TURING 图灵程序设计丛书

MANNING

DSLs in Action

领域专用语言实战

[美] Debasish Ghosh 著

郭晓刚 译

著名博客“Ruminations of a Programmer”作者

ACM高级会员20余年经验总结

多位业内大牛鼎力推荐

全面涵盖5种JVM语言

真正讲透DSL设计与实现

人民邮电出版社

POSTS & TELECOM PRESS

TURING 图灵程序设计丛书

DSLs in Action

领域专用语言实战

[美] Debasish Ghosh 著

郭晓刚 译

人民邮电出版社

北 京

图书在版编目（CIP）数据

领域专用语言实战 / (美) 戈施 (Ghosh,D.) 著 ; 郭晓刚译. -- 北京 : 人民邮电出版社, 2013.11
(图灵程序设计丛书)
书名原文: DSLs in action
ISBN 978-7-115-33174-8

Ⅰ. ①领… Ⅱ. ①戈… ②郭… Ⅲ. ①程序语言 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2013)第234324号

内 容 提 要

DSL（领域专用语言）的要旨在于沟通。精心设计的DSL可以以一种从外观到内在都极为自然的方式，传达出其所表示领域的本质和真意，帮助消除业务与技术的隔阂，促进项目干系人与程序员的沟通。

《领域专用语言实战》不仅介绍如何使用DSL解决问题，还会使用Ruby、Groovy、Scala、Clojure等现代语言阐述DSL的设计与实现，针对这些语言所代表的不同编程范式深入讨论其在DSL设计上的优劣。本书共分三部分。第一部分定位DSL驱动开发环境，寻找其在应用程序架构中的用武之地，帮助程序员或架构师了解如何调整现有开发工具和技术，使之适应DSL驱动的新范式。第二部分带你设计优秀的语义模型，使之成为上层语言抽象的有力后盾。该部分主要指导开发人员按照优秀抽象的设计原则搭建领域模型，由浅入深讲解了DSL实现技术，如元编程、解析器组合子，以及ANTLR、Xtext等开发框架。第三部分主要展望未来趋势，重点讨论解析器组合子和DSL工作台技术的发展前景。

本书适合开发人员、架构师、领域用户学习参考。

◆ 著　[美] Debasish Ghosh
序　Jonas Bonér
译　郭晓刚
责任编辑　毛倩倩
执行编辑　李　静　张　庆
责任印制　焦志炜

◆ 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　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京鑫正大印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：19.5
字数：479千字　2013年11月第1版
印数：1－4 000册　2013年11月北京第1次印刷
著作权合同登记号　图字：01-2011-2958号

定价：69.00元

读者服务热线：(010)51095186转604　印装质量热线：(010)67129223
反盗版热线：(010)67171154
广告经营许可证：京崇工商广字第0021号

版权声明

序

我很喜欢摆弄编译器，只要是亲手打造自己的语言，无论动手还是动脑都是一种享受。编程语言，尤其是DSL（Domain Specific Language，领域专用语言），能激起我极大的热情。

DSL这一概念并不是最近才发明的，Lisp开发者们早就在开发和使用所谓的“小语言”了。不过近年来，DSL确实在整个行业范围内被更广泛地使用和接受，相关工具和技术也日渐成熟。如果开发者想探索语言设计这一精彩世界，可以说现在的条件是前所未有的。

如同大多数语言，DSL的要旨在于沟通。精心设计的DSL可以以一种从外观到内在都极为自然的方式，传达出其所表示领域的本质和真意。DSL能帮助消除业务与技术的隔阂，促进项目干系人与程序员的沟通。这种能力比以往任何时候都更重要，更值得我们去追求。

Debasish在Scala和开源社区里都是受人尊敬的专家。他的博客既给人以学识上的启发，又充满阅读乐趣，多年来我一直在关注。Debasish一年前开始为Akka项目贡献力量，我这个长年的读者因而与他有了进一步的接触。往来言行一下子就表露出来，他不仅是一位深刻的思考者，还是一位有行动力的实干家。从那以后，与他讨论编程语言、设计成了我乐在其中的爱好。

这是一本令人激动的书。书中内容的涵盖面很广，而在此基础上又有相当的深度，除了带领读者穿梭于DSL发展的最前沿，它还将带领大家思考如何设计灵活而自然的DSL。此外，读者还将领略Scala、Groovy、Clojure、Ruby等各具特色的语言，掌握每一种语言解决问题的思路和手段。开卷有益，诚哉斯言。

Jonas Bonér
Akka项目、AOP框架AspectWerkz创建人
http://jonasboner.com

前　言

2001年春天，我供职的Anshinsoft公司（http://www.anshinsoft.com）开始涉足企业应用开发业务，客户是一家在亚太区数一数二的证券中介和资产管理企业。这段经历激起了我对一个专门的问题领域进行建模，然后将模型转换成软件的兴趣。于是我开始了一段考验毅力的学习旅程，仔细参详了Eric Evans的领域驱动设计著作（*Domain-Driven Design: Tackling Complexity in the Heart of Software*[①]），聆听了Josh Bloch关于如何设计优秀API的教诲（How to Design a Good API & Why it Matters，http://www.infoq.com/presentations/effective-api-design）以及Martin Fowler关于DSL的教义。

精心设计的DSL旨在向目标用户提供人性化的界面，而做到这一点的最佳途径是让编程模型使用领域专用语言来“说话”。我们一直以来总是把程序设计得像个“黑盒”，很少让业务人员得知其内部细节，这种做法可以休矣。经验告诉我，所有用户都希望查看一下你建模在代码里的业务规则，而不是白板上杂乱的框框和箭头。

嵌在代码里的规则要容易被用户理解，用户必须能看懂你使用的语言，这就是我从事十年领域建模的领悟。当规则可被理解的时候，DSL也就呼之欲出了。随之得到改善的不仅有开发团队和业务人员的沟通效率，还有软件面向用户的表达能力。

对于我们能否为用户提供表现力充沛的语法和语义，实现语言无论何时都是一个决定性的因素。有赖于当今生态环境的巨大发展，我们所设计的语言得以在表现力上有了长足进步。以鼓励开发者编写精炼而富有表现力的代码而论，Ruby、Groovy、Scala和Clojure是先行的表率。在这几种语言下的第一手编程经验让我感觉到，它们的语言风格和表达习惯远比大多数前代语言更适合领域建模。

写这样一本关于DSL的书是很大的挑战。我试图关注DSL的一切现实事物，所以从一开始就设定了具体的领域。当我们渐次展开论述，领域模型随着各种业务需求的累加而变得越来越复杂。这正好充分体现了DSL驱动的开发方式对问题域复杂度增长的适应能力。DSL方式并不是对API设计的颠覆，它只是鼓励你在API的设计思路上多考虑一个维度。请务必记住，你的用户才是DSL的使用者。凡事多从用户的角度去考虑，你一定会成功的！

---

① 中文版《领域驱动设计》，人民邮电出版社出版。——编者注

致　谢

首先，这可能有点儿不同寻常，我想感谢一下自己。我完全没料到自己能把注意力集中在一个兴趣点上这么久。编写本书的过程中，我真正意识到了毅力、决心和信念的价值。

感谢Anshinsoft的同事们创造了能够培育想法并让想法腾飞的工作环境。夜半之时的头脑风暴帮助我雕琢出了许多复杂的领域模型，也引发了我对DSL的热爱。

感谢我们的客户代表渡边桑（Tohru Watanabe先生），我从他那里学到了证券交易业务的领域模型。这本书里满是他多年来教给我的例子。

感谢以下审阅者帮助我提高书稿的质量：Sivakumar Thyagarajan、Darren Neimke、Philipp K. Janert、James Hatheway、Kenneth DeLong、Edmon Begolli、Celso Gonzalez、Jason Jung、Andrew Cooke、Boris Lenzinger、David Dossot、Federico Tomassetti、Greg Donald、John S. Griffin、Sumit Pal、Rick Wagner。特别感谢审阅者Guillaume Laforge和John Wilson指正Groovy DSL的编写细节，感谢Michael Fogus对第5章和第6章内容的建言，感谢Sven Efftinge对第7章中Xtext和外部DSL的意见。我还要感谢Franco Lombardo在本书付印前的紧迫时间里对文稿的最后技术审读。

Twitter网友在本书的写作过程中贡献了许多真知灼见，给了我无可估量的帮助和启发。

Manning出版社有一支优秀的团队，感谢他们的实心协助。项目编辑Cynthia Kane在文法和写作风格上不知疲倦地给出了指点，还站在读者的角度与我探讨了本书每一章的内容。如果读者觉得本书文字简单易懂，那要归功于Cynthia一遍又一遍的审读，是她敦促我一遍又一遍地修改行文。感谢Karen Tegtmeyer组织同行评议，感谢Maureen Spencer在本书撰写的全过程给予的帮助，感谢Joan Celmer在编辑过程中的积极响应，感谢Manning出版社对本书制作提供了大力支持的所有工作人员。我还要感谢出版人Marjan Bace对我的信任。

向我的夫人Mou致以特别的谢意，她在我写书的日子里一直激励我。这段漫长而辛劳的旅程因为她不同时期的鼓舞而有了非凡的意义和累累硕果。

关于本书

每一次我们在白板上设计领域模型，似乎总会在落实到代码的时候于纷杂中走了样。实现模型不管用哪种编程语言来表述，它都已经不是领域专家能理解的业务语言形式。白板上的模型是否精确反映了我们与领域用户商定的需求规格，这也无从让掌握领域规则的人员去验证。

对于这个症结，本书给出的解决之道是采用一种以DSL驱动的应用程序开发模型。假如我们围绕领域用户能够理解的语法和语义设计领域API，那么即使在应用程序代码的开发过程里，用户也能随时检查领域规则实现得是否正确。采用领域语言的代码更容易让人看懂，在这一点上，开发人员、维护人员、只懂业务不懂编程的领域专家都是受益者。

本书除了教你使用DSL来解决问题，还会教你实现DSL。在本书看来，DSL只是在语义模型外面包裹上薄薄一层以语言形态呈现的抽象。语义模型是把握领域核心结构的实现载体，语言层则使用领域用户的专门用语。

本书将使用Ruby、Groovy、Scala、Clojure等现代语言来讲授DSL的设计与实现，针对这些语言所代表的不同编程范式深入讨论它们在DSL设计上的长处和短处。读完本书，你将透彻理解一些必须掌握的概念，能够设计出用户理解且欣赏的优美的领域抽象。

读者对象

如果你希望自己设计的API其表现力既满足领域用户的需要，又能达到程序员同行的要求，那么这本书恰好适合你。如果你是一名领域用户，正期待着改善与开发团队的沟通效果，那么这本书恰好适合你。如果你是一名程序员，正为如何与领域用户核对业务规则的实现正确与否而苦恼，那么这本书同样适合你。

本书内容

图1、图2、图3除了勾画出了全书的组织脉络，对各章的内容也作了简略的阐述。本书分为三部分：

- 使用DSL；
- 实现DSL；
- DSL开发的未来趋势。

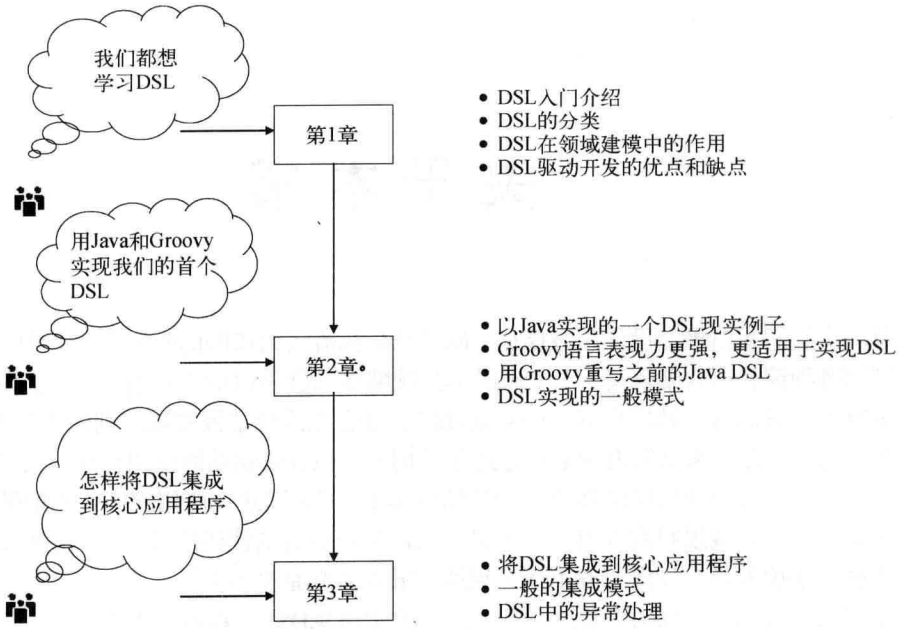

图1　第1章到第3章的学习历程

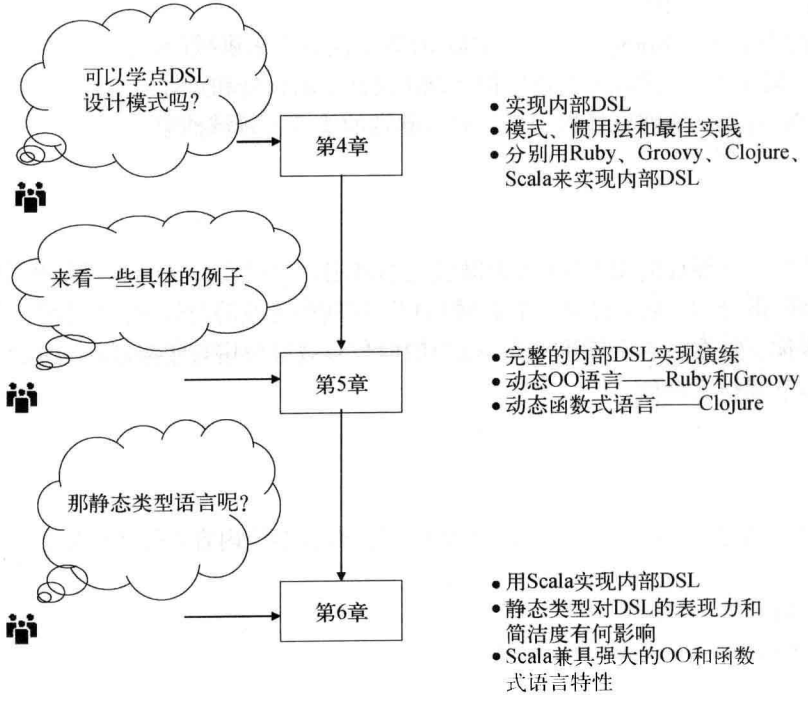

图2　第4章到第6章的学习历程

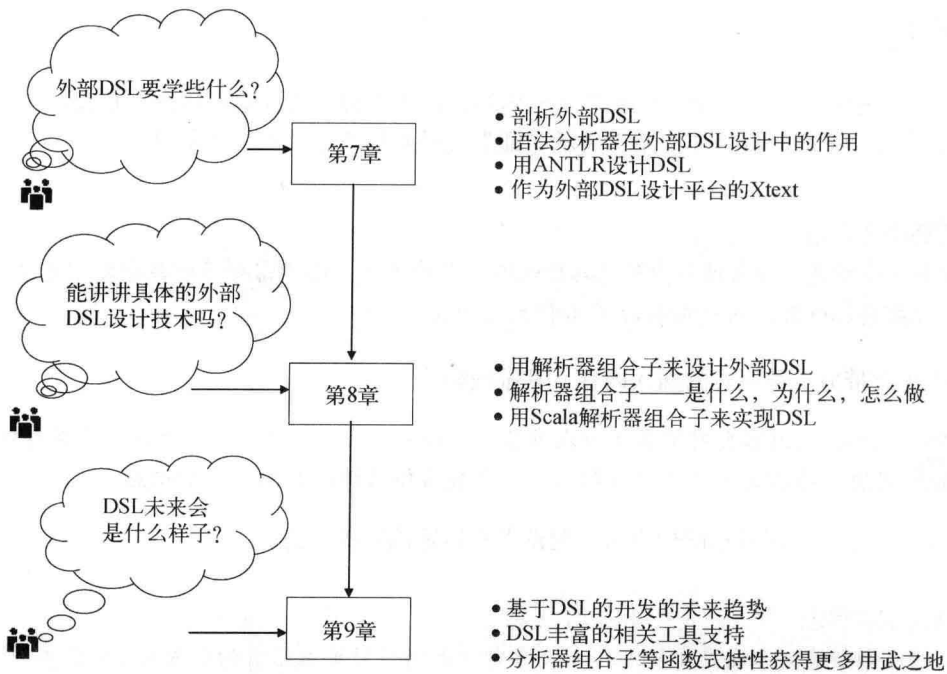

图3 第7章到第9章的学习历程

第一部分（第1章~第3章）作为总括，详细地阐述了DSL驱动开发环境的定位，帮助读者在自己的应用程序架构中找到它的用武之地。如果你是程序员或者架构师，这部分内容将协助你调整现有的开发工具和技术，使之适应DSL驱动的新范式。本书主要围绕各种JVM（Java虚拟机）语言展开论述。因此，Java程序员很快就能够从书中找到适合自身项目情况的DSL运用方式，在自己的Java项目内集成用表现力更佳的其他JVM语言开发出来的DSL。

DSL拥有各种贴近用户思维的语法结构，这些语言抽象有赖于语义模型在背后提供支撑。第二部分（第4章~第8章）探讨如何设计出优秀的语义模型，使之成为上层语言抽象的有力后盾。这个部分主要是给开发人员准备的，旨在指导开发人员按照优秀抽象的设计原则来搭建领域模型。从第4章到第8章，各章都含有丰富的DSL代码片段，实现语言包括Ruby、Groovy、Clojure、Scala等。如果正在或即将使用这些语言来实现DSL，那么你会发现这几章的内容特别实用。书中讲解了DSL的实现手法，而且将从最基本的技术入手，逐渐深入到高级技术，如元编程、解析器组合子，以及ANTLR、Xtext等开发框架。

第三部分（第9章）主要展望未来趋势，重点讨论解析器组合子和DSL工作台技术未来的发展。

本书面向真正的实践者。因此，虽然书中也含有理论知识，但只是作为帮助理解具体实现的铺垫而存在。我发自内心地相信这将是一本让奋战在开发第一线的实干家感觉有用的书。

排版约定

本书正文中穿插了不少插入内容和补充内容，用于提醒读者注意一些重要信息。

一般来说，下面的排版样式用于展示与证券交易及结算领域有关的信息。

金融中介系统

带有这个标志，即表示其中信息与DSL所属领域有关，读者需要了解其中知识才能理解上下文。此类内容一般是对一些特殊术语和概念的背景介绍。

书中还有带另一种标志的插入内容，其排版格式如下。

此类插入内容含有不属于所在章节讨论话题的知识。例如某种DSL设计的特殊惯用法、对前文讨论的重点归纳，或者我希望强调的其他重要知识点。

另外，我还用下面所示的标志来引起读者对特定内容的注意。

语言相关信息

看到这个标志，你就应该知道其中含有当前示例所用编程语言的小知识。你需要掌握这些特定的概念才能真正理解当前示例。

在这里，我提请大家注意不要轻易忽略这些带有不同标志的补充信息，它们都是可以帮助你透彻理解当前讨论内容的重要参考知识。

代码约定和下载

本书包含大量的DSL示例，其中不少例子的完成度很高，足以完整解释某一方面的领域规则实现。这些例子使用的编程语言有Java、Ruby、Groovy、Scala和Clojure。代码清单和正文中插入的代码片段都使用等宽字体，便于读者把它们和一般的文字区分开来。正文中出现的方法名、参数、对象属性、ANTLR和Xtext等脚本、XML元素和属性也都一律使用等宽字体呈现。

示例代码不一定总是短小的片段，有时候为了充分说明所涉领域的上下文和语义，也会占用较长的篇幅，保留较多的细节。书中的代码经常会为了适应书本的页面宽度，而在断行和缩进上做一些格式调整；偶尔还有调整不过来的情况，这时对于那些不得不折行的代码，我们会在折行的位置打上一个续行标记。

很多代码清单中会穿插一些标注，以便向读者提示重点。有时候标注还会带有数字编号，方便我们在后续的介绍中引用和参照。

书中用了多种编程语言来实现DSL，显然不可能所有的读者都熟悉其中所用的每一种语言。因此作为快速的参考，书后准备了每种语言的速查表格，见附录C到附录G。附录中的介绍只是

针对书中探讨DSL实现所用到的一些关键语言特性展开，并不完整全面，进一步的知识需要读者到表格后补充的参考资料中去寻找。

大多数时新的IDE都有能力支持开发者在同一项目中使用多种语言。如果读者不熟悉多语言开发环境，附录G是一个简单的入门指导。

书中所有示例的源代码都可以从Manning出版社网站下载[①]，地址为http://www.manning.com/DSLsinAction，配置构建环境和执行环境的相关指示也包含在内。阅读的时候在手边备一份源代码，这会对你很有帮助。

作者在线

本书有一个由Manning出版社运营的关联网络论坛，购买本书的读者具有免费访问论坛的权利。读者可以在上面发表评论、询问技术问题，并获得作者和其他用户的帮助。注册及使用论坛请访问http://www.manning.com/DSLsinAction。读者可从该页面了解论坛的注册和使用方法、论坛内提供的帮助、论坛守则等信息。

Manning出版社向读者承诺提供读者之间、读者与作者之间展开有意义对话的便利场所。作者只是志愿（且无偿）地参与论坛活动，因此Manning出版社不对作者参与论坛的程度做要求。我们建议读者尽量提出一些具有挑战性的问题，让作者有兴趣持续访问本论坛。在书籍在版期间，出版社网站将保证读者可以访问作者在线交流论坛及论坛上积累的讨论内容。

① 也可在图灵社区本书页面（http://www.ituring.com.cn/book/836）免费注册下载。——编者注

关于封面图片

本书封面图片为“来自克罗地亚斯拉沃尼亚地区奥西耶克城附近的久尔杰瓦茨村的男人”。这幅画是在克罗地亚历史名城斯普利特的民族博物馆一位热心馆员的帮助下取得的，来自该馆2003年重版的一本19世纪中期由Nikola Arsenovic编撰的克罗地亚传统服饰画集。斯普利特民族博物馆本身即坐落于中世纪的城市中心，也是罗马古迹的核心所在——建于公元304年前后的罗马帝国宫殿戴克里先宫遗址上。画集内收录了克罗地亚各地区人物形象的精细彩绘图样，并配有对服饰和生活习俗的文字说明。

久尔杰瓦茨村位于奥西耶克城附近，属于克罗地亚东部历史悠久的斯拉沃尼亚地区。斯拉沃尼亚的男人们传统上穿戴红色的帽子、白色衬衣、带刺绣图案的蓝色马甲和长裤，然后点缀上毛织或皮革的宽腰带和厚毛袜，最后罩一件棕色羊皮的短外套，也就是本书封面上的样子。

人们的衣着式样和生活习俗在最近的200年里发生了很大的变化。从前各地丰富多彩，极具地方特色的衣着和习俗已经消失殆尽。依现在的情况，甚至连不同洲的居民都趋于同化、难以区分了，相距数里的村落和市镇之间，就更不可能有什么差别了。也许，我们已经牺牲了文化的多样性来换取多姿多彩的个人生活——五光十色的、快节奏的科技生活。

Manning出版社特意从旧书和藏品里找回这些古老图样，让两个世纪以前丰富多彩的地方生活特色在图书封面上重焕光彩，以此来表达对计算机行业的创造精神和主动精神的赞美。

目　录