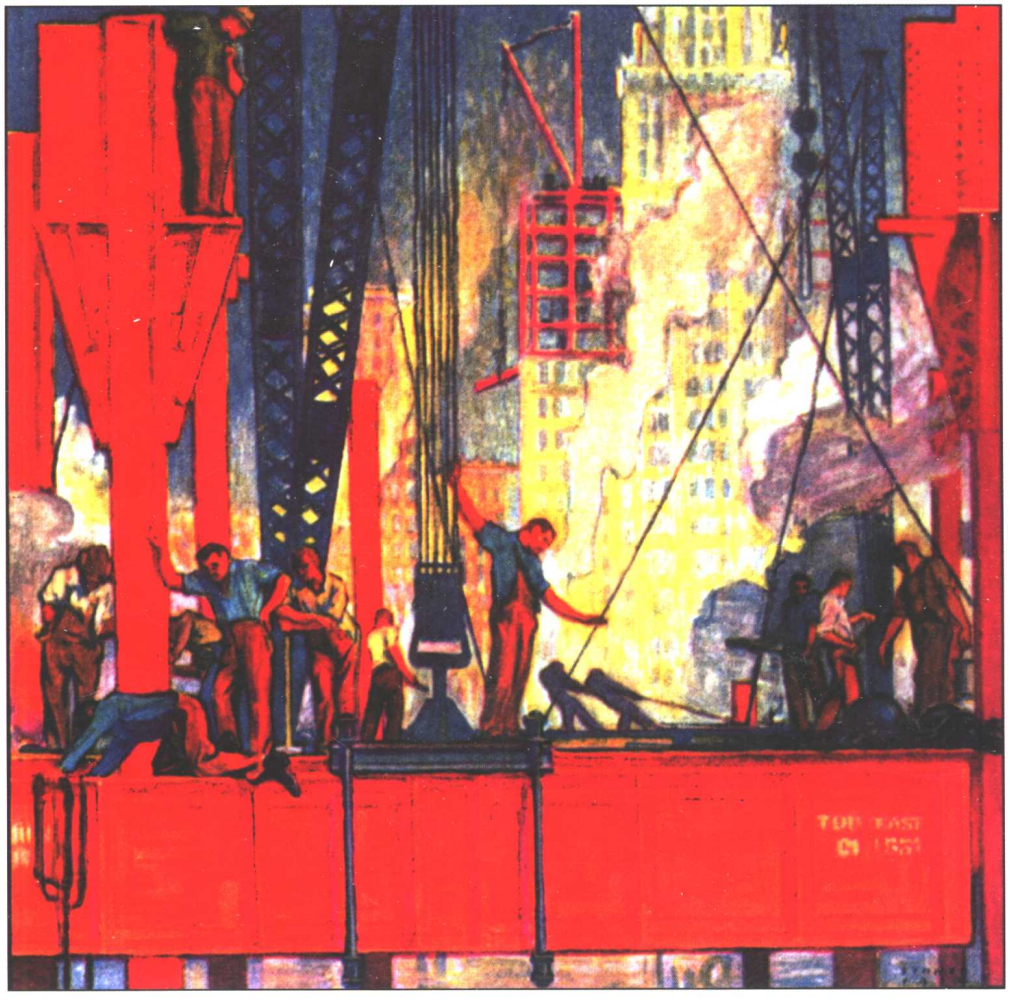
Addison Wesley

软件工程经典系列

Lessons Learned Developing Application Frameworks

框架过程模式

Framework Process Patterns



〔美〕 James Carey 著
Brent Carlson
林星 张夏 译

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

软件工程经典系列

框架过程模式

[美] James Carey Brent Carlson 著

林星 张夏 译

人民邮电出版社

图书在版编目（CIP）数据

框架过程模式 /（美）凯里（Carey, J.），（美）卡尔森（Carlson, B.）著；林星，张夏译.
—北京：人民邮电出版社，2003.6
（软件工程经典系列）
ISBN 7-115-11181-2

Ⅰ. 框… Ⅱ. ①凯…②卡…③林…④张… Ⅲ. 软件开发－基本知识 Ⅳ. TP311.52

中国版本图书馆 CIP 数据核字（2003）第 028398 号

软件工程经典系列

框架过程模式

◆ 著 [美] James Carey Brent Carlson
译 林 星 张 夏
责任编辑 俞 彬

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 http://www.ptpress.com.cn
读者热线 010-67132705
北京汉魂图文设计有限公司制作
北京顺义振华印刷厂印刷
新华书店总店北京发行所经销

◆ 开本：800×1000 1/16
印张：12.75
字数：278 千字 2003 年 6 月第 1 版
印数：1-4 000 册 2003 年 6 月北京第 1 次印刷

著作权合同登记 图字：01－2002－5940 号

ISBN 7-115-11181-2/TP・3392

定价：28.00 元

内容提要

本书将日常面向对象框架开发中遇到的（也是典型的）情况整理为模式的形式。其开发活动涵盖了从初始需求收集到框架文档制作的范围。本书共 10 章，第 1 章描述框架是什么，如何开发和使用框架。第 2 章提供了一个通用领域以及领域中的词汇，这样读者就可以提出领域中的问题，并引发对模式的思考。接下来的第 3 章至第 9 章则根据模式的主要用途，组织成相关的模式专题介绍。最后简要叙述了如何使用框架来开发应用。附录 A 描述了组件和框架之间相互补充的关系。附录 B 描述了 IBM SanFrancisco 项目中用于业务应用框架开发的开发过程。

本书是一本有关软件开发的优秀书籍，适合于软件开发人员、项目管理人员阅读参考，对相关专业研究人员也有很好的参考价值。

译者序

国内的软件业对于框架这个概念还比较陌生。事实上，框架是软件产品的一种形态。不同的软件形态决定了软件开发目的的不同，以及软件开发过程的不同。项目、产品、框架都是软件的不同形态，虽然它们针对的目标领域可能是相同的，但是由于形态的差异，导致了它们之间极大的不同。项目针对的是特定的用户群，例如某个企业。它不需要很大的灵活性，因为需求是相对固定的，但有很强的特殊性，必须能够满足用户的一些极端需求。产品针对的是某个领域的用户，例如某种类型的企业。它需要满足这一类用户群中的通用需要，并可以根据不同用户的需要做一定程度的定制。而框架是针对那些开发应用（项目或产品）的人而设计的，它的目标是使那些开发应用的人能够快速地制造出高质量的软件。这一目标决定了框架开发具有高度的灵活性，并允许将部分的实现留由应用开发人员完成。

虽然我们做出了如上的区分，但是在现实中，我们可能很难严格的按照这三种标准来区分软件。很多的软件可能具有产品特色，但又是以项目为单位开发的，甚至可能已经发展出一定的框架来了。本书所描述的内容，不但适合用于框架的开发，也同样适用于产品和项目的开发。而区别在于，某些对于框架特别重要的模式或技巧，对其他形态的软件而言可能并不是那么重要。例如，框架需要为不同用户建立不同针对性的文档（参见 Give 'Em What They Want 模式），但对于项目开发而言，这种需要就显得不那么急切了。

本书其实是一本项目管理的书，但是由于作者本身有着多年的面向对象框架的开发经验，书中讨论的内容和开发世界相当的接近。书中暗含了敏捷开发过程的很多思想和原则，尽管书中并没有提到敏捷这个词。敏捷的本质是根据现实情况（项目特点、组织特点）

进行权衡，本书的大部分内容正是围绕这一中心思想进行讨论的。如果认识到这一点，在阅读本书的时候，你就会对书中模式的上下文和你的现实环境进行比较，再做出适合自己的决策。这才是敏捷思想的真谛。

可以看出，本书中模式很多都来源于作者曾经经历的 IBM SanFrancisco 框架的开发。IBM SanFrancisco 框架是一个用于企业应用开发的框架，为企业应用提供了很多基础的支持。除了本书之外，作者还根据 IBM SanFrancisco 框架的设计经验，编写了一本设计模式的书（SanFrancisco Design Patterns[Carey 00]），在看完这些书之后，你可能会有一种立刻将这些模式应用于实践的冲动。请注意！不要忘了我们刚刚提到的敏捷思想，确保你真正理解了模式的上下文环境和模式的目的，并且你已经根据实际的情况对模式进行了调整。这样你才能够从本书介绍的模式中获益。事实上，这一原则适用于所有的软件工程类书籍。

作者在前言中，提到了本书主要是针对面向对象开发者的，事实上，对于非面向对象的项目开发者或产品开发者而言，本书也同样适用。除了使用技术的不同，他们面对的问题大致相同。但是我几乎无法想象一个非面向对象的框架，这种软件的开发难度似乎难以想象。

总之，本书不失为一本软件开发的优秀书籍。值得您花费宝贵的时间来阅读并思考其展现的内容。我们也尽了最大的努力来翻译本书，尽量做到把作者的思路完整的再现给读者们。对于书中一些容易引起歧义的地方，我们都做了注释。如果在阅读本书时仍发现有不容易理解的部分，请写信给我们，我们非常期待着与您共同进行探讨。

林星　张夏

2002 年 12 月于厦门

前 言

要如何才能创建一个面向对象的应用框架呢？又该如何创建能够提供应用核心，可用于构建大量应用的呢？面向对象编程技术是一个很好的开端，但对于创建一个成功的框架而言，这还不够。那么还需要些什么呢？

你们中的大多数人都经历过从过程式编程迁移到面向对象编程的变革。适应了这种变革，你才能够自如的运用 Java 或 C++编写程序，但是你仍然无法积极地使用面向对象编程技术（除非有奇迹发生）。通过阅读和接受培训（以及犯错误），你慢慢地获取经验，这些都在潜移默化的影响着你。你开始接收一系列面向对象编程的模式，它们帮助你正确的使用继承，使你能够真正进行面向对象设计，并帮助你形成有效地开发团队。要成为一个框架开发者，你也要经历类似的转变，这个转变虽然不大，但确实存在。

是什么使得框架如此独特呢？框架是一门艺术，它在提供可重用的内容（例如供业务应用使用的，预定义的业务对象和业务流程）和灵活性（允许对内容进行定制，以准确地满足框架用户需要）之间保持一种平衡。这是一种极其微妙的权衡艺术。如果过度的提供灵活性，那么框架就会分裂成微小的业务对象和低层次的业务流程——更像是一个类库。另一方面，如果过分注重内容，我们就会面临着使定制复杂到难以使用的风险——就像使用一个现存的系统来构建另一个系统一样。如何有效的找到这个平衡点，是那些希望成为框架设计者所需要学习的。

为了帮助你完成这种转变，本书介绍了成为一位框架开发者所需要学习的基本内容。我们把自己的经验精练为模式，这样你就可以立刻把它们应用到工作中——我们相信，这些模式不但对那些和框架打交道（开发者、使用者和评估者）的人有用，还能被其他软

件开发者所利用。框架开发能够测试并强化我们在成为面向对象开发者途中所学到的知识，因此该书也能够帮助那些面向对象开发者。此外，随着越来越多可定制的应用、组件、Web Service 的出现，有关技术团队（非领域团队）和非技术团队（领域团队）协作开发软件的模式就越发显露出其价值所在。

本书并不是一本描述面向对象开发的书，也没有向你讲述如何使用一门特定的面向对象语言编写程序。它也不曾提及神秘的算法、彻底革新的开发过程，或是对面向对象设计建模的新方法的讨论。实际上，我们只是把在日常的面向对象框架开发中遇到的（也是典型的）情况整理为模式的形式。其开发活动涵盖了从初始需求收集到框架文档制作的范围。我们还针对开发中人的问题投入了一些精力——如何协调不同类型的人员交互，以保证他们相互协作，开发出成功的框架。最后我们简要的叙述了如何使用框架来开发应用。

如果你有面向对象的开发经验，你会从本书中得到最大的价值。如果你刚开始学习面向对象的知识，这本书对你还是有价值的——它介绍了未来你会遇到的一系列问题。框架开发者和使用者也能够从本书中收益。框架用户不仅会学习到使用框架的模式，还能够了解到如何进入框架开发的世界。面向对象开发的项目经理们也可以发现本书的用处，它指出了软件开发过程中可能出现的一些潜在的缺陷和开发团队要处理的主要的问题。

这些模式到底是什么呢？我们拿 Tor's Second Cousin 模式（见 4.2 节）做一个例子。它是用一个在 IBM 的 SanFrancisco 框架中工作的领域专家的名字命名的。Tor 能够在框架中找出别人无法发现的需要灵活性的部分。我们把这些发现归功于 Tor 的虚构的、疯狂的表弟。在很多情况下，Tor 描述的场景表达简单但考虑了过多灵活性的需求。换句话说，只有极少数的框架用户才会按照 Tor 描述的方式那样使用框架。如果你是项目开发团队中的领导者，当遇到这种情况的时候，你就需要做出决定：是否需要在框架中囊括对该种灵活性的支持？Tor's Second Cousin 模式正是捕获了这种处于灵活性和复杂性之间的平衡艺术。框架的某些目标受众需要灵活性，但它却引入了额外的复杂性，而这恰恰是需要被避免的。这种规则提供了一种方法，令你能够对情况进行研究，同时不至于触怒他人。询问需求是否满足 Tor's Second Cousin 模式并不是意味着需求是错误的，或是说提出需求的人是傻冒或出格。它只是提醒我们注意复杂度和灵活性之间的权衡而已。给这条规则冠以一个幽默的名称，这样我们就可以在对需求发生争执的时候缓解紧张的气氛。我们发现，当幽默运用得当时，对于构建团队和缓解紧张气氛来说非常有效，特别是当软件开发的时间紧迫，面临着巨大的压力的时候。出于这些因素的考虑，本书中的大部分模式除了它们的描述性的名字外还有一个幽默的名字（译注，这也是我们保留这些原汁原味的

幽默的原因）。举个例子，Tor's Second Cousin 模式的另一个名称是怎样才是真正的极端？

为什么采用过程模式呢？Scott Ambler 把过程模式定义为一种经过证明的、成功的软件开发方法[Ambler 99]。本书提供了框架开发中的经过证明的成功方法。把这些方法捕捉为模式，使得我们能够以一种一致的方式来记录它们，这样你就可以很容易的检视这些模式，参考它们的适用性，来判断是否适合用于某个特殊的情形中，或者应该把它作为一个定义自己的解决方案的起点。所有的模式都包含下列的部分：

名称——模式的名称（通常都比较幽默）。

别名——模式的另一个名称。

意图——模式内容的简要叙述。

上下文——模式的动机。

示例——示例，通常是基于案例研究展开的。

问题——对模式所处理问题的简要描述。

方法——各种的解决方案，有不同的侧重点，同样是基于案例研究的。

解决方案——模式对问题的推荐解法的简要描述。

何时使用/何时不使用——是否使用模式的思考，一般是研究基于案例的。

适用性——是否使用模式的简要描述。

已知应用——模式曾经的应用之处，包括 IBM 的 SanFrancisco 框架例子。

相关模式——和当前的模式相关的其他模式，以及它们的关系。

我们的经验

和你们之间的许多人一样，我们原先也属于过程式编码人员阵营。我们在课堂上学习面向对象技术，并在实践中磨练自己的技艺。我们都参与过面向对象操作系统的开发（使用 C++）。我们的工作属于 20 世纪 90 年代初把 AS/400 从 CISC 转移到 RISC 项目的一部分。该项目涉及到对数百万源代码的修改和替换。组织中的大多数人都是面向对象开发的新手，因此我们并不只是关注自己的设计和代码，相反的，团队中的每一位成员都抱着开放的心态，相互帮助，讨论和复审彼此的设计和代码。当我们完成从 CISC 到 RISC 的转换时，我们所有人都成为了面向对象的专业开发人员。这为我们撰写此书提供了最初的经验。

从项目中出来之后，我们都投入到了 IBM 的 SanFrancisco 项目中去，准备开发一个分布式的面向对象业务应用框架（使用 Java）。我们设定了一个目标——为应用开发者提供一个应用核心，换句话说，提供每个人都需要的部分。框架使得应用开

发者能够在应用的特殊性上投入更大的精力，增加和竞争者不同的内容，而不是提供和他人完全一致的内容。框架还允许应用开发者针对特殊的需求定制框架元素。

SanFrancisco 框架是以多层的形式交付的。在底层的是更加传统、更面向技术的框架，提供了对诸如持久性和分布式对象的支持（目前的大多数功能都已被 EJB 实现了[1]）。在技术框架上构筑的是业务应用框架。框架的这些部分并非提供可重用的技术解决方案，而是提供了可重用的业务内容——业务对象和业务流程。而且，根据框架应用和目标领域的不同，这部分的框架被细分为不同的应用，例如库存管理和会计。虽然我们参与了大部分层次的开发工作，但我们的重点保持在提供业务内容的业务应用框架上。

SanFrancisco 的开发团队之所以选用面向对象技术，是基于以下几点原因：对复杂性的隔离和分解、对可维护性的改进，以及把开发工作分解为自含的模块。面向对象技术吸引人的一个主要原因是它把职责封装到对象中。这使得对框架某部分的修改得到控制，也容易理解，对于为实现定制性而需要改变的框架来说，该特性是非常关键的。我们选择 Java 的原因是它的平台无关性和因此而产生的最大灵活度。

当我们结束项目时，IBM 的 SanFrancisco 框架已经拥有超过 1000 个面向业务的类，近 100 万行的 Java 代码，这使得 IBM 的 SanFrancisco 框架成为有史以来完成并销售的、规模最大的商业化面向对象业务应用框架。

这么多类和代码难道都在本书的介绍范围之内吗？并非如此。它们仅是我们为处理大量问题而使用的框架技术。我们并不是编写了一个小框架，一旦发现一些东西就整理出一个模式，也不是我们想当然的认为它是个模式就把它整理为模式。我们之所以认定它们成为模式，是因为我们不断的发现它。它们被识别、精化，并应用到整个团队之中，包括技术专家和领域专家。最后，大量的软件开发人员通过构建他们自己的灵活的应用也验证了这些模式，其中的大部分应用都是在 IBM 的 SanFrancisco 框架的基础上为更大范围的销售或特定的业务需要进行的二次开发。

如何阅读本书

所有的读者都应当阅读第 1 章简介和第 2 章案例研究。简介通过描述框架是什么，如何开发和使用框架，为你提供了一个了解模式上下文环境的基础。它展示了模式是如何相互配合的，使你能够有兴趣深入了解模式。案例研究提供了一个通用领域以及领域中的词汇，这样我们就可以提出领域中的问题，并引发对模式的思考，

[1] 更多的信息请访问 http://java.sun.com/products/ejb。

以及想出解决问题的方法。

在结束了简介和案例研究章节的阅读之后，剩余的章节你可以以任意的顺序进行阅读。管理者和开发人员希望了解模式的大概，他们可以阅读所有模式的意图、问题、解决方法和适用性部分。剩余的部分可以等到以后需要时再详细阅读，例如第一次遇到似乎可以应用这些模式的情形。

根据模式的主要用途，我们把相关的模式组织成如下几个章节，包括：

- 第 3 章讨论了在整个框架开发过程中都适用的模式。
- 第 4 章讨论了识别和捕获需求的相关模式。
- 第 5 章讨论了分析阶段的模式。
- 第 6 章讨论了设计阶段的模式。
- 第 7 章讨论了框架的和文档相关的模式，对于框架开发来说，广泛的文档是非常重要的。
- 第 8 章讨论了框架（以及通用的面向对象软件）开发的组织特征，包括了属于不同人群的领域和技术专家的相关模式。
- 第 9 章讨论和使用框架相关的模式。

此外，本书还包括两个附录。附录 A 描述了组件和框架之间相互补充的关系。附录 B 描述了 IBM SanFrancisco 项目中用于业务应用框架开发的开发过程。

致　谢

没有大家共同的努力，该书将无法面世。IBM SanFrancisco 框架的成功是团队中众人智慧的结晶。每一位成员都以自己的方式贡献力量，形成了这些规则。我们感谢团队中的每一个人——在我们曾经工作过的团队中，你们是最出色的。

Brent 感谢他的妻子 Lisa，感谢她对自己写作爱好的容忍，并保证近期不再签任何图书出版合同。

Jim 感谢他的家庭对他花费大量时间用于本书写作上的支持。

感谢以下审阅者：Michele Chilanti，Will Traz，Simon Reason，Palle Nowack，Marcus Fontoura，Roger Snowden 帮助作者专注于本书的内容并和文字的推敲。此外还感谢 Addison-Wesley 出版社的同仁，特别是 Paul Beeker 和 Marcy Barnes。最后，特别感谢 Chrysta Meadowbrooke 为本书所做的文字编辑工作。

James Carey

Brent Carlson

目 录

第 1 章　简　介

在那些熟悉面向对象的开发者的眼中，面向对象框架（framework）的开发既熟悉又陌生。说它熟悉是因为框架的开发需要有熟练的面向对象开发经验，说它陌生则是因为开发者同时面临着新鲜的、令人惊奇的挑战。必须认识到，之前的开发工作中，你只需要交付可执行代码和一定数量的文档，而面向对象框架的工作可不仅如此。这就成为面向对象框架开发中最大的一项挑战。

1.1　什么是框架

在设计模式中，Gamma 等人为框架给出了一个定义：“框架就是一组协同工作的类，它们为特定类型的软件构筑了一个可重用的设计”[Gamma 94，p.26]。我们把这个定义稍微做一些扩展：框架是一组相互协作的组件（component）的集合，能够处理一个或多个问题域（domain）中的一系列问题。以上两种定义的本质是相同的，它们都包含了以下这些关键特征：

- 框架包括一系列的抽象体（abstraction）——类或组件。
- 这些抽象体相互协作，以完成任务。框架为一组相互独立的抽象体定义它们的协作方式，这是解决方案中可重用的核心（设计和实现）。
- 抽象体以及它们的行为方式都是可重用的。框架并不等同于类库（class library），类库的目的是提供一组到处都通用的类和函数。而框架提供的重用级别要高于类库，它更关注通用的过程（process）和功能（function）。实际上，框架通常可以在类库的基础上进行开发。此外，虽然框架非常依赖于设计模式，但它不仅仅是一些模式的简单叠加。框架是设计和实现的