

# 一种软件体系结构关注点多维分离模型

张琳琳<sup>1,2</sup> 应时<sup>1,3</sup> 倪友聪<sup>1,4</sup> 文静<sup>1</sup> 李宇翔<sup>5</sup> 张韬<sup>1</sup>

(武汉大学软件工程国家重点实验室 武汉 430072)<sup>1</sup> (新疆大学信息科学与工程学院 乌鲁木齐 830046)<sup>2</sup>

(武汉大学计算机学院 武汉 430072)<sup>3</sup> (安徽建筑工业学院数理系 合肥 230018)<sup>4</sup>

(中国移动通信集团广东有限公司深圳分公司 深圳 518048)<sup>5</sup>

**摘要** 传统方法设计出的体系结构中始终存在着一些横切现象,使得体系结构变得复杂、难以理解、难以演化和难以重用。基于关注点分离和面向方面软件开发的相关理论,探讨了体系结构中的关注点,提出一种体系结构层关注点多维分离模型。该模型通过形象地表示关注点之间的关系,识别那些在体系结构中可能产生横切的关注点,有效解决传统软件体系结构设计中的关注点横切问题,为软件体系结构的演化和重用提供了重要的指导作用。

**关键词** 软件体系结构,关注点多维分离,面向方面开发方法

## Model for Architectural Multi-dimensions Separating of Concerns

ZHANG Lin-lin<sup>1,2</sup> YING Shi<sup>1,3</sup> NI You-cong<sup>1,4</sup> WEN Jing<sup>1</sup> LI Yu-xiang<sup>5</sup> ZHANG Tao<sup>1</sup>

(The State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)<sup>1</sup>

(The School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China)<sup>2</sup>

(The School of Computer, Wuhan University, Wuhan 430072, China)<sup>3</sup>

(Department of Mathematics and Physics, Anhui Institute of Architecture & Industry, Hefei 230018, China)<sup>4</sup>

(Shenzhen Branch, Guangdong Province Limited Corporation, China Mobile Communication Group, Shenzhen 518048, China)<sup>5</sup>

**Abstract** In the architecture designed with traditional methods there are many crosscutting phenomena that make architecture more complex, unable to understand and reuse. This paper focused on the concerns in system architecture level, proposed a model for multi-dimensions separating of concerns (MDSoC), based on the separation of concerns principles and aspect-oriented software development (AOSD) techniques. Through describing lively the relationships between concerns and then identifying the potential crosscutting concerns in the architecture, the model can effectively resolve the crosscutting problems in traditional architecture design, and provide the helpful guides for architecture further evolution and reuse.

**Keywords** Software architecture, Multi-dimensions separating of concerns (MDSoC), Aspect oriented software development (AOSD)

## 1 引言

软件体系结构是当前软件工程学术界和产业界都十分关注的领域。研究人员在对软件体系结构的深入研究和广泛应用中发现利用常规方法设计出的软件体系结构方案中始终存在着一些横切(crosscutting)的行为和横切的特征。这些行为和特征横切了组成软件体系结构的组件和连接器,造成组件和连接器之间的紧耦合,使得组件和连接器中的内容变得复杂,限制和约束了体系结构的表现能力。近年来,随着面向方面程序设计(Aspect-Oriented Programming,简称AOP)技术<sup>[1]</sup>的成熟及进一步的深入发展和广泛应用,尤其是早期方面(early aspect)<sup>[2]</sup>的提出和深入研究,为体系结构关注点的

分离提供了新的理论和技术支持。

基于关注点分离(Separation of Concerns,简称SoC)<sup>[3]</sup>和面向方面软件开发(Aspect-oriented software development,简称AOSD)<sup>[4]</sup>的相关理论,本文提出了一种体系结构关注点多维分离模型。通过分析、识别体系结构层关注点以及它们之间的关系,从多维的角度分离它们,然后映射到体系结构的构成要素,达到从本质上消除体系结构中的横切行为和横切特征的目的。结合网上银行实例,说明该模型是如何多维分离体系结构关注点的。

## 2 MDSoC与AOSD

### 2.1 MDSoC

到稿日期:2008-04-02 本文受国家高技术研究发展计划863项目(2006AA01Z168),国家自然科学基金资助项目(60773006),高等学校博士学科点专项科研基金资助项目(20060486045),北京教育委员会科技发展计划面上项目(KM200710772003)资助。

张琳琳(1974—),女,博士生,讲师,主要研究方向为体系结构、面向方面软件开发,E-mail: zllnadasa@xju.edu.cn;应时(1965—),男,博士,教授,博士生导师,CCF会员,主要研究领域为面向对象软件工程方法、基于组件的软件工程方法、软件体系结构和模式、软件的可重用性与互操作性等。

Dijkstra 指出,SoC 是软件工程最重要的原则之一,它体现了“分而治之”的思想。然而现有的软件开发方法仅支持关注点沿一个主导维度进行分离——主导分解专制(tyranny of the dominant decomposition)。针对这一现象,Peri 及其同事提出了多维关注点分离(Multi-dimension separation of concerns,简称 MDSoc)<sup>[5]</sup>的方法,即沿着多维的分解维度分离相互重叠(overlapping)的关注点,在主导维度上用常规的方式,即基于模块进行分解,在其它维度上用超切片(hyperslices)封装在一个制品中或横切多个制品的一个特定关注点。这一方法为体系结构层关注点的多维分离提供了理论基础。

## 2.2 AOSD

AOP 技术用方面(aspect)封装那些散列(scattering)和混杂(tangling)在代码中的横切关注点(crosscutting concerns),提高了系统的模块化程度,并在此基础上逐渐形成了面向方面的软件开发方法。同时,AOSD 实现了整个系统中基本关注点与横切关注点的分离,无疑是实现关注点分离的优秀技术和手段之一。本文利用关注点分离和 AOSD 相关原理,探讨在体系结构层多维分离关注点的意义,识别那些在体系结构层可能出现的横切关注点,通过一个建模过程,实现一种通用的体系结构多维关注点分离模型,更好地支持 AOSD 中的方面概念。

## 3 体系结构多维关注点分离模型

### 3.1 动机

在软件体系结构的设计、分析、演化和重用过程中,涉及的关注点主要有 2 种,如图 1 所示:描述业务逻辑的功能性关注点和描述软件系统的系统性关注点。系统性关注点指的是与系统的整体性能密切相关的非功能性关注点。与功能性关注点相比,系统性关注点将涉及更多的涉众(stakeholders),如软件体系结构设计人员、维护人员、质量控制与保障人员、软件体系结构的重用者等。同时,系统性关注点也更多地涉及问题空间的很多方面,如业务需求、系统需求、质量需求、软件体系结构设计原理与方法、技术性约束条件等。软件系统的每个关注点代表一个特定的设计目标,需要在软件体系结构设计阶段得到独立的标识和描述,以便在软件开发生命周期中的其它阶段(如维护阶段等)得到跟踪。这些关注点形成了一个体系结构的关注点空间。

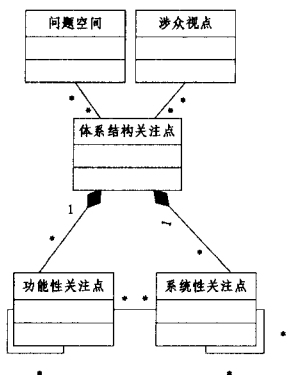


图 1 体系结构层关注点

常规的软件体系结构设计方法几乎都是首先以业务过程和用例脚本为主要关注点,设计出软件体系结构的基本结构。然后,在软件体系结构中以直接横切组件和连接器的形式,将

满足非功能需求的设计方案加入到基本结构中去。由于没有规范地考虑如何满足非功能需求的设计步骤,导致设计过程变得复杂起来。由于非规范地加入满足非功能需求的设计方案,模糊了组件和连接器的角色与边界,使得设计结果也变得难以理解。

因此,为了设计出正确的、高质量的软件体系结构,在软件体系结构抽象层次上,面向软件体系结构的设计、演化和重用目标,对各种关注点进行建模十分必要。此外,由于关注点牵涉的内容太多,为了降低软件体系结构设计、演化和重用的难度,简化其过程,需要在多个不同的维度上,分别考虑不同的关注点。

### 3.2 体系结构关注点多维分离模型

为了更好地描述体系结构层的关注点,本文提出一种通用的关注点多维分离模型,将系统的需求规约按照功能性关注点和系统性关注点的分类方式,良好地组合在一起,形成了一个关注点-视点空间。图 2 描述了体系结构关注点多维分离模型,该模型是一个可延伸的立方体空间,由水平或垂直两个方向上的平面构成。其中,最底层的平面,代表体系结构功能性关注点维度;平行于该底面的平面,代表体系结构系统性关注点维度;垂直于该底面的平面,代表体系结构涉众视点(viewpoint)维度。平面之间的垂直交错表明不同维度与涉众视点维度之间的关系,即系统性关注点、功能性关注点与涉众视点维度之间的关系。关注点之间的箭头表明不同平面之间关注点的投射关系,箭头的方向表明关注点的投射方向。系统性关注点往往横切多个功能性关注点,表示为从系统性关注点出发的箭头指向多个功能性关注点,因此关注点之间的箭头还表示了受某个关注点影响的其他关注点集合。如图 2 所示,日志影响了体系结构设计者所关心的多个功能性关注点。

软件体系结构功能性关注点维度包含系统全部功能关注点,每个功能性关注点实现了子功能的全部或部分功能,如查找、修改、删除等。体系结构系统性关注点维度由多个系统性关注点组成,一个系统性关注点就是一个维度,如日志、安全性、可靠性、健壮性、适应性等均作为一个单独的维度。涉众视点维度包括体系结构不同涉众角色的视点维度,如体系结构设计者或重用者。与系统性关注点相似,每一个维度对应一种涉众角色视点;与功能性关注点相似,一个视点维度是一组关注点集合。

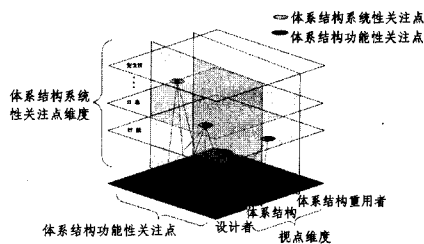


图 2 体系结构关注点多维分离模型

体系结构关注点多维分离模型形象直观地刻画了体系结构关注点之间的关系。在该模型中,那些相互之间存在一对多或多对多投射关系的关注点,势必会在体系结构的后期设计中表现出横切特性。因此,关注点多维分离模型能够帮助架构师在体系结构设计的早期阶段识别出 AOSD 的方面概念,有助于设计出正确、高质量的软件体系结构。此外,任何

系统的体系结构在设计的早期阶段,都可以用该模型表示关注点之间的关系,因此该模型是一个通用模型。

### 3.3 体系结构关注点多维分离建模过程

为了建立体系结构关注点多维分离模型,基于需求规约,本文提出一个系统地建立体系结构关注点多维分离模型的过程。该过程包括以下4个主要步骤,如图3所示:识别关注点、描述关注点、组合关注点以及对系统性关注点进行映射。

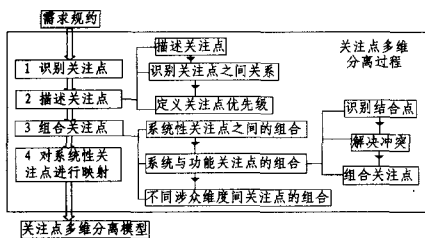


图3 体系结构关注点多维分离建模过程

(1)识别关注点。关注点的识别建立在完整的需求分析文档以及从各种涉众那里得到的有用信息的基础之上。识别体系结构的功能性关注点的方法很多,大多数需求分析的方法和工具都可用于此。本文采用用例获取功能性关注点,采用NFR(Non-Function Requirements)框架<sup>[6]</sup>获取系统性关注点。

(2)描述关注点。在识别出系统中的关注点之后,必须对关注点进行描述和规约。规范合理地描述关注点,不仅使关注点易于理解,而且便于进一步的设计、重用和跟踪。本文采用关注点模板<sup>[7]</sup>来描述关注点。该模板描述了关注点所属维度、涉及的涉众、关注点行为、典型应用示例等信息。除此之外,还必须包括关注点之间的关系描述,表明该关注点可能涉及到的关注点集合,此项工作由领域专家或有经验的设计人员来完成。需要说明的是,多维分离模型中的关注点描述是简单抽象的,不涉及具体问题域或特定系统的信息。在具体的问题域或特定系统中,将结合具体问题域或特定系统信息对关注点进一步做详细描述,如增加或删除关注点关系中的元素。由此可以看出,多维分离模型中关注点集合的构建是一个迭代和增量过程。

(3)组合关注点。通过一组组合规则,将体系结构的功能性关注点和系统性关注点组合起来,形成整个系统的初步概貌。本文定义了3种类型的关注点组合:系统性关注点之间的组合、系统性与功能性关注点的组合,以及同一水平维度上不同涉众视点维度间的组合。其中,系统性关注点的组合采用NFR实现;系统性与功能性关注点的组合情况比较复杂,需要进一步分析解决;同一水平维度不同视点维度间的组合采用同名组合策略,即两个或多个视点维度上的同名关注点可以被组合在一起,从而形成一个粒度较大的关注点。

系统性关注点与功能性关注点的组合中一个重要的问题是结合点的识别。结合点是指系统性关注点和功能性关注点可能发生组合的地方。考虑到一个系统性关注点可以同时横切多个功能性关注点,而多个系统性关注点也可能横切一个功能性关注点的情况,需要建立冲突检测表<sup>[7]</sup>来帮助结合点的识别。冲突检测表描述了关注点之间相互冲突的强弱程度,便于在体系结构选择时进行权衡。如安全性和响应时间,安全性越高,势必会加大系统开销,致使响应时间变长;响应时间越短,安全性就会下降,因而最终的体系结构需要在二者

之间进行权衡。另外,冲突检测表还考虑了多个关注点之间的优先级情况,明确多个关注点横切同一个关注点的顺序。最后通过定义一组组合规则,在这些组合规则的指导下,对关注点进行组合。

(4)对系统性关注点进行映射。一个体系结构系统性关注点在体系结构设计时可能会映射为系统的一个功能、设计决策或是影响系统过程的其它决策(如可维护性)<sup>[7]</sup>。对系统性关注点进行映射,将它们划分为不同种类,有利于判断它们对后期开发过程的影响。

## 4 案例研究

结合网上银行系统,说明本文的关注点多维分离模型。网上银行系统的需求简单描述如下:用户可以新建账户、删除账户和修改密码;提供转账交易服务,包括银行内部之间的转账以及跨行对外支付转账;提供查询功能,用户可以进行查询账户余额及明细查询等操作。除此之外,需要对用户的各项操作进行记录;所有的交易必须保证安全可靠,迅速快捷;具备良好高效的性能,方便用户使用。

按照体系结构关注点多维分离的建模过程,首先从网上银行系统需求规约中分解出系统的功能性需求与非功能性需求,即通过用例和NFR框架识别出功能性关注点和系统性关注点。系统的功能性关注点包括:(1)账户管理,包括①新增账户、②删除账户、③修改密码;(2)账户查询,包括④账户余额查询、⑤账户明细查询;(3)用户交易,本文指转账交易,包括⑥对内和⑦对外两种。

在本文中,仅列出几个典型的系统性关注点,包括响应时间、机密性、安全性、可用性和日志。响应时间描述用户进行各种转账业务时系统应具备快速响应的能力。机密性指用户的账户和密码不能为系统之外的他人所获取。安全性则包括两方面的内容:一是用户账户的安全,二是用户信息在网络传输过程中的安全。可用性则是指系统一直处于活动状态,方便用户使用。日志则要求记录整个系统的全部行为。

在识别了网上银行系统的关注点之后,需要对它们进行描述、组合以及映射。表1至表3分别是关注点描述、组合以及映射的实例。通过表1中对安全性关注点的具体描述,可知安全性与机密性、可用性相关。通过表2中对系统性关注点与功能性关注点组合的描述,可知二者在体系结构设计者视点维度上的结合点。其中,SC指系统性关注点,SV指涉众视点。表3描述了系统性关注点的映射结果。最后得到如图4所示的网上银行系统的体系结构关注点多维分离模型。

表1 安全性关注点的具体描述

关注点名称	安全性
所属维度	安全性维度
涉众	体系结构设计者 (1)系统必须处于活动状态;
描述	(2)用户登录时验证用户身份,确保用户身份的合法性; (3)用户交易过程中,保证用户信息不被他人窃取; (4)保证用户数据传输的安全。
涉及关注点	可用性、机密性

表2 功能性关注点和系统性关注点的结合点

	SC	响应时间	机密性	可用性	日志
SV	体系结构设计者	①-⑦	⑥⑦	①-⑦	①-⑦

表3 系统性关注点的映射

系统性关注点	映射结果
响应时间	功能、横切关注点
机密性	功能、横切关注点
安全性	设计决策
可用性	设计决策
日志	功能、横切关注点

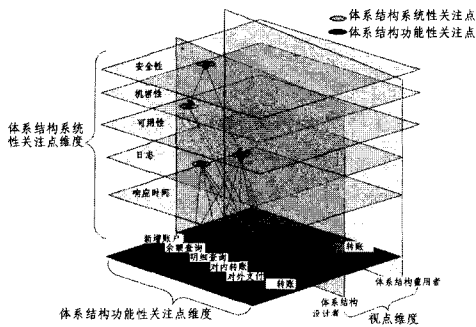


图4 网上银行系统体系结构关注点多维分离模型

由表1知安全性维度在机密性和可用性维度上均有投射。由表2知系统性关注点对功能性关注点的投射情况。由表3知安全性最终映射为体系结构的设计决策,因此它不直接对应功能性关注点维度上的任何关注点;其余系统性关注点均是如此。更重要的是,表3描述了体系结构中的横切关注点。由图1知,体系结构涉众视点关心所有的体系结构关注点,这在图4涉众角色视点维度中得以体现。体系结构设计者关心与设计有关的因素,重用者关心如何重用。由此,可将对内转账和对外支付两个关注点复合为转账关注点,重用于银行领域内。与图2相比,图4则更加具体且特定于银行领域。

### 5 相关工作

文献[5]是最早论述 MDSoc 方法的,提出沿着多维的分解维度、分离相互重叠的关注点方法。文献[8]提出的 Cosmos 是一个关注点-空间建模模式,通过关注点、关注点之间关系和谓词来建模关注点-空间。关注点被划分成表示概念的逻辑关注点和表示软件系统元素的物理关注点两类。然而,上述方法只是提出了一个基本的抽象模型和初步的有关想法,没有针对每个阶段的开发任务研究具体的多维分离关注点的方法。文献[9]提出一种基于关注点多维分离的需求工程方法,用超立方体(hypercube)表示需求层的关注点空间,然而并没有给出具体的多维分离模型,而且该文的工作仅限于需求阶段。文献[10]虽然提出了在体系结构阶段引入关注点多维分离的思想,但出发点是现有的体系结构描述语言不能充分地实现关注点的分离,与本文的工作顺序恰好相反。本文将关注点多维分离的结果应用于体系结构的设计,继而才进行体系结构的描述。

文献[11]针对模型驱动软件项目过程采用多种开发方式、缺少系统化方法指导的问题,提出了一种模型驱动过程框架设计方法。该方法采用一般化、行为化和抽象化作为元关注维,对开发方式进行比较,并组合这三维的期望演化曲线,给出过程实现模型框架。然而,该文所描述的三维只是一个抽象概念框架,并不对应具体软件系统的开发过程。

有关需求分析与体系结构平滑过渡问题的研究成果主要集中在传统开发方法上,较少地涉及 AOSD 领域。还有一些研究成果致力于研究体系结构需求,如文献[12]等,然而它们多数是从软件开发实践的角度来阐述这一问题,较少在学术方面进行深入的研究和论证。文献[13]针对面向方面需求向

面向方面体系结构映射过程中常常会丢失很多重要信息的问题,提出一个跟踪性框架(schema),记录在映射过程中产生的信息,把面向方面的需求和面向方面的体系结构工作关联起来。但是该文的重心是保证需求到体系结构中信息不丢失,并没有针对二者之间的鸿沟提出有效的解决方案。本文则针对面向方面的体系结构,提出了一种在体系结构层次上多维分离体系结构关注点的模型,为面向方面需求向面向方面体系结构的平滑过渡提供了支持。

**结束语** 本文提出了一种体系结构多维分离关注点的模型,并简要描述了这一模型的建模过程;结合网上银行系统,给出了这一具体系统的体系结构关注点多维分离模型。多维分离模型为体系结构架构师提供了形象直观的表达,便于架构师综合、全面地考虑体系结构的关注点。通过对关注点及其关系的识别,找出那些在体系结构中可能产生横切的关注点,从而有效地解决传统体系结构设计方法在体系结构中产生的横切问题。此外,多维分离模型建模过程的输入是需求规约,输出是可直接用于体系结构设计的多维分离模型,因而实现了需求分析到设计阶段的平滑过渡。下一步的工作方向是进一步完善多维分离模型的建模过程,探讨如何更加详细、规范地描述建模过程的各种制品。

### 参考文献

- [1] Kiczales G, Lamping J, Mendhekar A, et al. Aspect-oriented Programming//Proc. of 11th European Conference on Object-oriented Programming (ECOOP1997). New York: Springer-Verlag, 1997: 220-242
- [2] Early Aspects Web Site. <http://early-aspects.net/>
- [3] Dijkstra E W. A Discipline of Programming. Englewood Cliffs, NJ: Prentice Hall, 1976
- [4] AOSD Web Site. <http://www.aosd.net>
- [5] Tarr P, Osshier H, Harrison W, et al. N Degrees of Separation: Multi-dimensional Separation of Concerns//Proc. of the 21st Int'l Conf. on Software Engineering (ICSE1999). Washington: IEEE, 1999: 107-119
- [6] Chung L, Nixon B A, Yu E, et al. Non-functional Requirements in Software Engineering. Norwell: Kluwer Academic Publishers, 2000
- [7] Rashid A, Moreira A, Araujo J. Modularisation and Composition of Aspectual Requirements//Proc. of the 2nd on Aspect-oriented Software Development. New York: ACM, 2003: 11-20
- [8] Sutton S, Rouvellou I. Modeling of Software Concerns in Cosmos//Proc. of the 1st International Conference on Aspect-oriented Software Development (AOSD'02). New York: ACM Press, 2002: 127-133
- [9] Moreira A, Araujo J, Rashid A. Multi-dimensional Separation of Concerns in Requirements Engineering//Proc. of 13th Requirements Engineering Conference (RE'05). Washington: IEEE, 2005: 285-296
- [10] Kandé M M, Strohmeier A. On the Role of Multi-dimensional Separation of Concerns in Software Architecture//Proc. of the OOPSLA 2000 Workshop on Advanced Separation of Concerns in Object-oriented Systems. New York: ACM Press, 2000: 1-6
- [11] 段玉聪, 顾毓清. 多维关注分离的模型驱动过程框架设计方法. 软件学报, 2006, 17(8): 1707-1716
- [12] Svetinovic D. Architecture - Level Requirements Specification [EB/OL]. <http://se.uwaterloo.ca/~straw03/finals/Svetinovic.pdf>
- [13] Chitchyan R, Pinto M, Fuentes L, et al. Relating AO Requirements to AO Architecture [EB/OL]. [http://www.aosd-europe.net/allPublications/AnalysisDesign/Architecture/chitchyan\\_pinto\\_fuentes\\_rashid\\_ea05.pdf](http://www.aosd-europe.net/allPublications/AnalysisDesign/Architecture/chitchyan_pinto_fuentes_rashid_ea05.pdf)