

一种软件体系结构关注点分析方法

张琳琳^{1),2)} 应 时^{1),3)} 倪友聪^{1),4)} 赵 楷^{1),2)} 文 静¹⁾

¹⁾(武汉大学软件工程国家重点实验室 武汉 430072)

²⁾(新疆大学信息科学与工程学院 乌鲁木齐 830046)

³⁾(武汉大学 计算机学院 武汉 430072)

⁴⁾(安徽建筑工业学院数理系 合肥 230018)

摘 要 在体系结构的设计、演化和重用过程中涉及众多的关注点,而且它们之间存在着复杂的关系,然而目前还缺乏有效的对这些关注点及其关系进行描述和分析的方法.针对这一问题,在系统收集并显式标识各种体系结构关注点及其关系的基础上,文中提出一种软件体系结构关注点分析方法.该方法利用时段时序逻辑对关注点之间的横切关系进行形式化描述和分析,可以发现横切关注点之间的时序冲突,有助于提高面向方面软件体系结构的设计质量.最后结合案例给出了该方法的实施过程.

关键词 关注点多维分离;体系结构关注点;面向方面软件体系结构;时段时序逻辑

中图法分类号 TP311

DOI号: 10.3724/SP.J.1016.2009.01782

An Analysis Approach for Software Architectural Concerns

ZHANG Lin-Lin^{1),2)} YING Shi^{1),3)} NI You-Cong^{1),4)} ZHAO Kai^{1),2)} WEN Jing¹⁾

¹⁾(State Key Laboratory of Software Engineering, Wuhan University, Wuhan 430072)

²⁾(School of Information Science and Engineering, Xinjiang University, Urumqi 830046)

³⁾(College of Computer Science, Wuhan University, Wuhan 430072)

⁴⁾(Department of Mathematics and Physics, Anhui Institute of Architecture & Industry, Hefei 230018)

Abstract Varieties of concerns are involved in processes of software architecture design, evolution and reuse. What's more there are the complex relationships between the different concerns. At present the approaches, which can be used to describe and analyze architectural concerns and theirs relationships, are still lacking. So this paper proposes an analysis approach for software architectural concerns based on collecting and summarizing these architectural concerns. In this approach ITL(Interval Temporal Logic)is used to analyze the crosscutting relationship between architectural concerns in order to find the temporal conflict between the crosscutting concerns. The approach contributes to improvement of software architecture design quality. Finally, this paper illustrates implementation process of this approach combined with a case.

Keywords multiple dimension separating of concerns (MDSoC); architectural concerns; aspect oriented software architecture; interval temporal logic

收稿日期:2009-04-19;最终修改稿收到日期:2009-08-10. 本课题得到国家自然科学基金(60773006)和高等学校博士学科点专项科研基金(20060486045)资助. 张琳琳,女,1974年生,博士,讲师,主要研究方向为软件体系结构和面向方面软件开发. E-mail: zllnadasha@yahoo.com.cn. 应 时,男,1965年生,教授,博士生导师,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、语义 Web 技术. 倪友聪,男,1976年生,博士研究生,讲师,主要研究方向为软件体系结构和面向方面软件开发. 赵 楷,男,1976年生,博士研究生,讲师,主要研究方向为面向服务软件体系结构和语义 Web 技术. 文 静,女,1982年生,博士研究生,主要研究方向为软件体系结构和面向方面软件开发.

1 引言

在对软件体系结构进行长期深入研究的过程中,人们发现体系结构设计、演化和重用过程中存在大量的关注点.这些关注点不仅数量庞大,而且彼此之间的关系纷繁复杂,它们不但涉及多种不同类型的体系结构涉众(stakeholders),而且还涉及问题空间和解空间的多种事宜.具体而言,软件体系结构的不同类型涉众人员所关注的内容和感兴趣的事物有所不同,如体系结构设计人员可能更加关心如何将需求用具体的体系结构构成元素封装或表示,而体系结构的维护人员则更多地关心一个组件是否留有适当的接口,以便进行扩展等问题.从关注点所包含的信息和内容来看,关注点涉及问题域和解空间的多个方面,如软件系统的业务需求、系统需求、质量要求、软件体系结构设计原理与方法、技术性约束条件等.因此为了设计出高质量的软件体系结构,在软件体系结构抽象层次上,对各种关注点及其关系进行建模和分析十分必要.

许多表示非功能需求和质量要求的体系结构关注点大都表现出横切特征,可借助 Aspect 方法和技术进行处理.目前已涌现出许多面向方面软件体系结构的设计方法,这些方法运用独立的设计元素显式地封装软件体系结构横切关注点,并按照一定方式完成软件体系结构横切关注点和功能性关注点的组合.然而这些方法没有将软件体系结构设计过程中的各种关注点在多个维度上予以标识和组织,使得软件体系结构的关注点难以向需求分析阶段和编码实现阶段进行跟踪.此外,这些方法也未全面、系统地考虑体系结构关注点之间的关系,特别是横切关注点之间关系,这可能导致横切关注点之间潜在的时序冲突.

针对上述问题,本文以关注点分离原理为基本思想,以体系结构关注点为核心,在体系结构关注点多维分离模型 SA-MDSoC(Multi-Dimensional Separating of Concern at Software Architectural level)的基础上,提出一种软件体系结构关注点分析方法.它通过时段时序逻辑描述和分析关注点之间的横切关系,可以发现横切关注点之间的时序冲突,从而帮助提高面向方面软件体系结构的设计质量.

本文第 2 节总结相关研究工作;第 3 节介绍体系结构多维关注点分离模型,它从多个维度定义体系结构设计、演化和重用中可能所涉及到的各种关

注点及其之间的关系;第 4 节是本文的核心部分,在体系结构多维关注点分离模型的基础上,提出软件体系结构关注点分析方法;第 5 节结合网上拍卖系统的案例,给出本文所提出的分析方法的实施过程.最后总结全文.

2 相关工作

面向方面编程(Asspect-Oriented Programming, AOP)技术^[1]的兴起和成熟,再一次掀起了人们对关注点分离研究的热潮. Sutton 等学者提出将关注点视为软件开发生命周期中的一等实体,并归纳了代码层关注点之间各种关系^[2]. Peri 等人在文献^[3]中建议将关注点在多个维度上同时分解,以减少变化对软件系统带来的影响,从而提高软件的可演化能力,该文被 ICSE 2009 评为十年来最具影响力的论文.这些工作激发了人们在需求、设计阶段研究关注点及其之间关系的兴趣.

面向方面需求工程引入了关注点多维分离的思想,用于解决需求中的横切问题. Moreira 等人提出了面向关注点的需求分析方法 CORE^[4]. 该方法不再区分功能和非功能需求,而将它们统一用关注点予以表示.此外,COMPASS^[5]还给出了由 CORE 方法所建立的面向方面需求模型向面向方面体系结构映射的方法,从而将需求中的关注点映射为面向方面软件体系结构中相应的构成要素. CORE 方法强调需求关注点之间相互影响的分析,使得利用 COMPASS 映射方法获得高质量的面向方面体系结构成为可能. CORE 方法中使用超立方体(hypercube)表示需求层的关注点空间.超立方体的每个面表示一个关注点,可通过将超立方体的面投影到其它面上,以观察关注点之间的相互影响,进而实现在需求阶段的关注点权衡分析.然而 CORE 缺乏对关注点之间时序关系的描述,也没有给出关注点时序冲突的分析方法.

面向方面软件体系结构设计方法重点研究横切关注点在体系结构中的建模问题.面向方面软件体系结构特别关注如何处理体系结构层方面^[6],即那些在体系结构中不能被模块化、局部化的横切关注点.目前研究人员已在面向方面体系结构方面取得了许多研究成果.从对横切关注点的处理方式上可将这些研究成果分为两类.一类研究工作认为横切关注点与传统组件所封装的关注点没有本质的区别,它们仅在交互方式上有一定的差异.因此,这类

研究工作体系结构中用普通组件表示方面,同时引入方面连接器用于表示方面与其它体系结构元素之间的交互.其中有代表性的工作有:AO-ADL^[7]、Aspectual ACME^[8]、FuseJ^[9]等.另一类研究工作使用体系结构层方面组件封装横切关注点,并利用各自所特有的方式完成方面组件与其它体系结构元素的组合.其中具有代表性的工作包括:DAOP-ADL^[10]、Navasa 提议^[11]、PRISMA^[12]等.上述这些研究工作既未对体系结构横切关注点进行有效的组织和管理,也很少涉及体系结构横切关注点的时序冲突分析.

此外,在体系结构层用方面技术处理非功能需求方面也开展了许多研究工作,它们为本文的工作提供了一定的依据.非功能需求的处理长期以来一直困扰着研究人员和实践者.在对非功能需求进行了长期观察之后,人们发现非功能需求大多体现出横切特征,可借助方面方法和技术进行处理.目前在软件体系结构设计阶段,利用方面封装非功能需求,已取得了一定的成果.其中较有代表性的有:文献[13]提出了一种面向方面的框架 FDAF (Formal Design Analysis Framework) 用于设计和分析分布式、实时系统.在这一框架中,非功能需求被定义为方面库 (aspect repository) 中的可重用方面.文献[14]提出了用方面组件表示可操作化的非功能需求.文献[15]将非功能需求作为可方面化 NFRs (aspectable Non-Functional Requirements) 注入到软件体系结构中.这些研究工作主要集中于非功能需求的表示和管理,但对如何分析非功能需求之间的关系讨论较少.

3 体系结构关注点多维分离模型

为了能有效地标识、组织、描述和分析体系结构

关注点及其关系,我们设计了体系结构关注点多维分离模型 SA-MDSoC. 它从多个维度定义了体系结构设计、演化和重用过程中可能所涉及的各种关注点及其它它们之间的关系.

为了便于阐述本文的研究工作,下面列出 5 个必要的概念:体系结构关注点、主关注点和横切关注点等.

概念 1. 体系结构关注点是指体系结构设计、演化和重用过程中所关心或感兴趣的事物,它可以表现为某个问题解决方案的一部分,也可以是体系结构需要完成的一个通用性的功能或是某个质量属性.

概念 2. 主关注点 (primary concern) 是封装体系结构中表示功能和特征的成分.

概念 3. 横切关注点 (crosscutting concern) 是封装体系结构中表示非功能或约束的成分.

概念 4. 主维 (primary dimension) 是由一组体系结构主关注点组成的集合.

概念 5. 横切维 (crosscutting dimension) 是由一组体系结构横切关注点组成的集合.

SA-MDSoC 模型^[16-17] 以“1+X”风格组织、管理和描述各种体系结构关注点.在此“1”指一个主维,“X”指若干个横切维.此外,SA-MDSoC 模型还定义了三类关注点之间的关系,即主关注点-主关注点、横切关注点-主关注点以及横切关注点-横切关注点之间的关系.其中,横切关注点-主关注点之间的横切关系是由横切关注点上的横切动作与主关注点上的动作之间组合关系体现的.

如图 1 所示,SA-MDSoC 模型分为 3 种不同的模型,它们分别为:通用的 SA-MDSoC 模型、特定领域的 SA-MDSoC 模型以及具体应用系统的 SA-MDSoC 模型.通用的 SA-MDSoC 模型是在对目前已有研究工作^[18-19] 的收集和整理并归纳总结的基

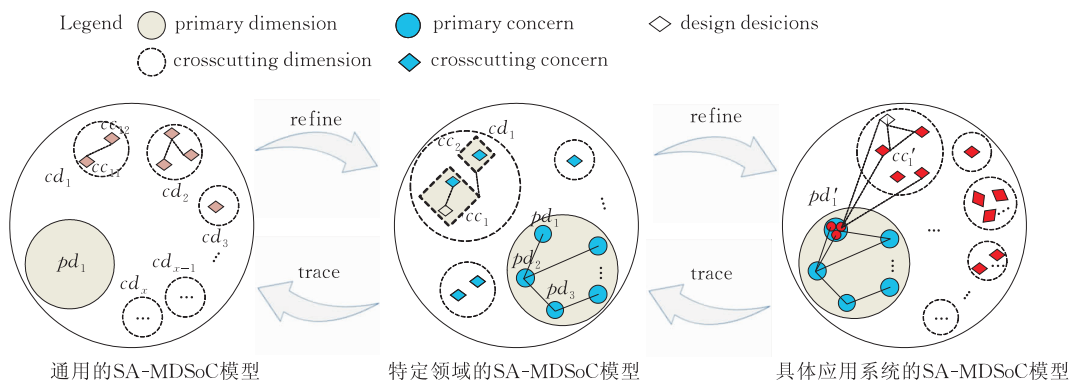


图 1 体系结构关注点多维分离模型示意图

基础上,尽可能全面地标识出体系结构设计、演化和重用过程中涉及到的各种横切关注点及其之间的关系,为体系结构设计人员根据软件需求进行体系结构关注点的甄别和选取、关注点之间关系的确定提供了有用参考.特定领域的 SA-MDSoC 模型则根据某个特定领域的相关知识,选取、精化通用的 SA-MDSoC 模型中的某些关注点,得到适用于该领域的 SA-MDSoC 模型.具体应用系统的 SA-MDSoC 模型可以根据具体应用系统的需求,通过对上述两种 SA-MDSoC 模型中定义的关注点及其关系进行相应的选取和精化而建立.

通用的 SA-MDSoC 模型定义了 13 个维度,70 个横切关注点.这些维度和关注点的设计考虑了诸多因素,如体系结构需求、体系结构层的非功能属性、质量特性、经典的体系结构设计方法以及产业界的典型应用(如 IBM 公司等)等.此外,通用的 SA-MDSoC 模型中还定义了 3 类关注点之间的关系:

① 主关注点-主关注点之间的关系有:数据交换、消息交换和调用等关系.主关注点为了完成某些功能,与其它主关注点之间可能发生数据交换、消息交换和调用等通信关系.

② 横切关注点-主关注点之间的关系有:横切关系、约束关系.横切关系表示横切关注点对主关注点有横切影响.当横切关注点不能具体化为某个或某些特定的横切操作,而是成为对某个功能的要求或期望某个功能所达到的目标时(如响应时间),这时横切关注点与主关注点之间将只存在普通意义上的约束关系.

③ 横切关注点-横切关注点之间的关系有:正贡献关系、负贡献关系和时序关系.横切关注点之间的正贡献关系表明横切关注点之间存在相互增强的关系,一个横切关注点的存在会增强另一个横切关注点的作用.横切关注点之间负贡献关系表明横切关注点之间存在相互削弱、干扰或冲突关系,一个横切关注点的存在会削弱或干扰另一个横切关注点的作用.横切关注点之间的时序关系是指如果不同横切关注点在相同时间间隔上横切相同的主关注点,则这些横切关注点之间可能存在某种时序关系.横切关注点之间存在的时序关系可分为两种:之前、之后.

特定领域的 SA-MDSoC 模型对于设计特定领域体系结构具有一定的参考价值.结合特定领域的功能需求,特定领域的 SA-MDSoC 模型中定义了若

干主关注点,建立了主维.此外,按照通用的 SA-MDSoC 模型对维度的划分,建立与特定领域相关的横切维,并从中选取和精化横切关注点.与此同时,特定领域的 SA-MDSoC 模型还定义了领域中各种体系结构关注点之间的关系.

具体应用系统的 SA-MDSoC 模型为体系结构设计人员建立具体应用系统的软件体系结构提供支持.具体应用系统的 SA-MDSoC 模型是在参考上述两个模型的基础上建立的,建立的过程与面向特定领域的 SA-MDSoC 模型的类似.与上述两个模型相比,该模型中关注点更具体、抽象层次更低、关注点粒度更小,并且关注点之间的关系更加繁多.

4 软件体系结构关注点分析方法

SA-MDSoC 模型中的体系结构关注点不仅类型各异,关注点之间的关系也较为复杂.在这些关系中最为核心、最为重要的是横切关注点与主关注点之间的横切关系.本文的软件体系结构关注点分析方法正是借助对横切关系的分析,推理出横切关注点之间是否存在时序冲突.由于横切关系具有鲜明的时序特性,因此我们选用时段时序逻辑 ITL^[20-22](Interval Temporal Logic)作为横切关系的描述和分析工具.

4.1 时段时序逻辑简介

时段时序逻辑 ITL 可用于描述事件和动作的时序特性,并对有外部事件参与、动作同时发生等复杂场景的推理提供了强有力的支持.ITL 用含时段变量的事件谓词描述事件发生,例如:事件谓词 $\forall e.ACTION(t,e)$ (事件谓词名用动作名的大写形式)表示动作 *action* 在时段 *t* 上发生.其中,*e* 为事件变量,可用函数 $Time(e)$ 获取事件发生的时段,该事件谓词可简写为 $\forall e.ACTION(e)$.通过 ITL 定义的 *Meet*、*Before* 和 *During* 等多种时段关系,刻画事件(或动作)发生的时段之间的关系.此外,ITL 还定义了一组公理模式用于推理和证明.下面给出本文将使用的一些公理模式.

① 同构公理模式(HOMogeneity Axiom Schema, HOA):

$$\forall x_i, t, t'. P(x_1, \dots, x_n, t) \wedge t' \sqsubseteq t \rightarrow P(x_1, \dots, x_n, t').$$

若谓词 $P(x_1, \dots, x_n, t)$ 在时段 *t* 为真,则对 *t* 的任意子时段 *t'*, $P(x_1, \dots, x_n, t')$ 也为真.

② 不重叠模式(DISJunct Axiom Schema, DISJ):

$$\forall t, t'. P(t) \wedge \neg P(t') \rightarrow t \not\subseteq t'$$

若 $P(t)$ 在时段 t 为真, $\neg P(t')$ 在时段 t' 为真, 则 t 和 t' 不重叠.

③ 离散变化公理模式 (Discrete Variation Axiom Schema, DVA):

$$\forall t \neg P(t) \rightarrow \exists t' \sqsubseteq t \wedge \neg P(t').$$

若谓词 $\neg P(t)$ 在时段 t 为真, 则存在 t 的某一子时段 t' , 使得 $\neg P(t')$ 为真.

④ 转换点公理模式 (Transition Point Axiom Schema, TRPT):

$$\forall t, t'. P(t) \wedge \neg P(t') \wedge t <: t' \rightarrow \exists T, T'. P(T) \wedge \neg P(T') \wedge T: T' \wedge t' \sqsubseteq T'$$

在任何相邻时段中, 如果 P 的真值发生变化, 则相邻时段中必然存在一个转换点.

⑤ 事件定义公理模式 (Event DEFINition Axiom Schema, EDEF):

$$\forall e. E(e) \wedge \Phi \rightarrow \psi,$$

其中 E 为事件谓词, Φ 和 ψ 不含事件变量的谓词. 通常用 Φ 表示事件 E 发生所需的前置条件, ψ 表示事件 E 完成后所满足的后置条件.

⑥ 事件产生公理模式 (Event GeNERation Axiom Schema, EGN):

$$\forall e. E(e) \wedge \Phi \rightarrow \exists e'. E'(e') \wedge \psi,$$

其中 E 和 E' 为事件谓词, Φ 和 Ψ 分别表示在 e 和 e' 上的约束. 该公理模式通常用于表示一个事件的产生引发其它事件的产生.

⑦ 属性强封闭公理模式 (Strong Closure on Property, SCP):

$$\forall t, t'. P(t) \wedge \neg P(t') \wedge t: t' \rightarrow$$

$$\exists e. (E_1(e) \vee E_2(e) \vee \dots) \wedge \text{time}(e): t'$$

其中, E_i 为事件谓词, 它可能影响 P 的真值. 该公理模式通常用于表示当 P 的真值发生变化时, 至少存在一个导致 P 真值变化的事件.

4.2 关注点横切关系的形式化描述

横切动作与动作之间的组合关系体现了关注点之间的横切关系. 具体而言, 这种组合关系定义了横切动作在满足其前置条件的情况下, 按照一定的时序关系横切指定的一个或多个动作, 横切动作完成后使得其后置条件得到满足, 其定义如下.

定义 1. 若 $\forall e. CM(t, e)$ 和 $\forall e'. M(t', e')$ 分别表示横切动作 cm 和动作 m 在时段 t 和 t' 上发生的事件谓词, 且 e 和 e' 分别为对应的事件变量, $\text{Time}(e)$ 和 $\text{Time}(e')$ 分别为 e, e' 发生的时段; 切点表达式 PCD 指定一组与 cm 组合的动作; pre 和 $post$ 分别为 cm

的前、后置条件; cm 发生的时段和与 cm 有组合关系的任一动作 m 发生时段之间的关系为 $crTemporal$, 则横切动作与动作之间的组合关系可用五元组 $Relation(cm, PCD, rtemporal, pre, post)$ 定义:

$$\begin{aligned} Relation(cm, PCD, rtemporal, pre, post) \equiv \\ \forall m, e, e', t_1, t_2. m \in matchPCD(PCD) \wedge pre(t_1) \wedge \\ t_1 \circ Time(e) \wedge post(t_2) \wedge Meets(Time(e), t_2) \wedge \\ crTemporal(Time(e), Time(e')). \end{aligned}$$

下面我们从匹配、时序关系及约束 3 个方面对 $Relation$ 进行详细阐述.

(1) 匹配

利用切点表达式 PCD 可描述横切动作 cm 与哪些动作之间存在组合关系, 而 $matchPCD(PCD)$ 函数则用于查找满足 PCD 表达式的动作, 返回符合 PCD 要求的动作集合. PCD 和 $matchPCD(PCD)$ 的定义分别如下.

定义 2.

$$\begin{aligned} PCD \equiv conName.motionName | conName.PRO | \\ conName.REQ | conName.* | \\ PCD_1 \wedge PCD_2 | \neg PCD. \end{aligned}$$

定义 3. $matchPCD(PCD)$ 函数定义为

- ① if $PCD = conName.motName$
 $matchPCD(PCD) \equiv \{a | a \in Motions \wedge$
 $concernName(a) = conName \wedge$
 $motionName(a) = motName\}.$
- ② if $PCD = conName.PRO$
 $matchPCD(PCD) \equiv \{a | a \in Motions \wedge$
 $concernName(a) = conName \wedge$
 $motionType(a) = "PRO"\}.$
- ③ if $PCD = conName.REQ$
 $matchPCD(PCD) \equiv$
 $\{a | a \in Motions \wedge concernName(a) =$
 $conName \wedge motionType(a) = "REQ"\}.$
- ④ if $PCD = conName.*.$
 $matchPCD(PCD) \equiv$
 $\{a | a \in Motions \wedge concernName(a) = conName\}.$
- ⑤ if $PCD = PCD_1 \wedge PCD_2$
 $matchPCD(PCD) \equiv$
 $matchPCD(PCD_1) \cap matchPCD(PCD_2).$
- ⑥ if $PCD = \neg PCD_1$
 $matchPCD(PCD) \equiv$
 $Motions - matchPCD(PCD_1).$

定义 2 中 6 种不同的 PCD 表达式所指定的动作

可由定义 3 的 $matchPCD$ 函数确定. 其中, $Motions$ 为具体应用系统的 SA-MDSoc 模型中所有动作构成的集合, $concernName(a)$, $motionName(a)$, $motionType(a)$ 分别表示获取动作 a 的关注点名称、动作名称和动作类型的函数. 动作类型有请求和提供两种类型, 分别用 REQ 和 PRO 表示. 下面以情形 ② 为例进行说明, 若 $PCD \equiv conName, PRO$, 则 $matchPCD$ 函数返回名称为 $conName$ 且类型为提供类型的动作, 其它情形可从相应的定义中直观看出其含义, 不再赘述.

(2) 时序关系

我们定义了 8 种不同的 $crTemporal$ 用于描述横切动作与动作发生的时序关系. 例如: 当 $crTemporal$ 为 $Meets$ 时, 即 $Meets(Time(e), Time(e'))$ 可定义为 $Time(e) : Time(e')$, 它表示横切动作 cm 发生的时段与动作 m 发生的时段相遇. 其它时序关系与 ITL 中同名时序关系符的定义也是类似, 具体见定义 4 的 ②~⑦.

定义 4. $crTemporal \equiv Meets | Before | During | Starts | Finishes | Overlaps | Concurrent | After$

$$\textcircled{1} Meets(Time(e), Time(e')) \equiv Time(e) : Time(e').$$

$$\textcircled{2} Before(Time(e), Time(e')) \equiv Time(e) \prec Time(e').$$

$$\textcircled{3} During(Time(e), Time(e')) \equiv \exists k, l. Time(e') = k + Time(e) + l.$$

$$\textcircled{4} Starts(Time(e), Time(e')) \equiv \exists k. Time(e') = Time(e) + k.$$

$$\textcircled{5} Finishes(Time(e), Time(e')) \equiv \exists k. Time(e') = k + Time(e).$$

$$\textcircled{6} Overlaps(Time(e), Time(e')) \equiv \exists k, l, m. (Time(e) = k + l) \wedge (Time(e') = l + m).$$

$$\textcircled{7} Concurrent(Time(e), Time(e')) \equiv Time(e) = Time(e').$$

$$\textcircled{8} After(Time(e), Time(e')) \equiv Time(e') \prec Time(e).$$

(3) 约束

组合关系中约束需要从两个方面进行描述. 一方面需要在传统一阶谓词中引入相应的时段变量使得横切动作的前、后置条件可在某时段上进行逻辑断言, 用 $pre(t_1)$ 和 $post(t_2)$ 分别定义横切动作的前、后置条件在时段 t_1 和 t_2 上的断言. 另一方面需要描述前、后置条件断言成立的时段与横切动作发生

的时段之间关系. 前置条件断言需要成立的时段 t_1 依赖于具体的横切动作, 它与横切动作发生的时段 $Time(e)$ 之间的关系用符号“ \circ ”表示, 最为常见的是 $Meets$ 或 $Finishes$ 关系. 横切动作发生的时段 $Time(e)$ 与后置条件断言需要成立的时段 t_2 之间的关系一定为 $Meets$ 关系.

4.3 时序冲突分析

多维模型是“1+X”风格的模型, X 个横切维上的横切关注点都可能会与主维上的主关注点之间存在着横切关系. 因而, 当描述这些横切关系的多个横切动作与相同的动作组合时, 可能引发横切动作之间的时序冲突, 从而得出相应的横切关注点之间存在时序冲突. 下面给出任意两个横切动作之间存在时序冲突的定义.

定义 5. 若任意两个横切动作和动作的组合关系分别为 $R_1(cm_1, PCD_1, crTemporal_1, pre_1, post_1)$ 和 $R_2(cm_2, PCD_2, crTemporal_2, pre_2, post_2)$. 设 $\forall e_1. CM_1(t_1, e_1)$ 和 $\forall e. CM_2(t_2, e_2)$ 分别为横切动作 cm_1, cm_2 在时段 t_1 和 t_2 发生的事件谓词. 当 R_1 和 R_2 满足式(1)和(2)时, 若存在某个谓词 $p(t)$ 及其否定 $\neg p(t)$ 在 t 上同时保持为真, 则称横切动作 cm_1 和 cm_2 在动作 m 上存在时序冲突.

$$cm_1 \neq cm_2 \wedge \exists m. m \in matchPCD(PCD_1) \wedge m \in matchPCD(PCD_2) \quad (1)$$

$$\exists t. (t \sqsubseteq t_1) \wedge (t \sqsubseteq t_2) \quad (2)$$

在上述定义中, 式(1)保证两个不同的横切动作 cm_1 和 cm_2 与相同的动作 m 进行组合, 式(2)则保证 cm_1 和 cm_2 发生的时段上有相同的子时段.

基于上述定义 1~5, 运用本文所提出的方法分析横切关注点之间是否存在时序冲突的基本步骤如下: ① 根据具体应用的需求规约和体系结构设计要求, 按照定义 1 所提供的描述方式, 基于 ITL 对横切关注点与主关注点之间的横切关系进行形式化描述; ② 在所描述的横切关系中, 查找满足定义 5 中式(1)和(2)的任意两个横切动作和动作之间的组合关系; ③ 运用 ITL 提供的公理模式进行推理, 确定横切动作之间是否存在时序冲突, 进而分析出横切关注点之间是否存在时序冲突. 通过对分析出的横切关注点之间时序冲突关系予以标识, 为后续的冲突解决提供有用线索.

5 案例研究

根据网上拍卖系统的需求规约和体系结构设计

要求,通过对通用的 SA-MDSoc 模型和电子拍卖领域的多维分离模型中的关注点及其关系进行选择 and 精化后,我们建立了网上拍卖系统的 SA-MDSoc 模型.下面仅以网上拍卖系统中竞价场景为例,给出体系结构关注点分析方法的实施过程.竞价场景的描述如下:买家发送竞价之前,系统需要对买家身份进行验证.只有合法买家才可以参与拍卖并发送竞价消息,系统需要对竞价消息进行加密.

图 2 是网上拍卖系统的 SA-MDSoc 模型中的竞价场景示意图.它展示:

① 主关注点 buyer 通过请求类型的动作 reqBid 发送竞价,主关注点 Auction 通过提供类型的动作 proBid 接收竞价.

② 横切关注点 Authentication 的横切动作 authen 和横切关注点 Confidentiality 的横切动作 encrypt 分别实现对买家身份的认证和对竞价信息进行加密.

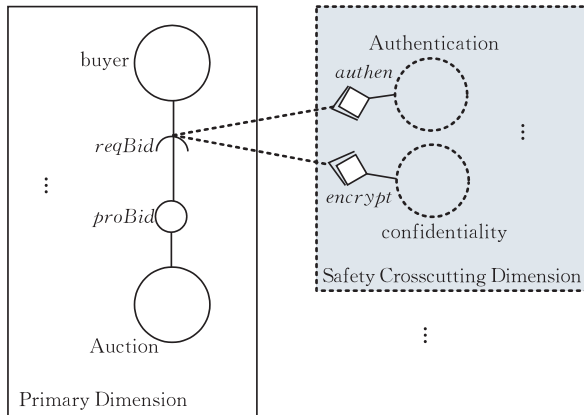


图 2 竞价场景示意图

(1) 竞价场景的形式化描述

横切关注点 Authentication 与主关注点 Auction 之间的横切关系由横切动作 authen 与 PCD₁ 所指示的动作之间的组合关系 R₁ 描述.横切关注点 Confidentiality 与主关注点 Auction 之间的横切关系由横切动作 encrypt 与 PCD₂ 所指示的动作之间的组合关系 R₂ 描述.按照定义 1 给出 R₁ 和 R₂ 的 ITL 描述如下:

① R₁ 的描述

用 $\forall e_1. AUTHEN(t_1, e_1)$ 和 $\forall e'. M(t', e')$ 分别表示横切动作 authen 和动作 m 在时段 t₁ 和 t' 上发生的事件谓词; authen 需要组合的动作由 matchPCD(buyer.reqBid) 描述; authen 的前、后置条件分别用谓词 $\neg authened(m)$ 和 authened(m) 表示,其中前者表示动作 m 未认证的断言,后者表示动作 m 已

认证的断言.则 R₁ 描述如下:

$$R_1(authen, buyer.reqBid, Before, \neg authened, authened) \equiv \forall m, e_1, e', t'_1, t''_1. m \in matchPCD(buyer.reqBid) \wedge \neg authened(m, t'_1) \wedge Finishes(Time(e_1), t'_1) \wedge authened(m, t''_1) \wedge Meets(Time(e_1), t'_1) \wedge Before(Time(e_1), Time(e'))$$

② R₂ 的描述

用 $\forall e_2. ENCRYPT(t_2, e_1)$ 和 $\forall e'. M(t', e')$ 分别表示横切动作 encrypt 和动作 m 在时段 t₂, t' 上发生的事件谓词; encrypt 需要组合的动作由 matchPCD(buyer.reqBid) 描述; encrypt 的前、后置条件分别用谓词 authened(m) \wedge $\neg encrypt(m)$ 和 authened(m) \wedge encrypt(m) 表示,其中前者表示动作 m 已认证且未加密的断言,后者表示动作 m 已认证且已加密的断言. R₂ 描述如下:

$$R_2(encrypt, buyer.reqBid, Meets, authened \wedge \neg encrypted, authened \wedge encrypted) \equiv \forall m, e_2, e', t'_2, t''_2. m \in matchPCD(buyer.reqBid) \wedge authened(m, t'_2) \wedge \neg encrypted(m, t'_2) \wedge Finishes(Time(e_2), t'_2) \wedge authened(m, t''_2) \wedge encrypted(m, t''_2) \wedge Meets(Time(e_2), t''_2) \wedge Meets(Time(e_2), Time(e'))$$

(2) 竞价场景的时序冲突分析

由 R₁ 和 R₂ 的形式化描述可知:

① $authen \neq encrypt \wedge reqBid \in matchPCD(PCD_1) \wedge reqBid \in matchPCD(PCD_2)$.

② 由于 $Before(Time(e_1), Time(e'))$ 且 $Meets(Time(e_2), Time(e'))$, 因此 authen 发生的事件 e₁ 和 encrypt 发生的事件 e₂ 的时段关系有 2 种情形, 分别如图 3(a)、(b) 所示.

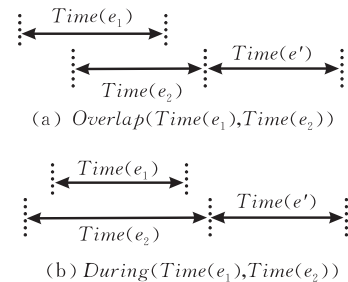
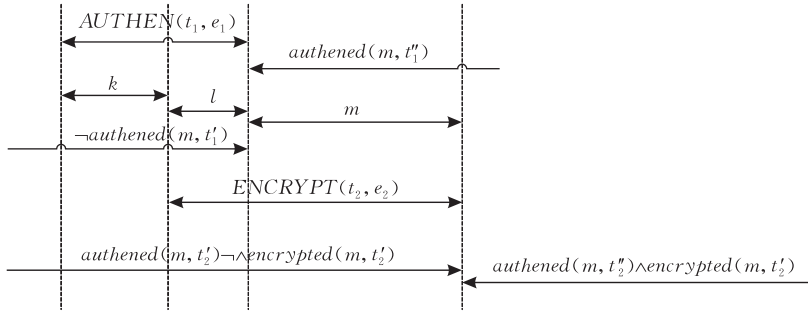


图 3 事件 e₁ 和 e₂ 可能的时段关系

图 3 中的两种情形都存在同为 Time(e₁) 和 Time(e₂) 的子时段, 因而它们均满足定义 5 中的两个条件, 需要进一步推理分析.

情形 1. $Overlap(Time(e_1), Time(e_2))$.

$Overlap(Time(e_1), Time(e_2))$ 情形下, authen 和 encrypt 发生的事件定义如图 4 所示.

图 4 情形 1 下 *authen* 和 *encrypt* 发生的事件定义

两次运用事件定义公理模式 EDEF 得

$$\text{EDEF1: } \forall e_1. \text{AUTHEN}(e_1) \wedge \neg \text{authenticated}(m, t'_1) \wedge \text{Finishes}(\text{Time}(e_1), t'_1) \rightarrow \text{authenticated}(t'_1) \wedge \text{Meets}(t'_1, t'_1),$$

$$\text{EDEF2: } \forall e_2. \text{ENCRYPT}(t_2, e_2) \wedge \text{authenticated}(m, t'_2) \wedge \neg \text{encrypted}(m, t'_2) \wedge \text{Finishes}(\text{Time}(e_2), t'_2) \rightarrow \text{authenticated}(m, t'_2) \wedge \text{encrypted}(m, t'_2) \wedge \text{Meets}(\text{Time}(e_2), t'_2).$$

情形 1 下的推理如表 1 所示. 由步骤 2 和 12 可知横切动作 *authen* 和 *encript* 在动作 *reqBid* 上存在时序冲突.

表 1 情形 1 下的推理表

步骤	结论	依据
1	$\exists k, l, m. \text{Time}(e_1) = k + l \wedge \text{Time}(e_2) = l + m$	$\text{Overlap}(\text{Time}(e_1), \text{Time}(e_2))$ (前提)
2	$l \sqsubseteq \text{Time}(e_1) \wedge l \sqsubseteq \text{Time}(e_2)$	步骤 1 和 \sqsubseteq 关系的定义
3	$\exists p. t'_1 = p + \text{Time}(e_1)$	$\text{Finishes}(\text{Time}(e_1), t'_1)$ (前提)
4	$\text{Time}(e_1) \sqsubseteq t'_1$	步骤 3 和 \sqsubseteq 关系的定义
5	$\exists q. t'_2 = q + \text{Time}(e_2)$	$\text{Finishes}(\text{Time}(e_2), t'_2)$ (前提)
6	$\text{Time}(e_2) \sqsubseteq t'_2$	步骤 5 和 \sqsubseteq 关系的定义
7	$l \sqsubseteq t'_1 \wedge l \sqsubseteq t'_2$	步骤 2, 4, 6
8	$\text{authenticated}(m, t'_2)$	EDEF2
9	$\text{authenticated}(m, l)$	步骤 7, 8 和 HOA 公理模式
10	$\neg \text{authenticated}(m, t'_1)$	EDEF1
11	$\neg \text{authenticated}(m, l)$	步骤 7, 10 和 DVA 公理模式及 <i>authenticated</i> 断言的同构性
12	$\exists l. \text{authenticated}(m, l) \wedge \neg \text{authenticated}(m, l)$	步骤 7, 10 和 11

情形 2. $\text{During}(\text{Time}(e_1), \text{Time}(e_2)).$

横切动作 *authen* 和 *encript* 也在动作 *reqBid* 上存在时序冲突, 其推理过程与情形 1 类似, 限于篇幅, 在此略去.

综上所述, 横切动作认证和加密在发送竞价动作上存在时序冲突, 因而横切关注点 Authentication 和 Confidentiality 存在时序冲突, 需要在多维分离模型中标识出该时序冲突. 对于案例中的时序冲突, 可通过在横切关注点之间的时序关系中定义横切动作认

证发生在横切动作加密之前予以解决. 即 $\text{Before}(\text{Authentication.authen}, \text{Confidentiality.encrypt})$.

从案例研究可以看出: 在体系结构设计的早期, 设计人员按照本文所提出的分析方法对具体应用系统的 SA-MDSoc 模型中横切关注点与主关注点之间的横切关系进行形式化描述和分析, 可以有效地发现横切关注点之间时序冲突. 此外, 通过对分析出的时序冲突加以标识或进行相应的解决, 能为后续设计无横切关注点时序冲突的体系结构模型提供有力支持, 进而能帮助提高体系结构设计质量.

6 结论和未来工作

在体系结构关注点多维分离模型的基础上, 本文提出了一种体系结构关注点分析方法. 基于时段时序逻辑, 该方法给出了关注点之间横切关系的形式化描述和横切关注点之间时序冲突的形式定义, 并在此基础上给出了时序冲突分析的步骤. 最后结合网上拍卖系统的案例, 给出了该方法的实施过程. 该方法的建立能在体系结构设计的早期发现潜在设计错误, 为提高体系结构设计质量提供支持.

未来的研究工作是进一步完善本文中的 SA-MDSoc 模型和形式化分析方法, 以支持各种关注点之间关系的分析, 并开发相应的支撑工具, 形成一套更加完整的、更具有实际应用价值的体系结构关注点分析方法.

参 考 文 献

- [1] Kiczales G, Lamping J, Mendhekar A et al. Aspect-oriented programming//Proceedings of the 11th European Conference on Object-Oriented Programming. Finland, 1997: 220-242

- [2] Sutton S M, Rouvellou I. Modeling of software concerns in cosmos//Proceedings of the 1st International Conference on Aspect-Oriented Software Development. Enschede, The Netherlands, 2002: 127-133
- [3] Tarr P, Ossher H, Harrison W et al. *N* degrees of separation: Multi-dimensional separation of concerns//Proceedings of the 21st International Conference on Software Engineering. Los Angeles, 1999: 107-119
- [4] Moreira A, Araújo J, Rashid A. A concern-oriented requirements engineering model//Proceedings of the 17th Conference on Advanced Information Systems Engineering. Porto, Portugal, 2005: 293-308
- [5] Chitchyan R, Pinto M, Rashid A, Fuentes L. COMPASS: Composition-centric mapping of aspectual requirements to architecture//Transactions on Aspect-Oriented Software Development IV. Berlin: Springer, 2007: 3-53
- [6] Cuesta C E, María del Pilar Romay, Pablo de la Fuente, Manuel Barrio-Solórzano. Architectural aspects of architectural aspects//Proceedings of the European Workshop on Software Architecture. Pisa, Italy, 2005: 247-262
- [7] Pinto M, G'amez N, Fuentes L. Towards the architectural definition of the health watcher system with AO-ADL//Proceedings of the 29th International Conference on Software Engineering Workshops. Minneapolis, MN, 2007: 93-101
- [8] García A, Chavez C, Batista T et al. On the modular representation of architectural aspects//Proceedings of the 3rd European Workshop on Software Architecture. Nantes, France, 2006: 82-97
- [9] Suvéé D, Fraïne B D, Vanderperren W. FuseJ: An architectural description language for unifying aspects and components//Proceedings of the Software Engineering Properties of Languages for Aspect Technologies associated with AOSD'05. Chicago, USA, 2005
- [10] Pinto M, Fuentes L, Troya J M. A dynamic component and aspect-oriented platform. The Computer Journal, 2005, 48 (4): 401-420
- [11] Navasa A, Peraz M A, Murillo J M. Aspect modelling at architecture design//Proceedings of the 2nd Europe Workshop on Software Architecture. Pisa, Italy, 2005: 41-58
- [12] Pérez J, Ramos I, Jaén J, Letelier P, Navarro E. PRISMA: Towards quality, aspect oriented and dynamic software architectures//Proceedings of the 3rd International Conference on Quality Software. Dallas Texas, USA, 2003
- [13] Dai L, Cooper K, Wong W E. Modeling and analysis of performance aspects for software architecture: A UML-based approach. Software Engineering and Knowledge Engineering, 2006, 16(3): 347-378
- [14] Xu L, Ziv H, Alspaugh T A et al. An architectural pattern for non-functional dependability requirements//Proceedings of the 2005 Workshop on Architecting Dependable Systems. St. Louis, Missouri, 2005: 1-6
- [15] Bagheri H. Injecting security as aspectable NFR into software architecture//Proceedings of the 14th Asia-Pacific Software Engineering Conference. Nagoya, Japan, 2007: 310-317
- [16] Zhang Lin-Lin, Ying Shi, Ni You-Cong et al. An approach for multi-dimensional separation concern at architecture level//Proceedings of the 2008 Workshop on Power Electronics and Intelligent Transportation System. Guangzhou, China, 2008: 541-545
- [17] Zhang Lin-Lin, Ying Shi, Ni You-Cong, Zhao Kai, Ye Peng. A generic model for multi-dimensional separation of concerns at architecture level//Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing. Dalian, China, 2008
- [18] Chung L, Nixon B A, Yu E et al. Non-Functional Requirements in Software Engineering. Norwell: Kluwer Academic Publishers, 2000
- [19] Eeles P. Capturing architectural requirements. IBM, 2001
- [20] Allen J F, Ferguson G. Actions and events in interval temporal logic. Journal of Logic and Computation, 1994, 4(5): 531-579
- [21] Allen J F. Maintaining knowledge about temporal intervals. Communications of the ACM, 1983, 26(11): 832-843
- [22] Allen J F, Hayes P J. Moments and points in an interval-based temporal logic. Computational Intelligence, 1989, 5(3): 225-238



ZHANG Lin-Lin, born in 1974, Ph. D., lecturer. Her current research interests include software architecture and aspect-oriented software development.

YING Shi, born in 1965, professor, Ph. D. supervisor. His current research interests include object-oriented software engineering methods, component-based software engineering methods, software architecture and pattern, software reus-

ability and interoperability, semantic Web technologies.

NI You-Cong, born in 1976, Ph. D. candidate, lecturer. His current research interests include software architecture and aspect-oriented software development.

ZHAO Kai, born in 1976, Ph. D. candidate, lecturer. His current research interests include service-oriented architecture and semantic Web technologies.

WEN Jing, born in 1982, Ph. D. candidate. Her current research interests include software architecture and aspect-oriented software development.

Background

Those concerns that represent non-functional or quality requirements mostly show crosscutting feature, which can be addressed by the methodology and technology about Aspect. At present, there are many approaches for aspect-oriented software architecture design. However in these approaches the architectural concerns are not identified and organized over multi-dimension in order that it is difficult to trace architectural concerns at requirement analysis phase and coding phase. Furthermore in these approaches the analysis on the temporal relationships between crosscutting concerns is seldom involved, which leads to difficultly discover temporal conflict. In order to derivate high-quality software architecture, it is necessary to model variety of concerns and theirs relationships at software architecture level.

Moreira et al. have proposed CORE(Concerns Oriented Requirement Engineering), which emphasizes the analysis on the influence between requirement concerns, but lacks the analysis on the temporal relationship between concerns. In addition some approaches such as DAOP-ADL, Navasa, PRISMA, AO-ADL and Aspectual ACME have improvement on

modeling aspect-oriented software architecture. However in these approaches not only architectural crosscutting concerns have not been effectively organized and managed but also the analysis on temporal conflict between crosscutting concerns has not been involved.

In the paper, the authors consider the architectural concerns and theirs relationships. And then the authors propose an analysis approach for software architectural concerns based on model of multi-dimensional separating of concern at software architectural level. The crosscutting relationship between architectural concerns is emphatically researched in this approach. The authors use ITL(Interval Temporal Logic) to describe and analyze the crosscutting relationship between architectural concerns in order to discover the temporal conflict between the crosscutting concerns. The approach contributes to improvement of software architecture design quality.

This work is supported by the National Natural Science Foundation of China under grant No. 60773006 and the Doctoral Fund of Ministry of Education of China under grant No. 20060486045.