和面向对象的业务工程，以系统化地开展复用领域工程和体系结构开发。
- 发现了需要如何改变业务过程、软件工程过程和组织，以优化基于实质性复用的企业信息系统的开发。

我们尽量使本书使用的标记与统一建模语言保持一致，在修改本书的终稿时，统一建模语言仍然在发展。在出版之前进行的一些修改能够保证本书与 1997 年 1 月发表的 UML 1.0 100% 兼容，希望没有任何遗漏。

致谢

很多人提供的灵感和鼓励最终导致本书得以出版。有些人对本书的形式和内容提出了很好的建议，有些人参加了实践并提供了反馈意见，有些人对本书的思想进行了无保留的讨论。

特别感谢 Ramesh Balasubramanian、Per Björk、Stefan Bylund、Patricia Cornwell、Nathan Dykman、Håkan Dyrhage、Christian Ehrenborg、John Favaro、Daryl Foy、Steven Fraser、Agneta Jacobson、Sten Jacobson、Per Kröll、Philippe Kruchten、Reed Letsinger、Mary Loomis、Ruth Malan、Patricia Markee、Vered Marash、Susan McBain、Ware Myers、Keith Moore、Karin Palmkvist、Joe Podolsky、Jeff Poulin、Mats Rahm、David Redmond-Pyle、Mark Simos、Mike Short、Greg Siu、Kevin Wentzel 和 Lorna Zorman，感谢他们的评论和所作的工作。

特别感谢 Karin Palmkvist、Susanne Dyrhage 和 Staffan Ehnebm，他们与 Ivar Jacobson 一起，开发了“内部互联系统”的概念。感谢 Stefan Bylund、Christian Ehrenborg、Magnus Christerson、

Staffan Ehneborn 和 Gunnar Overgaard, 他们参加了本书出现的接口、过程和物理设备标记的开发。感谢 Gunnar Magnusson 和 Håkan Dyrhage, 他们对将面向对象的业务工程运用到软件开发机构提供了很有帮助的建议和评论。Patricia Cornwell 是惠普公司的重用成熟度模型和渐进式迁移战略开发工作的主要参加者, 这种战略对本书第 13 章的思想具有很大的影响。

Intecs Sistemi 公司的 John Favaro 把重用业务的初步版本应用到了意大利电信业的重用中, 并提供了很有价值的反馈意见。他发现组件系统、分层体系结构、显式扩展点、变体与变化点以及用例组件的概念, 对于源自 FODA 的领域分析方法论是强有力的补充。他的意见对于澄清并深化一些概念的解释很有帮助。

尤其要感谢 Ware Myers 所付出的辛勤劳动, 他对所有各章作了评论, 并多次帮助我们重新编写本书的一些章节, 以达到更好的效果。没有他的帮助, 我们难以按时完成本书, 也不会有现在的质量。还要感谢 Lorna Zorman, 他认真阅读了本书很多章节的初稿, 并提出了很好的改进意见。

感谢惠普公司和 Rational 公司对本书写作的支持, 感谢 Mary Loomis 的鼓励和支持, 它使我们从头到尾顺利地完成了本书的写作。

最后, 感谢我们的家庭成员和伙伴, 在写作本书的过程中, 我们经常占用周末和夜晚的时间, 对此他们表现出了极大的耐心。

瑞典 Stockholm

Ivar Jacobson

ivar@rational.com

美国加利福尼亚州帕洛阿尔托

Martin Griss

griss@hpl.hp.com

瑞典 Stockholm

Patrik Jonsson

patrik@rational.com

注意:

- 本书页边标注为英文原书页码, 便于读者参考原书的内容。
- 索引中标注为英文原书页码, 用于查阅。

目 录

序言

前言

第一部分 复用驱动的软件工程业务的引入

第 1 章 软件复用的成功因素	2
1.1 软件复用是一种简单的思想	3
1.2 组件推动着应用系统开发的革命	4
1.3 系统化的方法才能取得复用 工作的实效	4
1.4 爱立信和惠普公司复用经验揭示的 通用原则	6
1.5 复用要求改变过程	10
1.6 复用要求机构的变化	13
1.7 系统、渐进地采用复用	15
1.8 从其他复用程序输入	17
1.9 需要一套原则	19
1.10 小结	20
1.11 补充读物	20
第 2 章 复用驱动的软件工程业务	21
2.1 这是所需要的业务吗	21
2.2 使复用在经济上合算	23
2.3 复用业务具有业务特征	24
2.4 结构组件与应用系统	26
2.5 软件工程过程	29
2.6 建立与管理复用业务	31
2.7 小结	34
2.8 补充读物	35

第二部分 体系结构风格

第 3 章 面向对象的软件工程	39
3.1 软件工程将需求转换为代码	39
3.2 软件工程是一种团队过程	39
3.3 软件工程是系统化的模型构建	40
3.4 对象统一建模过程	43

3.5 用例模型获取系统需求	47
3.6 分析模型形成系统体系结构	50
3.7 设计模型定义实现	54
3.8 实现模型是代码	56
3.9 测试模型检验系统	56
3.10 小结	57
3.11 补充读物	57
第 4 章 应用系统与组件系统	59
4.1 应用系统开发人员可以复用 OOSE 模型组件	59
4.2 应用系统系列使显著复用成为可能	60
4.3 通过可复用组件构建应用系统	61
4.4 把组件划分为组件系统	62
4.5 正面控制对组件系统内部的访问	64
4.6 正面与组件系统是特殊的软件包	65
4.7 组件系统通过正面输出组件	66
4.8 复用之前具体化一些组件	69
4.9 可变性出现在变化点上	71
4.10 使用多种可变性机制	72
4.11 复用可变组件构建应用系统	76
4.12 打包并形成组件系统 文档以便复用	79
4.13 小结	81
4.14 补充读物	81
第 5 章 用例组件	83
5.1 构建用例模型确保组件复用	83
5.2 用例模型形成系统其他部分的框架	83
5.3 复用组件以构建用例模型	84
5.4 设计用例组件以便有效复用	85
5.5 并不是所有用例都应该是可复用组件	86
5.6 复用具体或抽象执行者与用例组件	87
5.7 表达用例的可变性	89
5.8 打包与建立用例组件文档	94

5.9 小结	95	形式交付模型	159
5.10 补充读物	96	8.4 使用业务执行者和用例表示增值过程	159
第6章 对象组件	97	8.5 使用工作人员和实体类型表示 人员和结果	160
6.1 对象模型定义系统体系结构 和设计	97	8.6 根据技能将工作人员划分为能力单元	162
6.2 复用分析和设计组件	98	8.7 信息系统必须支持业务用 例和工作人员	163
6.3 表达对象模型组件中的可变性	99	8.8 小结	165
6.4 用例可变性到对象模型的跟踪	100	8.9 补充读物	166
6.5 可复用分析组件	106	第9章 使用业务工程定义 过程和机构	167
6.6 与类型和类相关的子系统组件组	109	9.1 复用业务与体系结构匹配的 过程和机构	167
6.7 可复用设计和实现组件	114	9.2 复用业务中的软件工程过程	168
6.8 对象组件和变体的打包与文档建立	121	9.3 将工作人员组织为能力单元	176
6.9 小结	123	9.4 复用业务过程之间的相互影响	182
6.10 补充读物	123	9.5 小结	184
第7章 分层体系结构	124	9.6 补充读物	185
7.1 体系结构定义系统的结构、接口和 交互模式	124	第10章 应用系统系列工程	186
7.2 好的体系结构对于维护系统的完整 性至关重要	124	10.1 为应用系统系列开发体系结构	186
7.3 根据通用程度使用分层体系 结构组织软件	125	10.2 根据用例优先级规划产品开发进度	188
7.4 分层体系结构降低软件依赖性	128	10.3 应用系统系列工程第一步：获取对 体系结构有影响的需求	191
7.5 中间件层支持分布式对象计算环境	128	10.4 应用系统系列工程第二步：运行 健壮性分析	197
7.6 与特定业务有关层支持快速应用 系统开发	135	10.5 应用系统系列工程第三步：设计 分层系统	201
7.7 处理分层系统体系结构时 使用多个模型	138	10.6 应用系统系列工程第四步：将体系 结构实现为分层系统	209
7.8 把分层系统表示为上级系统	139	10.7 应用系统系列工程第五步：测试 分层系统	209
7.9 分层系统关系中的用例	143	10.8 管理体系结构变更	210
7.10 应用系统和组件系统的执行者	147	10.9 通过工作人员表示应用系统 系列工程	211
7.11 应用系统和组件系统的用例	148	10.10 实施应用系统系列工程的较瘦方法	212
7.12 包装老系统以适应体系结构	149	10.11 小结	212
7.13 分层系统的分布式处理与节点	150	10.12 补充读物	213
7.14 小结	153	第11章 组件系统工程	214
7.15 补充读物	153	11.1 构建灵活的组件系统	214
11.2 组件系统工程第一步：获取			
第三部分 过 程			
第8章 面向对象的业务工程	157		
8.1 业务流程再造获得巨大改进	157		
8.2 面向业务流程再造的定义完备的过程	158		
8.3 业务工程通过未来发展图的			

重点是可变性的需求	215
11.3 组件系统工程第二步：执行健壮性分 析以尽可能提高灵活性	219
11.4 组件系统工程第三步： 设计组件系统	223
11.5 组件系统工程第四步： 实现组件系统	224
11.6 组件系统工程第五步： 测试组件系统	226
11.7 组件系统工程第六步：用于复用的组件 系统的最后打包	227
11.8 通过工作人员描述组件系统工程	229
11.9 小结	230
11.10 补充读物	230
第 12 章 应用系统工程	231
12.1 通过可复用组件构建应用系统	231
12.2 应用系统工程第一步：获取需求	233
12.3 应用系统工程第二步：执行健壮性分析 以实现灵活的应用系统	236
12.4 应用系统工程第三步、第四步和第五步： 设计、实现和测试应用系统	237
12.5 应用系统工程第六步：打包应用 系统以便于安装	238
12.6 通过工作人员表示应用系统工程	239
12.7 小结	240
12.8 补充读物	240

第四部分 组织复用业务

第 13 章 向复用业务迁移	242
13.1 系统化的渐进迁移控制风险	242
13.2 渐进迁移过程	244
13.3 向复用业务迁移第一步：发出对现 有软件业务实施再工程的指示	250
13.4 向复用业务迁移第二步： 构想新的复用业务	253
13.5 向复用业务迁移第三步：对现有软件 业务实施逆向工程	257
13.6 向复用业务迁移第四步：对新的复 用业务实施前向工程	258
13.7 向复用业务迁移第五步：	

实现复用业务	262
13.8 小结	266
13.9 补充读物	267
第 14 章 管理复用业务	268
14.1 持续管理对于 RSEB 的成功至关重要	268
14.2 度量是管理复用业务的关键	270
14.3 经济模型与复用投入决策	275
14.4 向复用业务迁移第六步： 不断过程改进	283
14.5 人员与机构管理	283
14.6 小结	287
14.7 补充读物	287
第 15 章 后记：运营复用业务	289
15.1 把所有内容串联在一起	289
15.2 复用改进业务过程的性能	289
15.3 常见的错误概念	289
15.4 推行复用是困难的	290
15.5 没有构想，员工就会失去活力	292
15.6 复用依赖于体系结构	293
15.7 管理层通过机构发挥作用	294
15.8 复用业务必须实现投资回报	295
15.9 软件工程依赖于过程	296
15.10 对象技术可以帮助过程	298
15.11 业务工程：检查业务模型	299
15.12 小结	300
15.13 补充读物	302

第五部分 附 录

附录 A 名词解释	304
附录 B 带注释的书目	319
B.1 系统化软件复用	319
B.2 面向对象技术	322
B.3 体系结构与模式	325
B.4 软件工程	327
B.5 业务过程再工程与机构变革管理	328