

函数响应式领域建模



[函数响应式领域建模_下载链接1](#)

著者:【美】 Debasish Ghosh

出版者:电子工业出版社

出版时间:2018-1

装帧:平装

isbn:9787121323928

传统的分布式应用不会切入微服务、快速数据及传感器网络的响应式世界。为了捕获这些应用的动态联系及依赖，我们需要使用另外一种方式来进行领域建模。由纯函数构成的领域模型是以一种更加自然的方式来反映一个响应式系统内的处理流程，同时它也直

接映射到了相应的技术和模式，比如Akka、CQRS以及事件溯源。《函数响应式领域建模》讲述了响应式系统中建立领域模型所需要的通用且可重用的技巧——首先介绍了函数式编程和响应式架构的相关概念，然后逐步地在领域建模中引入这些新的方法，同时《函数响应式领域建模》提供了大量的案例，当在项目中应用这些概念时，可作为参考。

作者介绍:

译者简介

李源，曾在华为技术有限公司工作8年，经历过开发、SE、PM和PQA等多个岗位，目前在途牛旅游网担任研发总经理一职，是美国质量协会（ASQ）注册质量工程师（CQE）；译者有丰富的开发、架构设计及研发管理经验，先后负责过多个大型项目的方案设计和系统规划，对于C++、Java以及设计模式等领域都有比较深入的研究；曾翻译《Java 性能调优指南》一书。

目录:

1. 函数式领域建模：介绍.....	1
1.1 什么是领域模型	2
1.2 领域驱动设计介绍	4
1.2.1 边界上下文	4
1.2.2 领域模型元素	5
1.2.3 领域对象的生命周期	8
1.2.4 通用语言	13
1.3 函数化思想	14
1.3.1 哈，纯粹的乐趣	17
1.3.2 纯函数组合	21
1.4 管理副作用	26
1.5 纯模型元素的优点	28
1.6 响应式领域模型	31
1.6.1 响应式模型的3+1 视图	31
1.6.2 揭穿“我的模型不能失败”的神话	32
1.6.3 伸缩性与消息驱动	34
1.7 事件驱动编程	35
1.7.1 事件与命令	37
1.7.2 领域事件	38
1.8 函数式遇上响应式	40
1.9 总结	41
2 Scala 与函数式领域模型.....	42
2.1 为什么是Scala	43
2.2 静态类型与富领域模型	45
2.3 领域行为的纯函数	47
2.3.1 回顾抽象的纯粹性	50
2.3.2 引用透明的其他好处	53
2.4 代数数据类型与不变性	53
2.4.1 基础：和类型与乘积类型	53
2.4.2 模型中的ADT 结构数据	56
2.4.3 ADT 与模式匹配	56
2.4.4 ADT 鼓励不变性	58
2.5 局部用函数，全局用OO.....	59
2.5.1 Scala 中的模块	60
2.6 用Scala 使模型具备响应性	64
2.6.1 管理作用	65

2.6.2 管理失败	65
2.6.3 管理延迟	67
2.7 总结	69
3 设计函数式领域模型	70
3.1 API 设计的代数	71
3.1.1 为什么是代数方法	72
3.2 为领域服务定义代数	72
3.2.1 赋值抽象	73
3.2.2 组合抽象	74
3.2.3 类型的最终代数	76
3.2.4 代数法则	77
3.2.5 代数解释程序	79
3.3 领域模型生命周期中的模式	80
3.3.1 工厂——对象从何处来	82
3.3.2 智能构造器	82
3.3.3 通过更有表现力的类型进一步提升智能	84
3.3.4 用代数数据类型聚合	86
3.3.5 用透镜更新聚合功能	88
3.3.6 仓储与解耦的永恒艺术	94
3.3.7 高效地使用生命周期模式——结论	101
3.4 总结	102
4 领域模型的函数式模式	103
4.1 模式——代数、函数、类型的聚合	104
4.1.1 领域模型中的挖掘模式	106
4.1.2 用函数式模式使领域模型参数化	107
4.2 强类型函数式编程中计算的基本模式	112
4.2.1 函子——建立模式	112
4.2.2 加强版函子模式	114
4.2.3 单子作用——applicative 模式的变体	121
4.3 如何用模式对领域模型进行塑形	130
4.4 用代数、类型和模式演进API	134
4.4.1 代数——第一稿	136
4.4.2 改进代数	137
4.4.3 最终组合——采用类型	138
4.5 用模式和类型增强领域的不变性	139
4.5.1 贷款处理模型	139
4.5.2 使非法状态不可表示	141
4.6 总结	142
5 领域模型的模块化	144
5.1 将领域模型模块化	145
5.2 模块化的领域模型——案例学习	146
5.2.1 模块的解剖	147
5.2.2 模块的构成	154
5.2.3 模块的物理组织	155
5.2.4 模块鼓励组合	156
5.2.5 领域模型中的模块化——结论	157
5.3 类型类模式——模块化的多态行为	157
5.4 边界上下文的聚合模块	160
5.4.1 模块与边界上下文	161
5.4.2 边界上下文间的通信	162
5.5 模块化的另一个模式——free monad	163
5.5.1 账户存储	163
5.5.2 使它免费	165
5.5.3 账户存储——free monad	167

5.5.4 free monad 解释程序	169
5.5.5 free monad——重点回顾	172
5.6 总结	173
6 响应式模型	174
6.1 响应式领域模型	175
6.2 使用future 的非阻塞API 设计	177
6.2.1 异步作为堆叠作用	178
6.2.2 基于monad 转换器的实现	181
6.2.3 用并行存取降低延迟——一种响应式模式	183
6.2.4 使用scalaz.concurrent.Task 作为响应式构造	187
6.3 明确的异步消息传递	189
6.4 流模式	191
6.4.1 一个案例	191
6.4.2 领域管道图	195
6.4.3 后端压力处理	197
6.5 actor 模型	198
6.5.1 领域模型与actor	199
6.6 总结	203
7 响应式流建模	205
7.1 响应式流模型	206
7.2 何时使用流模型	207
7.3 领域用例	208
7.4 基于流的领域交互	208
7.5 实现：前台	210
7.6 实现：后台	211
7.7 流模型的主要结论	214
7.8 使模型具有弹性	215
7.8.1 使用Akka Streams 监管	216
7.8.2 冗余集群	217
7.8.3 数据的持久化	217
7.9 基于流的领域模型与响应式原则	219
7.10 总结	220
8 响应式持久化与事件溯源	221
8.1 领域模型的持久化	222
8.2 关注点分离	224
8.2.1 持久化的读/写模型	225
8.2.2 命令查询责任分离	226
8.3 事件溯源	228
8.3.1 事件溯源领域模型中的命令和事件	229
8.3.2 实现CQRS 和事件溯源	231
8.4 实现事件溯源的领域模型（函数式）	232
8.4.1 作为头等实体的事件	233
8.4.2 命令是事件上的free monad	235
8.4.3 解释程序——隐藏所有有趣的东西	237
8.4.4 投影——读取端模型	242
8.4.5 事件存储	243
8.4.6 分布式CQRS——一个短信	243
8.4.7 实现的总结	244
8.5 其他持久化模型	245
8.5.1 将聚合作为ADT 映射到关系型表	245
8.5.2 操作数据（函数式）	247
8.5.3 到Akka Streams 管道的响应式获取	248
8.6 总结	249
9 测试领域模型	250

9.1 测试领域模型概述	251
9.2 设计可测试的领域模型	252
9.2.1 解耦副作用	253
9.2.2 为领域函数提供自定义解释程序	254
9.2.3 实现参数化与测试	255
9.3 基于xUnit 的测试	256
9.4 回顾模型的代数	257
9.5 基于属性的测试	258
9.5.1 建模属性	258
9.5.2 验证领域模型中的属性	259
9.5.3 数据生成器	264
9.5.4 是否比基于xUnit 的测试更好	266
9.6 总结	267
10 核心思想与原则.....	268
10.1 回顾	268
10.2 函数式领域建模的核心原则	269
10.2.1 表达式思想	269
10.2.2 早抽象, 晚赋值	270
10.2.3 使用合适的抽象	270
10.2.4 发布要做什么, 在组合器中隐藏如何做	270
10.2.5 从实现中解耦代数	271
10.2.6 隔离边界上下文	271
10.2.7 偏向future 而不是actor	271
10.3 展望未来	272
• • • • • (收起)	

[函数响应式领域建模_下载链接1](#)

标签

函数式编程

领域驱动设计

响应式编程

编程

函数式

scala

计算机科学

计算机

评论

这真是一本奇书，用函数式编程来实现以前面向对象编程拿手的领域功能开发。书中运用大量函数式编程技巧解决现实问题，但是没有详细介绍函数式编程基础知识，为了读懂书中的例子我又找其他书重新学习了一遍函数式编程，即使这样还是很多地方没有看懂。频繁使用Scala举例对其他语言开发人员也不够友好。

(还好学过一点 scala，所以 scala 相关的概念并没有 block 读书的过程。 不错的书

对函数式编程了解不多，领域驱动设计理解还比较深入，响应式原则非常熟悉，三者结合在一起感觉还是有所欠缺。

奇书，超级赞！

本书利用领域建模为试验田，讨论了函数式编程的一种实践，我觉得对在实际场景中理解函数式编程思想的使用非常有帮助，帮我理解了之前死活没感觉的Monad。至于响应式/并发的部分，则差强人意，这部分可以阅读《反应式设计模式》来理解。对于领域驱动设计的部分，也说得不是特别清楚，这部分可以读一遍《实现领域驱动设计》来理解。

前半段地DDD结合函数式编程地理念，通过函数式通用模式体现出来了。但后半段有些过于冗长了。另外感觉作者对函数式本身并不了解，一些函数式专属词汇翻译的并不好。

理论不少，实践也有，scala大法保平安

[函数响应式领域建模 下载链接1](#)

书评

这本书名为 Functional And Reactive Domain Modeling，主要是将 DDD (Domain Driven Design)、Functional Programming 和 Reactive Pattern 整合在一起应用在分布式环境中，这个话题本身是非常有趣的，尤其是当下 IoT 行业正在成形的当下，我们或许需要一种能够应对真实世界中...

说来惭愧，这本书是凑单的时候顺手买的，正所谓“买书如山倒，读书如抽丝”，买了以后就一直放着吃灰。上周末随手一翻，我去，还挺有趣！于是用两个周末 + 一周的空闲时间把这本书读完了。本书主要涉及三个方面的内容：* 领域驱动设计 * 函数式编程 * 反应式设计 并以个人银...

[函数响应式领域建模 下载链接1](#)