

<<分析模式>>

图书基本信息

书名：<<分析模式>>

13位ISBN编号：9787111305309

10位ISBN编号：7111305302

出版时间：2010年07月

出版时间：机械工业出版社

作者：Martin Fowler

页数：319

译者：樊东平,张路

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>

<<分析模式>>

前言

不久前，还没有任何有关面向对象分析和设计的书籍。

而现在，却有如此之多的书籍，以致于任何一个专业人员都无法全部涉猎。

其中大部分书籍都专注于：传授一种图符表示法，提出一个简单的建模过程，并用几个简单的示例来加以说明。

本书是一本与它们完全不同的书。

它并不把重点放在过程——即如何建模，而是把重点放在过程的结果——即模型本身。

我是一个信息系统对象建模方面的顾问。

客户常常聘请我训练他们的员工如何建模和为他们的项目提供指导。

我的大部分技能来自对建模技术以及如何运用这些技术的了解。

然而，更重要的是我的实际经验，这些经验是在建造许多模型和经常分析重复出现问题的过程中积累起来的。

我经常发现项目在很多方面会遇到以前我曾面对的同样问题。

这些经验使得我可以重用以前所建造的模型，我只需要对这些模型加以改进，使之适应新的需求。

在过去的几年里，越来越多的人已经意识到这一现象，并且认识到那些通常介绍方法论的书籍虽然很有价值，但都只提出了学习过程的第一步，而这个学习过程还必须捕获要被建模的实际事物本身。

这种认识逐渐发展成为“模式”运动，在这一运动中汇集了各种各样的人，他们有着不同的兴趣和观点，却抱着共同的目标，即传播有用的软件系统模式。

由于这个模式群体构成的多样性，我们很难给“模式”一个准确的定义。

我们中的所有人都相信，一旦我们看到一个模式，就能辨别出它；我们认为我们在大多数情况下是一致的，但我们无法给出一个简单的定义。

我对模式的定义是：模式是一种问题解决思路，它已经适用于一个实践环境，并且也可能适用于其它环境。

我喜欢给出一个宽松的定义，因为我希望能尽可能地接近模式研究的初衷，而不需要增加太多限制性的内容。

模式可以有多种形式，而每一种形式增加了对于该模式有用的特性（1.2节讨论有关模式研究的现状以及本书所处的地位）。

本书讨论的是分析方面的模式，这些模式反映的是业务过程的概念架构，而不是实际的软件实现。

绝大部分章节讨论不同业务领域的模式。

这些模式很难按照传统的行业（如制造、金融、医疗保健等）进行分类，因为它们通常可用在多个领域。

这些模式非常重要，因为它们可以帮助我们了解人们对世界的认识。

基于这样的认识去设计计算机系统并确实去改变这种认识是非常有价值的，而认识中需要改变的地方正是需要进行业务过程重组（BPR）的地方。

然而，概念模式不可能孤立存在。

对于软件工程师来讲，只有在他们明白如何实现概念模型时，这些概念模型才有用。

本书介绍了一些可用于将概念模型转化成软件实现的模式，并且讨论了在一个大型信息系统中这些软件实现是如何适应系统构架的，另外还讨论了使用这些模式的具体实现技巧。

我写本书是因为它也正是我在开始时想要阅读的书。

建模人员会从本书中找到可以帮助他们如何在新领域中大展拳脚的基本思路。

这些模式包括：有用的模型、设计背后的论证以及适用范围。

拥有这些信息，建模人员就可以为特定的问题改造现有的模型。

<<分析模式>>

内容概要

《分析模式可复用的对象模型》的作者Martin Fowler是国际著名的OO专家，敏捷开发方法的创始人之一，现为ThoughtWorks公司的首席科学家，《分析模式可复用的对象模型》是作者的代表作之一，深受业界专业人士和广大读者的好评，经久不衰。

《分析模式可复用的对象模型》讲述各种分析模式（即来自概念性业务模型的模式）和支持模式（即讲述如何使用分析模式的辅助性模式），把论述重点放在介绍面向对象分析和设计的最终结果——即模型本身。

作者透过平实朴素的语言，将自己丰富的对象建模经验与读者分享，使读者可以马上采纳这些经验性模式。

《分析模式可复用的对象模型》适合的读者范围非常广：面向对象的计算机分析人员和设计人员（尤其是那些参与系统分析的人员）、数据建模人员、编程人员以及专业的软件工程师都可以从《分析模式可复用的对象模型》中获得宝贵的知识和经验。

<<分析模式>>

作者简介

作者：（英国）福勒（Martin Fowler）译者：樊东平 张路 等福勒（Martin Fowler），在面向对象分析设计、UML、模式、软件开发方法学、XP、重构等方面，都是世界顶级的专家，现为Thought Works公司的首席科学家。

Thoughtl Works是一家从事企业应用开发和集成的公司。

早在20世纪80年代，Fowler就是使用对象技术构建多层企业应用的倡导者，他著有几本经典书籍：《分析模式》、《UML精粹》和《重构》等。

<<分析模式>>

书籍目录

Ralph Johnson序 Ward Cunningham序 前言 第1章 绪论 1.1 概念模型 1.2 模式世界 4 1.2.1 Christopher Alexander 5 1.2.2 描述格式 5 1.2.3 关于模式的抽象程度 6 1.3 本书中的模式 7 1.3.1 建模实例 8 1.3.2 模式的来源 8 1.3.3 跨领域的模式 9 1.4 概念模型与业务过程重组 9 1.5 模式与框架 10 1.6 本书的使用 11 第一部分 分析模式 第2章 责任模式 17 2.1 团体 18 2.2 组织层次 19 2.3 组织结构 21 2.4 责任 22 2.5 责任知识级 24 2.6 团体类型泛化 26 2.7 层次型责任 27 2.8 操作范围 29 2.9 职位 31 第3章 观察和测量模式 33 3.1 数量 34 3.2 转换率 36 3.3 复合单位 37 3.4 测量 38 3.5 观察 40 3.6 观察概念的子类型化 43 3.7 观察方案 44 3.8 双时间记录 44 3.9 被否决的观察 45 3.10 临床观察、假设与推理 45 3.11 关联观察 46 3.12 观察过程 48 第4章 针对公司财务的观察模式 52 4.1 企业片断 53 4.1.1 定义维度 57 4.1.2 维度的属性以及企业片断 59 4.2 测量方案 60 4.2.1 保持计算的有效性 61 4.2.2 比较和因果测量方案 62 4.2.3 状态类型：定义计划的和实际的状态 63 4.2.4 构造测量 66 4.2.5 维度合并 66 4.3 范围 69 4.4 带范围的现象 70 4.4.1 带范围属性的现象 71 4.4.2 范围函数 73 4.5 使用最终框架 75 第5章 引用对象 77 5.1 名称 77 5.2 标识方案 79 5.3 对象合并 81 5.3.1 复制并替换 82 5.3.2 替代 82 5.3.3 本质/表象 83 5.4 对象等价 83 第6章 库存与账务 85 6.1 账目 87 6.2 事务 88 6.3 汇总账目 90 6.4 备注账目 92 6.5 记入规则 93 6.5.1 可逆性 94 6.5.2 不使用事务 94 6.6 个体实例方法 95 6.6.1 使用 singleton 类实现 95 6.6.2 使用策略模式实现 96 6.6.3 使用内部 case 语句实现 97 6.6.4 使用参数化方法实现 98 6.6.5 使用解释器实现 98 6.6.6 实现方式的选择 99 6.7 记入规则的执行 99 6.7.1 急切触发 99 6.7.2 基于账目的触发 101 6.7.3 基于记入规则的触发 102 6.7.4 向后链式触发 102 6.7.5 触发手段的比较 102 6.8 多个账目的记入规则 103 6.9 选择条目 106 6.10 账务实践 107 6.11 条目来源 109 6.12 结算单和所得计算书 110 6.13 对应账目 111 6.14 专门化的账目模型 112 6.15 登记条目到多个账目 113 6.15.1 使用备注账目 116 6.15.2 派生账目 116 进一步阅读 118 第7章 使用财务模型 119 7.1 结构模型 120 7.2 结构的实现 122 7.3 设置新的电话服务 124 7.4 建立通话 126 7.5 实现基于账目的触发 127 7.6 把电话分成白天和夜晚两类 128 7.7 按时间收费 130 7.8 计算税款 133 7.9 结论 134 7.9.1 记入规则的结构 134 7.9.2 什么时候不能使用框架 136 7.9.3 账务实践图 137 第8章 计划 139 8.1 提议和执行的动作 140 8.2 完成和放弃的动作 141 8.3 挂起 142 8.4 计划 143 8.5 方案 146 8.6 资源分配 149 8.7 输出和启动函数 153 第9章 交易 156 9.1 合同 156 9.2 合同夹 160 9.3 报价 165 9.4 场景 168 第10章 派生合同 176 10.1 期货合同 177 10.2 期权 179 10.2.1 多头、空头、看涨和看跌：体现一种谋略的词汇 181 10.2.2 子类型化或者非子类型化 182 10.3 产品 184 10.4 子类型状态机 188 10.4.1 确保状态图的一致 190 10.4.2 一致性的使用问题 192 10.5 并行的应用和领域层次结构 194 10.5.1 应用外观的类型检查 195 10.5.2 给超类型一个包装性接口 196 10.5.3 使用一个运行时属性 196 10.5.4 使应用外观对领域模型可见 198 10.5.5 使用异常处理 199 第11章 交易包 201 11.1 对一个包的多重访问级别 201 11.2 相互可见性 205 11.3 包的子类型化 208 11.4 结论 209 第二部分 支持模式 第12章 信息系统的分层构架 213 12.1 两层构架 214 12.2 三层构架 215 12.3 表示层和应用逻辑层 218 12.3.1 表示层/应用逻辑层分离的优点 222 12.3.2 在客户/服务器环境中伸展外观 222 12.4 数据库交互 224 12.4.1 把领域层连接到数据源 224 12.4.2 数据库接口层 225 12.5 结论 227 第13章 应用外观 229 13.1 一个医疗保健示例 229 13.2 外观的内容 231 13.2.1 方法的类型 232 13.2.2 样本方法 233 13.3 公共方法 234 13.4 操作 235 13.5 类型转换 236 13.6 多重外观 237 第14章 类型模型的模式—设计模板 240 14.1 实现关联 242 14.1.1 双向关联和单向关联 243 14.1.2 关联的接口 243 14.1.3 基础类型 245 14.1.4 实现一个单向关联 246 14.1.5 在两个方向上都使用指针的双向实现 246 14.1.6 在一个方向上使用指针的双向实现 247 14.1.7 使用关联对象的双向实现 248 14.1.8 双向实现的比较 248 14.1.9 派生映射 249 14.1.10 非集合映射 249 14.2 实现泛化 249 14.2.1 用继承实现 249 14.2.2 用多重继承组合类实现 250 14.2.3 用标志实现 250 14.2.4 用委托给一个隐藏类来实现 251 14.2.5 通过创建一个替换来实现 253 14.2.6 泛化的接口 254 14.2.7 实现 hasType 操作 255 14.3 对象创建 255 14.3.1 创建的接口 256 14.3.2 创建的实现 256 14.4 对象析构 256 14.4.1 析构的接口 257 14.4.2 析构的实现 257 14.5 入口点 258 14.5.1 查找对象的接口 259 14.5.2 查找操作的实现 260 14.5.3 使用类或者登记表对象 260 14.6 实现约束 260 14.7 其它技术的设计模板 261 第15章 关联模式 263 15.1 关联类型 264 15.2 带键值的映射 266 15.3 历史映射 268 第16章 后记 273 第三部分 附录 附录A 技术和符号 277 附录B 模式列表 293 索引 301

<<分析模式>>

章节摘录

插图：如果一个模型有多个隶属层次关系，我们可以用一种类型化的关系（如图2.6所示）来表示。我们把层次关联关系转化成一种类型，通过使用组织结构类型的不同实例来区分不同的层次关系。这样就能用组织结构的两个实例（销售组织和服务组织）来处理上一节的场景（双层次关系）。新产生的层次关系可以通过简单地增加新的组织结构类型的方式加以处理。

显然，这种抽象方式使我们能够在复杂性适度增加的情况下的增加更多的系统柔性。

对于双层次关系，这样去做并不值得，但对于多层次关系，就很有必要。

另外请注意，组织结构有时间周期；这使我们可以有效地记录组织结构的周期性变化。

进而要注意的是，我并没有把组织结构类型看成是一种属性——类型属性是一个很重要的概念，我们将在后面具体谈到。

例：为波士顿的2176大容量卡布奇诺咖啡机而设立的服务小组向波士顿的销售办事处负责。

我们可以将其刻画成这样一个组织结构模型：父节点是波士顿的销售办事处，子节点是波士顿的2176服务小组，组织结构类型叫做产品线管理。

例：为波士顿的2176大容量卡布奇诺咖啡机而设立的服务小组向产品支持结构中的2170产品系列服务中心负责。

我们可以把它看成是一个单独的组织结构，它的父节点是2170产品系列服务中心，而子节点是波士顿的2176服务小组。

组织结构类型叫做产品支持。

要简化对象结构，应把重点放在约束规则上。

这些规则的具体形式可以是：“对于一个组织结构，如果其类型是销售组织并且其子节点是一个部门的话，那么其父节点必须是一个区域子公司”。

请注意，约束规则被表示成指向组织结构的属性，其暗含着约束规则针对该组织结构。

然而，这也意味着当通过增加新的组织结构类型的方式来扩展系统时，会改变该组织结构中的约束规则。

而且，随着组织结构类型数量的增加，这些规则将变得难以处理。

可以把约束规则放到组织结构类型中（如图2-7所示）。

针对特定的组织结构类型的所有规则被集中到一个地方，这样便于增加新的组织结构类型。

然而，如果我们很少改变组织结构类型而是经常增加新的组织子类型，图2-7就难以处理了。

在这种情况下，组织子类型的每次增加都会导致约束规则的改变。

更好的方法是让约束规则跟随组织子类型。

概括起来，我们的目标是尽量减小模型的变化。

我们应该按照这种方式，在不影响模型的其它部分的前提下，把约束规则放在最容易发生变化的地方

。

<<分析模式>>

媒体关注与评论

“ 本书是对不断发展的模式文献的一个重要贡献。

它捕捉来自不同领域的深奥的对象建模专业知识，形成一个模式目录。

这些领域模式将有助于你解决不同领域中具有挑战性的建模问题。

” ——Erich Gamma “ Martin Fowler为我们给出答案，而不仅仅是一个可以找到这些答案的过程

。

在本书中，透过作者平实朴素的语言，你将找到自己下一个业务对象模型的重要内容。

” ——Ward Cunningham “ 就像‘四人帮’在他们的经典著作《设计模式》中总结出了通用的设计模式，Martin Fowler在这本让人期待已久的书中为我们总结出应用领域的诸多模式。

本书是从事面向对象业务建模和业务过程重组工作的所有分析人员和设计人员的必备之书。

” ——Donald G. Firesmith

<<分析模式>>

编辑推荐

《分析模式可复用的对象模型》：Martin Fowler认识到面向对象研究团体需要一本超越传统方法学著作所包含的工具和技术的书，因此撰写了《分析模式可复用的对象模型》，重点介绍面向对象分析和设计的最终结果——模型本身。

他将自己丰富的对象建模专业经验与读者分享，着眼于找出重复问题并把这些问题的转换为可复用的模型。

《分析模式可复用的对象模型》提供一个模式目录，涉及交易、测量、财务和组织内部关系等广泛领域。

鉴于概念模式不能孤立存在，Martin Fowler还提出一系列“支持模式”，这些支持模式讨论如何将概念模式转变为适合大型信息系统构架的软件。

在介绍每种模式时，都讲述设计背后的缘由以及使用这种模式的规则。

书中的示例包含有用模型的使用细节并进一步探讨了将会改进分析、建模和实现的复用技巧。

<<分析模式>>

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>