

领域分析综述

邬保明

(上海邦德职业技术学院 上海 200444)

摘要 领域分析是软件复用开发中的主要分析方法。探讨领域分析的脉络有利于对软件开发的技術及其发展趋势做出分析和预测,进而预测未来软件开发模式的发展趋势,并及时寻找新的软件分析方法。回顾了领域分析的出现及其不断发展变迁的过程,从中掌握领域分析的演化规律。通过对各个时期的领域分析概念的对比分析,找到领域分析的各自不同的特点和运用,总结领域分析的发展规律。最后展望了领域分析发展趋势并给出评论。

关键词 领域工程,组件开发,需求分析

中图法分类号 TP311.5 文献标识码 A

Survey of Domain Analysis

WU Bao-ming

(Shanghai Bangde College, Shanghai 200444, China)

Abstract Domain analysis is an important work in modern software analysis field used on reusability. Study on the evolution of domain analysis assists the decomposability and anticipation of software analysis and software development patterns, thereby aiding to forecast the perspective of future software development. This paper retrieved the emergence and transition history of domain analysis, and tried to find laws of evolution of domain analysis. By the way of comparison of domain analysis in different periods, we pointed out the aims and characteristics of domain analysis during each time. And a new conception and perspective of domain analysis were given in the conclusion of this article.

Keywords Domain engineering, Components development, Requirement analysis

1 面向对象的分析与领域分析

业内普遍把领域分析当作是面向对象分析方法的一个新的应用,即面向分析方法在特定领域的应用。甚至于不加区别地把它当作一回事。实际上,它们是并列着的两个不同的分析体系,分别解决不同层次的软件分析问题。尽管总体而言,面向对象的分析和领域分析都是在软件复用的思想指导下出现的方法,而且最初的领域分析是从面向对象分析中产生的,使用了面向对象的方法,所以领域分析是面向对象分析的发展。但是目前领域分析已经独立成为一个新的分析方法,它在解决比较复杂的、比较大型的软件分析问题时有优势。

领域分析的结果是给出领域模型,领域模型可以作为项目开发的数据模型的模板或者元模型。因此领域模型对于应用项目的开发非常重要,它可以提高软件开发的效率。这一优点使得领域模型的使用越来越多。2004年美国软件设计师埃文斯(Eric Evans)倡导的领域驱动的设计,把领域分析推进到了实现领域,在软件项目实施中可以直接使用领域模型,扩展了领域模型的用途。目前领域分析已经成为软件分析的主流方法。

与此同时,云计算作为一种新的软件服务模式迅速发展起来,云计算软件的开发对领域分析提出了新的要求。作者在近年来做了一些工作,主要总结了领域分析的不同模式,并努力从中找到未来领域分析发展的新方向。总结如下:

1. 领域分析概述

复用是与软件开发与生俱来的思想。当分析师把复用限定在一定范围内的时候,就产生了领域和领域分析这些概念。1980年 Neighbors 在他的博士论文《使用组件的软件构筑》中首次提出领域分析(Domain Analysis)一词,并给出定义“是指识别、获取和组织特定领域内某类相似系统中的对象、操作等可复用信息的过程”。领域分析的目的是实现系统化的软件复用。由于它的实践指导意义,业界经过反复的实践和研究,不断提出新的概念和方法,领域分析的理论逐渐成熟。

综合领域分析的这些概念,可以认为,领域分析是在特定领域内开展的以领域定义、领域共性和特性描述、领域内概念与数据和功能及其关系识别、领域体系结构开发为目标的系统化分析过程。与系统分析不同,领域分析所关心的是一个领域内所有相似系统中的对象和活动的共同特征与演化特性,它产生的是支持系统化复用的基础设施,这些基础复用设施主要包括领域定义、开发标准、领域模型和可复用构件仓库等[1]。

在领域分析研究的发展过程中,先后在 7 位专家的领导下提出了 6 种不同的领域分析概念,建立了 5 个里程碑式的领域分析机制。他们的领域分析之间的原理各有特点、各不相同,各自适应他们自己的软件开发环境要求。但是把它们链接起来,可以找到领域分析的发展脉络,进而去寻找领域分析方法的新方向。

2. 内博兹 (James M. Neighbors) 领域分析 (1980)

在 1980 年内博兹他的论文《使用组件的软件构筑》中首次使用了领域分析的概念。他在使用 DRACO 软件开发可重

用组件的时候,把领域分析作为 DRACO 软件开发的第一步,他的领域分析概念是基于编程的,他这样描述:领域是问题域或者应用领域,领域分析不同于系统分析,它不关心特定系统中的特定活动,而是关心领域中所有系统的活动和对象,需要开发领域中的对象的通用模型,比如领域中所使用的文档的提纲的模型。在他看来,领域分析是对一定范围的系统的描述,完成领域分析的成本很高。它就像是在领域中为了设计标准部件和集成件而构建对象和操作一样。这需要问题领域中富有经验的能工巧匠来完成^[1]。

事实上,他的领域分析是与设计相结合的概念,其目标是给出 DRACO 中的领域语言表示,DRACO 领域语言是现实事物的对应映射,实际上就是现在所说的模型,只是用文字语言和文本格式来表示。作者认为内博兹领域分析实质是析取领域分析。从他的文章中我们可以分析出,他的领域分析的特点是找到领域内各个系统共同拥有的对象和操作,是对各个系统的对象和操作取交集。

3. 阿朗戈和普列托_迪亚斯(Guillermo Arango and Rubén Prieto-Díaz)领域分析(1989-1991)

阿朗戈和普列托_迪亚斯把领域分析看作一个行动过程,他们用一张 SADT 软件的输入输出图来表示他们的思想,他们的领域分析涉及的方面比较广,已经接近领域工程的思想,只不过还停留在结构化设计的阶段。作者认为他们对领域分析的看法和定位比其他专家要高一个逻辑处理的层次,将生产领域模型的分析 and 设计过程也可复用化了。作者认为阿朗戈和普列托_迪亚斯领域分析是一种过程领域分析,他们把领域分析看作一个可以自动生产领域产品的生产线,很多资源都可以作为这个生产线的输入^[2]。

可作为领域分析过程的输入有:技术文档、现有的实现、用户的调查、领域专家意见、现在和将来的需求分析,甚至开发的范式也可作为一个输入,任何相关资源的原始领域知识都可以作为领域分析过程的输入。同时,系统的输出也可以很多,各种可以形式化的东西,如概念、标准、领域过程、逻辑结构等等。对这些输出的后续处理产生通用设计模块、框架等^[3]。

4. 克日什托夫·恰尔内茨基(Krzysztof Czarnecki)领域分析(1998-2000)

恰尔内茨基对领域的概念稍有扩大,领域分析的范围也扩大了,并且划分了纵向领域和横向领域的概念。在恰尔内茨基的博士论文里,他把前人的以具体应用问题为中心的领域概念推广到以软件系统为中心,并且把领域分析放在了领域工程这个大的框架之中作为一个过程部件(process component),专门用于获取可复用特定系统的需求。整个系统有它的领域问题,这个叫做纵向领域,系统的每个部分(子系统)也有他们各自的领域问题,这个叫做横向领域。领域工程针对纵向领域开发的结果是可复用于具体系统的软件,比如一个框架可以应用于整个纵向领域,而针对横向领域开发的结果是可复用的子系统,就是组件^[4,5]。

恰尔内茨基的领域分析的目的除了开发领域模型之外,还包括选择和定义领域。通过分析系统内和系统外的例子、分析通用规则及其例外来界定领域的大小和领域上的上文特征。因此他的领域分析同时也是一种创造活动,这点对后来的人们启发很大。

而且他的领域模型也与阿朗戈不同,其既包括系统的共

同功能特征,也包括不同应用的不同特征属性及其依赖关系。注重定义领域内具体化系统的可复用、可配置的需求,这种需求一般被称作特征,将特征组合起来更有复用意义,开发特征模型也成为领域分析的一个目的。这就把领域工程提高到了配置的高度。恰尔内茨基的领域分析把系统中的不同差异都考虑在内,因此作者认为可以称之为合取领域分析。

恰尔内茨基的领域分析具体包括以下活动:领域计划、领域鉴定、领域范围的确定以及领域建模。他的领域分析为领域设计和领域实现服务,但是这种服务关系是松散的,不像阿朗戈和普列托_迪亚斯领域分析那样是一个严格的过程。

5. 埃文斯(Eric Evans)领域分析(2004)

埃文斯在 2004 年出版了他的专业著作《领域驱动设计》。全书的中心思想是为了解决软件设计的复杂性问题而进行领域建模,对于大型软件的开发经常遇到的是沟通问题,互相不能理解成为造成复杂性的主要原因^[6]。埃文斯的领域概念是一个沟通空间,他的领域分析是为了克服沟通的障碍而努力搭建一个桥梁,其内容和方法无所不包,所有的能够促进用户、软件设计师、软件分析师和编程人员之间进行沟通的方法都被他拿来使用。通过领域分析,他们能够获取一些可供领域内沟通使用的意义一致的符号、术语和概念。因此作者分析认为,埃文斯的领域分析是沟通领域分析,并不像恰氏追求一个生成式编程的完美境界,埃文斯从务实的角度出发,他的领域分析旨在减少混乱和提高效率。他把领域分析的内容延伸到软件实现方面,领域分析有时候和 Coding 联系起来进行,只要能够促进沟通,把它们纳入领域分析就是值得推崇的。这是前人所不曾拥有的特点。

而且,埃文斯把领域建模提高到了软件工程中最高层次的统领地位,一切以领域模型为中心,领域分析可以把任何东西拿来为之所用,从和有关人员进行交谈、头脑风暴,到对模型在逻辑上的离析,以至于反复整合分析、柔性设计分析,甚至于他把模式(Pattern)也用来进行领域分析。他是领域分析的集大成者。

埃文斯的领域分析方法注重均衡,没有一个方法可以标记他的领域分析特点,所有的方法都是一视同仁的,模型之间努力维持一种均衡状态,除了模型的概念上不会出现某些方面过重或者过轻的格局之外,领域分析还用来协调软件开发的每一个部分的工作,促进了开发人员之间的沟通和工作强度、工作量的彼此协调,埃文斯的领域分析中充满着优化管理的思想。

埃文斯主张用软件所涉及的业务的策略原理来引导领域分析,尽量减轻模型各部分之间的依赖性,在不失开发人员之间的合作效率的前提下提高领域模型的清晰度,让领域分析既能获取模型的概念核心,又能让模型反映出领域内要开发的系统的概貌和思想。他探索了化解大型软件的复杂性的领域分析和建模方法,从上下文(context)、精练(distillation)和大尺度结构(large-scale structure) 3 个方面进行领域分析和领域建模。

6. 布施曼_亨利_施密特(Buschmann F, Henney K, Schmidt D C)领域分析(1996-2007)

布施曼和其他专家合著的《面向模式的体系结构》共 5 卷,这套著作把模式视作软件设计的核心,其他概念都纳入了模式的体系中,领域模型也不利外。他们的软件分析的重心在于模式分析^[7-9]。尽管没有对领域分析作出专门的论述,但

是我们可以从文中找到一些涉及领域模型的材料,从这些材料中看到他们的领域分析思路。现作者总结如下:

布施曼和亨利·施密特在2007年出版的丛书第四卷的附录中给出一个领域分析的概念。领域分析是通过系统地检查应用过程来确定应用领域的核心难点和进行设计的方面,并把它们映射到一个有效的解决技术上来的一个由反馈机制驱动的归纳过程。

尽管他们把领域定义为实体域,但是同时在第五卷(218页)又说领域的概念有时是松散的,按照最为通用的解释可以这样定义:领域是在同一规则下的区域。只要赋予不同的规则以不同的意义,领域就可以对软件开发中的各种应用发挥出很好的适应性。正因为如此,领域的尺寸大小是可调节的,如同福乐(Fowler)一样,布施曼等在他们的著作中也使用领域模型来作为表示模式结构的手段。这可以认为是领域模型的最大尺寸。

在用领域模型表示和描述模式的时候,他们对领域作了问题域和解决域的区分,而且是采用从解决域中分离出问题域的思路。这点对于模式的表示比较重要,因为设计模式进而框架、结构模式都是从技术领域演化而来的,注重问题域可以让模式产生如下特点:一方面可以把开发人员、构架师、项目管理人员的工作方便地统一起来,也便于任何有兴趣的相关人员的理解和介入;另一方面,这样的注重点可以在分析和使用时避免忽视领域的弊端,越过模式而直视领域,通过对不同领域的共同经验的一般化,产生大量的编程的、构架的和管理的思想。作者分析认为,他们的领域分析是模式领域分析。

他们的领域分析主要着眼于应用领域,寻找有关真实世界的可视化的模式的分类法以便启动工作,如电信、金融、健康咨询、导航、教育等等都存在可视特征。通过问题的解构分析、领域的解构分析,建立每个问题域的结构。这样建立起来的模式在轮廓和应用方面具有偏向用户中心的特点。

7. 彼约尔奈(Denis Bjørner)领域分析(2014)

丹麦技术大学的戴南斯·彼约尔奈(Denis Bjørner)教授,在2014年(其个人网站上的日期是2013年)发表了几篇文章^[10-12],他使用形式化描述的方法,划分离散实体和连续实体进行分析。我在2013年5月之前也曾经有类似的想法。惊奇之余,简要介绍如下:

彼约尔奈认为领域分析是对实体进行划分将之组成类(或型)。他的领域分析是建立在实体关系代数的基础上,而使用了形式化语言描述的逻辑体系。其分析基于如下的概念体系。彼大师将实体划分为静实体(Endurant)和动实体(Perdurant),截至2014年3月底开发出了静实体的描述,动实体方面的成果还没有检索到。同时实体又划分为离散实体和连续实体,前者被称为零件(Part),后者被称为原料(Material)。然后离散实体又被分为原子离散实体和组合离散实体。接着使用形式化方法描述和建立了原子离散实体和组合离散实体的渠道复形(Canal Complexes)表示和分类模型(Sort Model)表示,在此基础上进行了一系列的形式化定义和推算,建立过程模型。

彼约尔奈领域分析引入了形式化方法,是形式化领域分析。其模型是用严格的形式化算式表示而不是图形表示的。我从2009年开始曾经有过使用形式化方法来描述领域模型

的想法,由于没有条件而放弃。今天看到彼约尔奈将之付诸实践,我坚信这应该是未来领域分析的主流,我们应当高度重视彼氏领域分析及其对软件分析,进而对软件设计的影响。

结束语 以上分析说明,领域分析依次经历了析取领域分析、过程领域分析、合取领域分析、沟通领域分析、模式领域分析,到形式化领域分析的不同历史阶段。这一演化过程反映了领域分析从简单到复杂,从低级到高级的发展变化过程。从中可以看出领域分析外延不断扩大。从对问题的分析开始,逐步扩展到对软件使用过程中的常用词分析,进而开发过程也被囊括在内,再进一步包揽软件管理的所有分析问题,最后对问题域进行行业分类涵盖所有领域问题。在领域分析的外延的不断扩展和推广的同时,其概念和内涵的抽象层次也不断提高,从找出可复用的共性,到过程的可复用分析,再到专门解决复用开发的分析,再到软件开发中的通用语言的分析,进一步上升到模式分析的层次,最后提升到形式化的高度。这样的变化体现了领域分析的内涵螺旋式上升的发展过程,体现了领域分析的在整个软件工程中的地位越来越高、作用越来越大的发展规律。

领域分析的发展过程的核心是领域分析方法的演变,领域分析方法的演变是由不同时期的软件开发技术的背景的变化所决定的方法进化的结果。如前所述,不同技术基础上的领域分析具有不同的特点和目的。同时,领域分析又为软件分析和设计方法的不断向前发展提供了动力。每个时期的领域分析模式和方法都促进了当时的软件设计技术的发展。面向组件开发的领域分析的长足发展就是领域分析在基于复用技术设计问题上的延伸和深入以后的结果。面向沟通的语言化领域分析是在大型软件开发技术的形成及其管理的需求发展的背景下发展起来的。模式领域分析是在软件体系结构技术在软件开发中起到越来越大重要作用的背景下产生的。

在当今软件设计技术发展背景下,云计算、普适计算等新方式、新技术的发展也给领域分析的发展带来了机遇。面向领域驱动设计的领域分析在2007年左右因为埃文斯及其同时代的福乐(Flower)的工作而成为目前软件分析的主流。2014年以来又出现了彼约尔奈的关于领域分析的形式化描述,有可能成为另一个值得重视的支流。作者研究认为,在新的技术背景下,诸如云计算、普适计算和社会计算等新方法、新领域的不断发展将会促进领域分析从以下两个方向逐渐分化,形成两个领域分析的分支:

1. 行业化。近年来领域分析呈现出行业分析的不断发展的趋势。已经成熟的一些领域模型如:地理信息模型、网络对象管理模型、建筑信息模型、医疗信息模型、财务信息模型等,在未来,行业模型所普及的领域可能进一步扩大,行业模型的分类可能进一步细化。

2. 一般化。行业之间的交叉和跨越行业的需求正在兴起,消除行业界限和技术层次的分析活动会逐步增多。在行业化的同时,一般化的需求正在悄然兴起。具体又可以从两个方向展开研究,一个方向是作者在2010年在通用需求分析方面作的一些探索^[13],可以在此基础上进行非严格逻辑方面的一般化的深入研究;另一个是方向形式化研究,彼约尔奈的形式化探索已经为领域分析在严格的一般化分析方面指明了方向^[10-12]。

行业化和一般化可能是未来领域分析的两大分野,在其

发展过程中,二者的发展规模和速度可能会出现两种情形。一种可能是领域分析将会呈现出二者平行发展的趋势,另一种可能是行业化成为领域分析研究的主干,一般化成为辅助部分。

不管怎样,仍然需要有一个统一的理论体系囊括这两个分支,探索和发现一些一般性的原理和方法,形成领域分析的基本原理。目前,领域分析还没有形成一套完整的理论,很需要也很有必要逐渐形成领域分析的独立的理论体系和研究方法,把领域分析作为一门学科来建设和开发,建立领域分析学。

在领域分析的发展过程中,领域分析的概念不断演变。今天我们认为,领域分析是对软件开发所涉及到的领域内的概念、方法进行总结和研究的活动的总称,这些活动的目标是建立领域模型,这些活动包括对领域模型中将要使用的概念、元素、实体、方法、映射等模型构件的抽象和约简。领域分析是软件分析中的一项创新活动,它可以找到和发现新的概念或者方法,或者映射等,从而可以改进和优化软件的设计和应用程序的实现,甚至可以对软件开发方法促成突破性的变革,因此,领域分析是软件开发过程中的一个十分重要的环节。软件分析师应该更多地致力于领域分析。

事实上,领域分析需要创造性的思维,是一种创新活动。未来的领域分析应该以创新为主旋律。在可预见的很长一段时间内,领域分析仍然是软件分析的可靠基础。目前,诸如云计算、普适计算和社会计算等新技术不断涌现和发展起来,给领域分析提出了新的问题和挑战。领域分析要解决这些新的问题,就应该在应对这些新技术变革带来的不相适应、不相配套等问题的过程中,全面调整自己的理论框架和方法,形成一套在软件开发中可以有效使用的领域分析技术和方法。

参 考 文 献

[1] Neighbors J M. Software construction using components[D]. University of California, Irvine, 1980

[2] Arango G, Prieto-Diaz R. Domain analysis concepts and research

directions[M]. Domain Analysis and Software Systems Modeling. IEEE Computer Society, Washington DC, 1991: 9-33

[3] De Champeaux D, Lea D. Object-oriented system development [M] // Reading, MA: Addison-Wesley, 1993

[4] Czarnecki K, Eisenecker U W. Generative programming [M]. Addison-wesley Professional, 2000

[5] Czarnecki K. Generative programming: Principles and techniques of software engineering based on automated configuration and fragment-based component models[OL]. <http://citeseerx.ist.psu.edu/showciting?cid=106655>

[6] Evans E. Domain-driven design: tackling complexity in the heart of software[M]. Boston: Addison-Wesley Professional, 2004

[7] Bushmann F, Meunier R, Rohnert H, et al. Pattern-oriented software architecture: A system of patterns [M]. John Wiley & Sons, 1996

[8] Buschmann F, Henney K, et al Pattern-oriented Software Architecture: a Pattern Language for Distributed Computing, Volume 4[M]. New York: John Wiley & Sons, 2007

[9] Buschmann F, Henney K, Schmidt D C. Pattern-oriented software architecture: On patterns and pattern languages, vol. 5 [M]. 2007

[10] Bjørner D. Domain Analysis: Endurants-An Analysis & Description Process Model[C] // Specification, Algebra, and Software: A Festschrift Symposium in Honor of Kokichi Futatsugi. Springer, 2014

[11] Bjørner D. Domain Analysis & Description: Perdurants [R]. DTU Compute and Fredsvej 11, DK-2840 Holte, Denmark, Fall/Winter, 2014

[12] Bjørner D. Domain Endurants[M] // Specification, Algebra, and Software. Springer Berlin Heidelberg, 2014: 1-34

[13] Wu Bao-ming. General analysis of requirements for risk-oriented financial data modeling[C] // Proceedings-2010 2nd International Conference on Modeling, Simulation, and Visualization Methods, WMSVM 2010. Sanya, Hainan, China, 2010: 46-49

(上接第 541 页)

[13] Lee R, Luo T, Huai Y, et al. Ysmart: Yet another sql-to-mapreduce translator[C] // 2011 31st International Conference on Distributed Computing Systems(ICDCS). IEEE, 2011: 25-36

[14] Chung W C, Lin H P, Chen S C, et al. JackHare: a framework for SQL to NoSQL translation using MapReduce[J]. Automated Software Engineering, 2014, 21(4): 489-508

[15] Ahmad F, Chakradhar S T, Raghunathan A, et al. Tarazu: optimizing MapReduce on heterogeneous clusters [J]. ACM SIGARCH Computer Architecture News, ACM, 2012, 40(1): 61-74

[16] Zhang K, Chen X. Large-scale Deep Belief Nets with MapReduce [J]. Access, IEEE, 2014, 2: 395-403

[17] Li F, Ooi B C, Özsu M T, et al. Distributed data management using MapReduce[J]. ACM Computing Surveys(CSUR), 2014, 46(3): 31

[18] Zou Q, Li X B, Jiang W R, et al. Survey of MapReduce frame operation in bioinformatics[J]. Briefings in bioinformatics, 2014, 15(4): 637-647

[19] Qian J, Miao D, Zhang Z, et al. Parallel attribute reduction algorithms using MapReduce[J]. Information Sciences, 2014, 279: 671-690

[20] Zaharia M, Konwinski A, Joseph A D, et al. Improving MapReduce Performance in Heterogeneous Environments[J]. OSDI, 2008, 8(4): 29-42

[21] Teng F, Yang H, Li T, et al. Scheduling real-time workflow on mapreduce-based cloud[C] // Innovative Computing Technology (INTECH), 2013 Third International Conference on. IEEE, 2013: 117-122

[22] Zaharia M, Borthakur D, Sarma J S, et al. Job scheduling for multi-user mapreduce clusters[R]. UCB/EECS-2009-55. EECS Department, University of California, Berkeley, 2009

[23] Chang F, Dean J, Ghemawat S, et al. Bigtable: A distributed storage system for structured data[C] // Proceeding of Conference on Usenix Symposium on Operating System Design and Implementation. 2006: 205-218

[24] 董西成. Hadoop 技术内幕 [M]. 北京: 机械工业出版社, 2013

[25] 李建江, 崔健, 王聘, 等. MapReduce 并行编程模型研究综述 [J]. 电子学报, 2012, 39(11): 2635-2642

[26] 吴煜祺, 曾国荪, 曾媛. 云计算环境下调度算法的趋势分析 [J]. 微电子学与计算机, 2012, 29(9): 103-108