

面向方面业务过程建模的正确性控制与检测

张璇^{1),2)} 王旭³⁾ 李彤^{1),2)} 陈清毅^{1),2)} 刘俊晖^{1),2)}

¹⁾(云南大学软件学院 昆明 650091)

²⁾(云南省软件工程重点实验室 昆明 650091)

³⁾(云南大学经济学院 昆明 650091)

摘 要 在组织机构不断将其业务转向面向过程模式从而获得低成本、高效率、高服务质量与高生产力的同时,积累了成百上千的业务过程模型. 为了支持这些大批业务过程柔性建模并提高业务过程模型可重用性,采用关注点分离思想,将实现不同功能的活动按照横切关注点活动和基本业务活动实施分离,提出面向方面业务过程建模框架. 提供业务过程可重用建模方法的同时,灵活可控地实现过程模型正确性保障与检测. 在面向方面业务过程建模框架中,首先定义面向方面业务过程元模型,将横切关注点活动封装为横切方面,将基本业务活动建模为基本过程. 然后,按照业务过程模型正确性需求,基于面向方面编织特征,定义面向方面业务过程模型正确性,设计方面间编织正确性准则和方面织入基本过程正确性准则,为面向方面业务过程建模提供预防错误与异常建模的方法,并提出过程建模正确性检测方法,在建模过程中保证无错误与异常. 在此基础上,提出面向方面业务过程建模方法,证明过程模型的结构正确性、性质合理性和行为一致性,并设计实现建模辅助工具 TPAT(Trustworthy business Process Aided Tool),为面向方面业务过程建模提供技术支持. 最后,通过在两个银行业务案例分析中使用所提出的理论、方法和技术,并进行性能评估,说明所提出的理论和方法是可行的,开发的辅助工具是有效的. 面向方面方法可以为业务过程建模提供灵活可重用的建模方法,通过使用建模正确性准则并在建模过程中监控异常与错误,保证业务过程模型正确性,支持组织机构获得更好的过程支持和日益变更环境的适应性.

关键词 面向方面建模;业务过程建模;Petri 网;正确性;性能

中图法分类号 TP311 **DOI号** 10.11897/SP.J.1016.2018.00521

Correctness Control and Detection in Aspect-Oriented Business Process Modeling

ZHANG Xuan^{1),2)} WANG Xu³⁾ LI Tong^{1),2)} CHEN Qing-Yi^{1),2)} LIU Jun-Hui^{1),2)}

¹⁾(School of Software, Yunnan University, Kunming 650091)

²⁾(Key Laboratory of Software Engineering of Yunnan, Kunming 650091)

³⁾(School of Economics, Yunnan University, Kunming 650091)

Abstract Organizations increasingly work in a process-oriented manner, which offer promising perspectives for reducing costs and flow times, and increasing quality of service and productivity. As a consequence, single organizations may be dealing with collections of hundreds or thousands business process models. In order to support this abundance of business process modeling flexibility and improve business process models reusability, separation of concerns is used to separate the crosscutting activities and basic activities in accordance with the different functions of

收稿日期:2016-04-15;在线出版日期:2017-02-22. 本课题得到国家自然科学基金(61502413,61262025,61379032,61662085)、云南省应用基础研究计划面上项目(2016FB106)、云南省教育厅科学研究基金重点项目(2015Z020,2013A056)、云南省软件工程重点实验室开放基金面上项目(2015SE202)、云南省创新团队“数据驱动的软件工程创新团队”项目、云南大学高水平创新团队“软件工程创新团队”专项项目、云南大学“中青年骨干教师培养计划”专项项目、云南大学人文社科项目基金(13YNUHSS007)资助. 张璇,女,1978年生,博士,副教授,中国计算机学会(CCF)会员,主要研究方向为业务过程、软件过程、需求工程. E-mail: zhxuan@ynu.edu.cn. 王旭,男,1976年生,博士,副教授,主要研究方向为业务过程、金融风险. 李彤,男,1963年生,博士,教授,博士生导师,主要研究领域为软件过程、形式化方法. 陈清毅,男,1981年生,博士,讲师,主要研究方向为业务过程、云资源调度. 刘俊晖,男,1980年生,博士,讲师,主要研究方向为业务过程、软件工程.

these activities, and an aspect-oriented business process modeling framework is proposed. This framework provides a reusable process modeling method and also provided a flexible and controllable way to assure correctness of the whole business process models. In the framework, an aspect-oriented business process meta-model is defined first. The crosscutting concerns activities are encapsulated as crosscutting aspects, while the basic business activities are modeled in basic processes. Then, according to the correctness requirements of the business process model, based on the weaving characteristics of aspect-oriented approach, weaving correctness is defined. Weaving correctness criteria between multi-aspects and between aspects and basic processes are designed. Errors or mistakes of aspect-oriented business process modeling are prevented during the procedure of modeling. In addition, a detection method for errors or mistakes in process modeling is proposed to ensure error-free business process modeling. Based on the meta-model and the correctness criteria and the detection method, an aspect-oriented business process modeling approach is proposed. The corresponding correctness of the structures, properties and behaviors are proved. Thirdly, an aided tool TPAT (Trustworthy business Process Aided Tool) is developed to provide modeling support. Finally, the theory, the approach and the tool were used in the cases of two banking business processes. Through the case study and performance evaluation, the theory and the approach are proved to be feasible and the tool is proved to be effective. Aspect-oriented modeling is suitable for providing flexible and reusable business process modeling. By adopting the correctness criteria and detecting errors and mistakes during modeling, aspect-oriented business process modeling approach assures the correctness of process models and provides better process-oriented support and adaptability to the ever-change environment.

Keywords aspect-oriented modeling; business process modeling; Petri net; correctness; performance

1 引言

业务过程模型是理解与分析业务过程的重要工具,伴随着组织机构不断增加的业务工作,业务过程模型逐渐积累下来,大批大型组织机构已经设计并保存了成百上千的业务过程模型,例如:Suncorp 的过程模型库中就包含了多于 6000 个的保险相关过程模型,BIT 过程库包含 735 个过程模型,IBM 应用架构包含 250 个过程模型,荷兰市政局存储了大约 600 个参考过程模型^[1].在这些过程模型中,大部分的模型本质上是相似的,即这些相似的模型有可重用的活动、子过程或者组织实体,而这些可重用部分没有被有效的重用,导致更多相似过程模型被新建并存储下来,面对如此庞大的过程模型库,如何有效的管理与维护这些大量被保存的业务过程模型已经成为一个新的研究方向^[1].

导致相似模型无法重用的一个重要原因是业务过程模型中的基本活动和横切活动交织建模,基本活动是完成业务目标而强制指定用于体现业务功能

的活动,横切活动是针对特定应用领域定制用于保证基本活动安全可追踪的活动,例如:审核、签署、控制、归档等活动通常是对基本活动的约束,用于保证基本活动安全且可追踪.将横切活动与基本活动合成,就实现一个特定应用领域的业务过程^[2],但如果在建模阶段将这两类活动不加区分地进行交织建模,则难于实现业务过程模型的重用,也无法针对不同业务需求定制业务过程.另外,在实际的应用领域中,预先设计好完整的业务过程模型是不可能的,即便能够设计好完整的业务过程模型,外部环境和业务需求的变化也迫使业务过程需要随之变更,因此,我们只能在建模时完成大部分的过程模型,其余不确定的过程只能在后期明确以后逐步补充至已有业务过程中^[3].在对过程模型实施如此频繁的修改行为时,如果两类活动“纠缠”(tangling)在一起,对其中任何活动的改变都有可能影响其他活动甚至破坏整个业务过程,而相同横切活动“分散”(Scattering)编织在基本过程的不同位置,又难于保证这些横切活动的一致性,这类“纠缠”和“分散”问题必然导致业务过程难以维护、缺乏柔性且不利于重用.关注点

分离是一个解决上述模型重用及实现柔性建模的有效方法,可以有效解决“分散”与“纠缠”问题,有利于降低过程模型复杂度,同时,也利于过程模型的理解、维护、改变和重用。

另外,在过程模型投入使用之前,必须保证模型是正确的。针对模型正确性问题,已有的研究方法主要在业务过程建模完成后对过程模型进行验证^[4-8],尽管存在多种不同的过程模型,通常的验证都是从控制流和数据流的角度来定义过程模型的正确性。过程模型的控制流正确性主要关注过程模型活动间控制依赖关系的正确性,即保证无死锁、弱终止等;过程模型的数据流正确性则关注过程模型中活动之间数据依赖的可满足性,例如,需要避免过程模型中某一活动的输入没有事先定义的情形^[9]。另外一个基本的验证目标是过程模型的合理性,合理性来源于工作流领域的工作流网 WF-nets^[10]定义,之后应用到了普遍业务过程建模领域,用于描述过程模型一定存在一个变迁序列能够保证从起始状态到达唯一的终止状态且不存在死变迁^[6]。而针对业务过程的变更正确性,主要的验证方法是针对变更过程片段研究变更前后过程模型的兼容性问题^[11],或者设计变更模式^[12]指导过程模型的变更。本文使用面向方面方法,在业务过程建模之前定义建模的正确性准则,在建模过程中控制并检测方面间编织正确性和方面织入基本过程正确性,保证面向方面业务过程建模得到的模型是正确的。

为实现上述目标,在第2节分析并对比相关工作的基础上,在第3节提出面向方面业务过程建模框架,首先定义面向方面业务过程元模型,形式化定义业务过程中所有活动和活动间的结构及行为,同时,定义面向方面扩展及约束方法;在此基础上,在第4节给出面向方面业务过程建模正确性定义、条件及检测方法。使用面向方面业务过程元模型,基于建模正确性定义,在第5节提出面向方面业务过程建模方法,实现方面间编织和方面织入基本过程,并对建模正确性进行证明,设计并实现建模辅助工具。最后,通过第6节的案例分析和性能评估说明所提出的理论和方法是可行的,并在第7节中总结本文的贡献,对未来进一步的研究工作进行展望。

2 相关工作

本文研究面向方面业务过程建模,包括过程建模方法和过程模型的正确性,下面就相关建模与验

证领域的研究现状进行总结并与本文工作进行比较。

基于对已有相关文献的调研,将面向方面思想运用至业务过程管理的研究大部分仅提出了初步研究思路,并未形成一套系统的方法。最早引入面向方面思想的是 Odgers 和 Thompson,他们将面向方面编程 AOP (Aspect-Oriented Programming) 思想引入业务过程管理,提出面向方面业务过程工程 ASOPE (Aspect-Oriented Process Engineering) 概念,ASOPE 的提出是为了将不同的需求引入业务过程,其中,通用作过程模式是为达成通用目标而强制指定的必要的过程步骤,而过程方面则是针对特定领域指定定制的过程步骤,当过程方面织入通用过程后,实现了一个特定领域的业务过程^[2]。之后, Bachmendo 和 Unland 将面向方面思想引入工作流演化,基于面向方面提出工作流的控制过程演化方法,支持工作流活动关系的变更^[13]。Cappelli 等人^[14]将业务过程管理核心过程与横切过程分离,从上层架构角度提出面向方面业务过程管理 AO-BPM (Aspect-Oriented Business Process Management) 概念,用于增强业务过程建模方法的易用性和可理解性。在此基础上, Santos 和 Cappelli 等人针对面向方面业务过程建模的 5 个问题,包括了方面识别、建模元素定义、模块化层次、方面到组织机构角色的映射和方面模型生成与可视化,提出解决思路^[15]。之后, Leite 和 Cappelli 等人^[16-17]继续基于 Cappelli 提出的面向方面业务过程建模方法 AO-BPM,使用 i* 模型描述策略角色的方法,建立业务过程与过程所有者以及方面过程与方面所有者之间的映射关系^[18]。Pourshahid 和 Amyot 等人^[19-21]基于 ITU-T Z. 151 标准化的用户需求符号 URN (The User Requirements Notation), 提出面向目标和方面的业务过程工程概念,目标是要实现业务过程的改进和动态适应性。上述 Odgers、Bachmendo、Cappelli、Pourshahid 和 Amyot 等人将研究视角定位于高层架构角度,仅仅提出面向方面在业务过程应用的高层概念,并没有具体提出建模方法。从具体实现层面研究面向方面业务过程管理的只有 Charfi、Witteborg、Park 和 Jalali 等人, Charfi 和 Mezini^[22-24]提出基于 BPEL 的面向方面工作流语言 AO4BPEL (Aspect-Oriented for Business Process Execution Language) 支持 Web 服务合成。在此基础上, Charfi 和 Müller 等人^[25]提出面向方面业务过程建模符号 AO4BPMN (Aspect-Oriented for Business Process Model and Notation)。之后, Witteborg 和 Charfi 等人^[26]改进 AO4BPMN 使其

支持 BPMN2.0, 并开发切点语言和编织器支持面向方面业务过程建模. Park 等人^[27]提出业务规则应该作为一个重要的横切关注点而独立于业务过程实例, 因此, 他们提出一个基于规则的 AOP 框架实现业务规则的有效分离表示与度量, 并使用两个开源产品 Mandarax 规则引擎和 Bexee EPEL 引擎实现业务规则方面的动态编织与执行. Jalali 研究了面向方面业务过程建模的静态编织方法^[28]、动态编织方法^[29]和基于声明规则的混合编织方法(支持定义强制与可选横切方面)^[30]. Charfi、Witteborg、Park 和 Jalali 都提出了具体的面向方面业务过程建模语言或者方法, 但 Charfi、Witteborg 和 Park 并未研究业务过程模型的正确性问题, Jalali 研究了过程模型的合理性, 但他们都没有考虑使用面向方面方法有可能因为方面织入而产生错误的问题. 面向方面方法推广的最大阻碍在于方面与基础模型或者方面之间产生不需要的相互作用, 导致方面编织后产生异常和错误^[31], 而业务过程模型的正确性是业务过程模型的一项重要指标. 因此, 将面向方面方法引入业务过程建模必须研究方面编织有可能引入错误的问题.

为了保证面向方面业务过程建模的正确性并解决方面编织有可能引入错误的问题, 本文使用形式化语言 Petri 网作为建模语言. 在业务过程领域, 可用的建模语言包括 Petri 网、BPMN、UML 和 EPCs, 其中 Petri 网提供了严格形式化的语言基础, 同时提供可视化表达方式, 可以准确描述业务过程模型并支持过程模型验证. 将面向方面方法应用于 Petri 网语言也有相关研究, 基于 Petri 网的面向方面建模最早是由 Roubtsova 和 Aksit 于 2005 年提出, 他们基于经典 Petri 网将方面 Petri 网定义为一个包含基本网、方面网和编织表达式的三元组, 并定义连接点模型, 用于描述在庫所和变迁上的静态编织^[32]. 之后, Xu 和 Nygard^[33]于 2006 年在其文章中提出基于 AOP 思想的面向方面谓词/变迁网, 他们借用 AspectJ 的思想, 定义封装了切点、通知和声明(introduction)的方面模型, 其中: 声明网建模通知的功能, 连接点是基本网的变迁、谓词和弧, 通知网定义连接点上的切点操作模式和相对位置(before、after 或 around). 2008 年, Guan 等人^[34]延续 Xu 和 Nygard 的面向方面谓词/变迁网提出面向方面库所/变迁网, 其定义中去除了 AspectJ 的声明定义, 方面网仅封装了通知和切点, 并且定义了编织网用于描述方面网编织入基本网后得到的面向方面库所/变迁网, 在此基础上, 他们重点研究了方面间的

冲突问题及解决方法. 2012 年, Molderez 通过定义编织器、连接点模型、切点语言、通知语言和合成机制, 非形式化地将面向方面特征加入到 Petri 网中^[35]. 上述 Roubtsova、Xu 和 Molderez 等人的研究工作都是仅仅将面向方面 Petri 网作为建模语言应用于某一领域, 并未探讨模型正确性问题, Guan 针对共享连接点的方面间干扰问题进行了研究, 但没有考虑方面与基本模型间的干扰问题.

另外, 为满足面向服务合成的松耦合及需求易变特征, 面向方面方法也被用于扩展过程语言, 支持服务合成过程. 在这一领域, Sun 等人^[36]将服务合成事务策略定义为方面并转换为标准 BPEL(Business Process Execution Language)元素, 在编织入基本 BPEL 过程后实现松耦合扩展事务的 BPEL 过程. 与此类似, Yu 等人^[37]针对 WS-BPEL(Web Services Business Process Execution Language), 使用面向方面方法将易变模型建模为决策逻辑方面, 并开发面向方面的模型驱动平台, 支持面向方面 web 服务过程建模. 另外, 为支持软件过程运行时修改, 面向方面方法也成为支持软件过程可修改的一个方法^[38].

上述研究工作将面向方面方法引入业务过程建模, 做出了非常有意义的工作. 本文使用 Petri 网作为建模语言, 并使用面向方面方法扩展 Petri 网, 实现业务过程中横切关注点的分离, 使业务过程结构清晰易于理解, 减少业务过程模型复杂度, 有利于业务过程重用, 提升过程模型的可维护性, 重点解决了方面编织的正确性问题, 保证面向方面业务过程建模方法的有效性.

3 面向方面业务过程建模框架

业务过程建模与验证是业务过程管理的核心任务^[6]. 业务过程建模是为实现特定业务目标而定义一个活动集合以及活动之间的依赖关系, 用以描述业务过程实例执行的所有情况, 是信息系统配置的基础, 也是分析、理解和改进过程的基础. 业务过程验证则关注业务过程模型的正确性, 正确性的满足是过程模型符合业务过程预期目标的重要基础. 基于此两项业务过程管理的核心任务, 本文提出如图 1 所示的面向方面业务过程建模框架.

面向方面业务过程建模框架的第一步是定义面向方面业务过程元模型, 此元模型是定义面向方面业务过程模型的形式化工具, 元模型不仅形式化地定义了业务过程中所有活动和活动间的结构及行

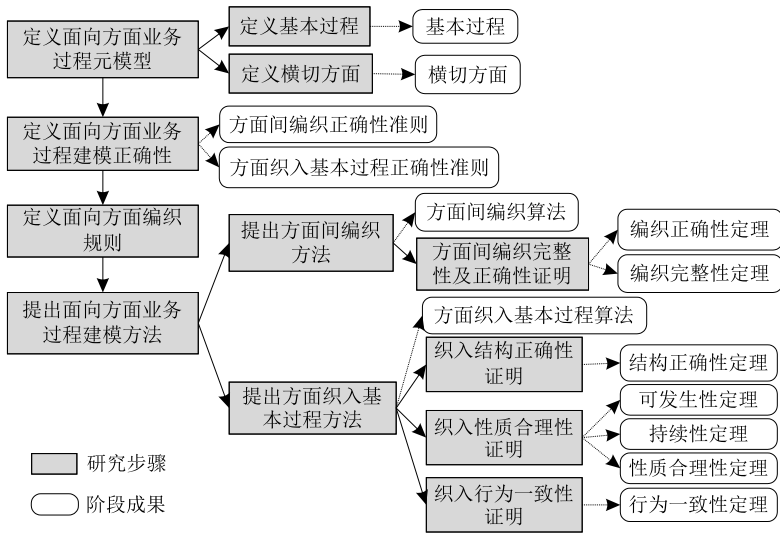


图 1 面向方面业务过程建模框架

为,还体现了面向方面扩展及约束方法,是面向方面业务过程建模的工具.下面,使用 Petri 网作为过程建模语言,定义面向方面业务过程建模的基本过程.基本过程由业务过程必须定义的核心活动组成,这些活动定义为 Petri 网的变迁,活动执行条件定义为 Petri 网的库所,库所与变迁之间的弧规定了条件与活动之间的关系,过程的执行状态则用 Petri 网的格局描述.

定义 1. 基本过程. 一个基本过程是一个四元组 $b = (P, T; F, M_0)$, 其中:

(1) $(P, T; F)$ 是一个没有孤立元素的 Petri 网, $P \cup T \neq \emptyset$;

(2) P 是基本过程中条件的有限集, $\forall p \in P$ 称为一个条件;

(3) T 是基本过程中活动的有限集, $\forall t \in T$ 称为一个活动,活动 t 的执行称为 t 的发生或者点火;

(4) F 是弧的有限集, $F \subseteq (P \times T) \cup (T \times P)$, 对于 $\forall p \in P, \forall t \in T$, 若 $(p, t) \in F$ 或者 $(t, p) \in F$, 则在 p 和 t 之间有一条有向边,称为弧;

(5) $M \subseteq 2^P$ 是格局集, 其中的每一个格局描述了基本过程的一个执行状态, 2^P 是 P 的幂集, $M_0 \in M (M_0 \subseteq P)$ 是一个格局, 并且称为 p 的初始标识 (marking), $m \in M_0$ 称为托肯, 通过库所中托肯的动态变化表示过程执行的不同状态;

(6) $\forall t \in T, \exists M_a \in M, t$ 在格局 M_a 有发生权.

基本过程是由细粒度的 Petri 网合成的, 我们将这些 Petri 网称为块, 块包括基本块和过程包. 基本块是一个单入口、单出口且内部结构可见的 Petri 网, 基本块中活动间的关系定义为顺序关系、选择关系、迭代关系和并发关系, 对应于基本块, 我们在文

献[39]中定义了顺序块、选择块、迭代块和并发块; 过程包是一个单入口、单出口且内部结构不可见的 Petri 网[39]. 基本块和过程包用于基本过程的白盒建模与黑盒建模[39].

基于面向方面方法的思想, 我们将业务过程的核心活动定义为基本活动, 基本活动反映了具体应用领域中的业务功能, 而将安全、日志、确认、签署、审核和归档等保证基本过程活动安全和可追踪的横切活动封装为横切方面, 未织入基本过程的横切活动不能反映业务功能, 只有将横切活动编织入基本过程, 对基本活动进行扩展和约束才能体现其在业务功能中的作用. 将横切活动与基本过程分离, 能够有效减少业务过程不同功能活动之间的“纠缠”和横切活动在基本过程中“分散”交织的问题.

定义 2. 横切方面. 一个横切方面是一个三元组 $s = (i, d, J)$, 其中: i 是横切方面的标识符, d 是方面通知, J 是方面切点集合.

横切方面 s 的标识符 i 如果为 0, 表示这个横切方面是将多个横切方面编织在一起得到的, 如果不为 0, 则标识单个横切方面. 方面通知 d 定义方面提供服务的内容, 方面切点 $j \in J$ 定义基本过程可以被方面修改或改进的位置, 下面对横切方面中的方面通知和切点进行定义.

定义 3. 方面通知. 一个方面通知是一个五元组 $d = (P_s, T_s; F_s, T_e, T_x)$:

(1) $(P_s, T_s; F_s)$ 是一个没有孤立元素的网, $T_s \cup P_s \neq \emptyset$;

(2) P_s 是方面通知中条件的有限集, $\forall p_s \in P_s$ 称为一个条件;

(3) T_s 是方面通知中活动的有限集, $\forall t_s \in T_s$ 称

为一个活动,活动 t_s 的执行称为 t_s 的发生或者点火;

(4) F_s 是弧的有限集, $F_s \subseteq (P_s \times T_s) \cup (T_s \times P_s)$, 对于 $\forall p_s \in P_s, \forall t_s \in T_s$, 若 $(p_s, t_s) \in F_s$ 或者 $(t_s, p_s) \in F_s$, 则在 p_s 和 t_s 之间有一条有向边, 称为弧;

(5) $T_e, T_x \subseteq T_s$ 分别是方面通知的入口活动集和出口活动集. 若 $(P_s, T_s; F_s)$ 包含一个环 $(p_{s1}, p_{s1}), (t_{s1}, p_{s2}), \dots, (p_{sn}, t_{sn}), (t_{sn}, p_{s1})$ 导致 T_e 和 T_x 无法确定, 或者 T_e 无法确定, 或者 T_x 无法确定, 则针对无法确定的 T_e 或者 T_x , 指定 $t_e \in T_s$ 为环的入口活动, $t_x \in T_s$ 为环的出口活动, 并在 t_e 前增加一个活动 t' , 满足 $t' \cdot = t_e$ (t' 是 t' 的后集, 即 t' 的输出库所, t_e 是 t_e 的前集, 即 t_e 的输入库所), 同时, 在 t_x 后增加一个活动 t'' , 满足 $t_x \cdot = t''$, 则 $T_e = \{t'\}$ 和 $T_x = \{t''\}$.

方面通知是一个没有格局的基本 Petri 网, 即方面通知中的活动没有发生权, 只有将方面编织入基本过程后, 其中的活动在编织后的过程模型格局中才有发生权. 同样, 方面通知也是通过采用与基本过程一样的黑盒建模与白盒建模方法^[30]得到.

定义 4. 方面切点. 方面切点是由基本过程中的元素 r 和方面织入类型 q 构成的集合 $J = \{(r, q) \mid r \in b, P \vee r \in b, T \vee r \in b, F, q \text{ 是织入类型}\}$, 按照基本过程元素类型, r 定义为基本过程的条件、活动或者弧; 方面织入类型 q 根据方面与基本过程的关系定义为活动前、活动后和活动间顺序关系织入, 选择关系织入, 迭代关系织入和并发关系织入 6 种类型, 分别用数字 1~6 表示.

方面切点是识别方面连接点的方法, 连接点是在基本过程执行时动态产生的基本过程元素, 也就是过程执行出现网中的条件、活动或者弧, 连接点用静态的切点定义方面提供服务的位置模式, 不需一一指明每个连接点就可以捕获织入方面的连接点.

由于横切方面织入基本过程时, 方面之间、方面与基本过程之间存在着交互关系, 其中的一些交互关系有可能导致方面织入基本过程后使过程无法执行、出现异常或错误, 因此, 面向方面业务过程建模框架的第 2 步是基于 Petri 网的相关性质和面向方面方法的特点, 对面向方面业务过程建模的正确性进行分析并提出保证过程模型正确性的方法.

4 面向方面业务过程建模正确性

面向方面方法可以基于更高的模块化程度为大型业务过程建模奠定基础, 可以非常有效地避免传统建模方法中不同功能活动之间的纠缠及分散问

题. 然而, 方面的编织有可能在违背原有意图的情况下引入恶意行为, 产生结构或性质错误, 从而导致编织后的模型不符合预期需求. 因此, 接下来我们对面向方面业务过程建模的正确性进行分析和定义, 并提出保证正确性的方法.

业务过程模型验证正确性主要以合理性 (soundness) 验证为基础, 合理性来源于 workflow 领域的工作流网 WF-nets 定义, 之后应用到了普遍业务过程建模领域. 合理性规定工作流网一定存在到达终止状态的路径、不存在死变迁且到达终止状态时仅有终止库所有托肯^[6]. 本文根据面向方面建模方法的特点, 在保证业务过程合理性的基础上扩展方面编织正确性.

下面, 首先探讨方面间编织正确性问题, 因为在基本过程一个切点上如果有多个相同织入类型的方面需织入, 则需要先将这些方面编织为一个方面后再织入基本过程, 由于这些方面中的活动可能会存在依赖关系, 因此需要按照方面间依赖关系对方面间编织正确性进行分析.

4.1 方面间编织正确性

Kniesel 和 Bardey^[40] 认为大部分的方面间编织错误问题源于不完整或者不正确的编织, 所谓不完整的编织是指方面没有完全编织到应该编织的所有连接点上, 造成方面影响的丢失; 而不正确的编织是指方面没有正确的编织在应该编织的连接点上, 造成错误的方面影响. 基于此思想, 我们定义方面间编织的完整性和正确性:

(1) 完整性. 方面应在所有切点定义的连接点处织入基本过程且保证不丢失所有织入方面的影响;

(2) 正确性. 方面仅在切点定义的连接点处织入基本过程且不造成错误的方面影响.

针对方面间编织完整性, 只要切点定义的连接点存在且是正确的, 方面织入可以保证所有方面在其切点定义的连接点处织入基本过程, 对于连接点不存在或者连接点定义错误的情况, 可以在方面织入过程中被检测到并解决. 但是, 需要保证所有织入的方面具有发生权且共享切点上所有存在数据依赖关系的方面按照依赖关系顺序编织, 否则织入的流程方面可能会丢失影响, 因此, 编织完整性准则定义如下.

定义 5. 编织完整性准则. 设 $S = \{s_1, \dots, s_n\}$ 是一个共享切点的横切方面集合, 对于其中任意两个横切方面 s_α 和 s_β 中的通知 d_α 和 d_β ($1 \leq \alpha \leq n, 1 \leq \beta \leq n$ 且 $\alpha \neq \beta$), 其中有数据依赖关系的活动 t_{s_α} 和 t_{s_β} 按照数据依赖关系顺序编织, 满足 $M[t_{s_\alpha}]$ (在格局

M 下 t_{sa} 可发生)但 $\neg M[t_{s\beta}]$ (在格局 M 下 $t_{s\beta}$ 不可发生) $M[t_{sa}]M' \rightarrow M'[t_{s\beta}]$ (在格局 M 下 t_{sa} 执行后进入格局 M' , 在格局 M' 下, $t_{s\beta}$ 可发生)。

活动间的数据依赖关系定义了活动的执行顺序关系, 如果一个活动 $t_{s\beta}$ 数据依赖于另一个活动 t_{sa} , 那么, 活动 $t_{s\beta}$ 要发生必须得到 t_{sa} 的输出数据, 否则 $t_{s\beta}$ 不可发生^[39]。

另外, 活动间还存在控制依赖关系, 控制依赖关系定义了活动间的控制执行关系。如果一个活动 $t_{s\beta}$ 受控依赖于另一个活动 t_{sa} , 那么, 活动 $t_{s\beta}$ 是否可发生必须根据 t_{sa} 的输出结果来判定, t_{sa} 的输出结果判定分为两种情况, 一种情况是 $t_{s\beta}$ 可发生, 且 $t_{s\beta}$ 执行完后执行后续活动; 另一种情况是 $t_{s\beta}$ 不可发生, 直接执行后续活动^[39]。因此, 针对方面间编织正确性, 需要所有存在控制依赖关系的方面可发生且按照控制关系编织才能保证方面织入不造成错误的影响, 因此, 编织正确性准则定义如下。

定义 6. 编织正确性准则。设 $S = \{s_1, \dots, s_n\}$ 是一个共享切点的横切方面集合, 对于其中任意 3 个流程方面 s_α, s_β 和 s_γ 中通知 d_α, d_β 和 d_γ ($1 \leq \alpha \leq n, 1 \leq \beta \leq n, 1 \leq \gamma \leq n$ 且 $\alpha \neq \beta \neq \gamma$), 其中有控制依赖关系的活动 $t_{sa}, t_{s\beta}$ 和 $t_{s\gamma}$ 按照控制依赖关系选择编织, 满足 $M[t_{sa}]M', M'[t_{s\beta}]$ 或者 $M'[t_{s\gamma}]$ 但 $\neg M'[t_{s\beta}, t_{s\gamma}]$ 。

对于共享切点上具有相同织入类型, 但没有依赖关系的横切方面, 采用并发方式编织, 在将所有这些方面编织为一个方面后, 需要保证在将其织入基本过程时, 方面织入基本过程不引入错误且编织后得到过程模型无异常。因此, 下面我们对方面织入基本过程正确性进行分析。

4.2 方面织入基本过程正确性条件

由于横切方面仅在切点定义位置织入基本过程, 方面对基本过程产生影响的区域仅限于切点周围的活动, 对于织入位置影响区域之外的活动, 其结构、性质和行为完全不受影响。因此, 我们假设方面定义的通知模型和基本过程模型在编织之前已通过正确性验证, 本文对方面织入基本过程的正确性分析不覆盖织入方面后的整个业务过程模型, 而仅限于切点织入的区域, 由于这个区域外的活动与方面没有直接或间接关系, 没有必要将其引入织入正确性验证, 以免带来不必要的状态空间爆炸问题。van der Aalst 和 Sun 在研究工作流动态改变保持合理性的分析中采用了同样的方法以避免状态空间爆炸问题, 有利于大规模复杂系统的分析^[11,41]。下面, 我们将方面织入基本过程所影响的活动以及这些活动

相关的条件和弧定义为编织区域。

定义 7. 编织区域。设 $s = (i, d, J)$ 是一个横切方面, $b = (P, T; F, M_0)$ 是一个基本过程, s 按照编织类型 q 织入 b 的切点 j , $r (r \in b.P \vee r \in b.T \vee r \in b.F)$, 得到编织区域 $g = (P_g, T_g; F_g)$ 满足以下 3 个条件:

- (1) $g.T_g = \{t_g \mid t_g \in j.r \cup j.r' \cup j.r \cup \text{dom}(j.r) \cup \text{cod}(j.r) \wedge t_g \in b.T\} \cup d.T_e \cup d.T_x$;
- (2) $g.P_g = \{p_g \mid p_g \in g.T_g \cup g.T_g \wedge p \in b.P \cup d.P_s\}$;
- (3) $g.F_g = \{f_g \mid f_g \in (g.P_g \times g.T_g) \cup (g.T_g \times g.P_g)\}$ 。

编织区域由横切方面的织入位置决定, 是以切点周围的活动和织入方面的通知入口及出口活动为中心, 将这些活动以及与这些活动相关的条件和弧定义为编织区域。因此, 编织区域也是一个 Petri 网, 这个网中的所有活动都是受到横切方面直接或者间接影响的活动。

例如: 一个基本过程 $b = (\{p_1, p_2, p_3\}, \{t_1, t_2\}; \{(p_1, t_1), (t_1, p_2), (p_2, t_2), (t_2, p_3)\}, \{p_1\})$, 一个横切方面 s 的通知 $d = (\{p_{s1}, p_{s2}\}, \{t_{s1}\}; \{(p_{s1}, t_{s1}), (t_{s1}, p_{s2})\})$ 在基本过程 b 的切点 $j = (b.t_2, 4)$ 选择织入, 如图 2 所示。

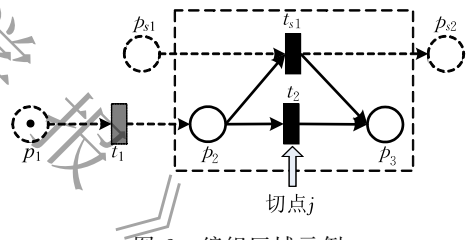


图 2 编织区域示例

按照定义 7, 编织区域由虚线框中的 Petri 网构成, 其形式化描述为 $g = (\{p_2, p_3\}, \{t_2, t_{s1}\}; \{(p_2, t_2), (t_2, p_3), (p_2, t_{s1}), (t_{s1}, p_3)\})$ 。

在定义了编织区域后, 下面, 基于 Petri 网的正确性、工作流的合理性和过程规约定义面向方面业务过程建模的正确性, 包括方面织入基本过程的结构正确性、性质合理性和行为一致性。

4.2.1 方面织入基本过程正确性定义

(A) 结构正确性

面向方面业务过程模型的结构性质取决于其拓扑结构, 与初始格局无关, 即: 任何初始格局下都应保持的结构性质。

定义 8. 结构正确性。设 $s = (i, d, J)$ 是一个横切方面, $b = (P, T; F, M_0)$ 是一个基本过程, s 织入 b 后, 在编织区域 g 保持结构正确当且仅当满足以下

3 个条件:

- (1) 如果 $\forall t_g \in g.T_g, \dot{t}_g \cap t_g = \emptyset$;
- (2) 如果 $\forall \alpha_g \in g.P_g \cup g.T_g, \dot{\alpha}_g \cup \alpha_g \neq \emptyset$;
- (3) 如果 $P_{g1} \subseteq g.P_g$, 不存在 $\dot{P}_{g1} \subseteq P_{g1}$ 或者 $P_{g1} \subseteq \dot{P}_{g1}$.

定义 8 中, 第 1 个条件确保编织区域 g 中的活动未被引入伴随条件^[42], 因为伴随条件让活动永远没有发生权. 第 2 个条件确保编织区域 g 未被引入孤立结点^[43]. 第 3 个条件确保编织区域 g 中的条件子集无死锁($\dot{P}_{g1} \subseteq P_{g1}$)和陷阱($P_{g1} \subseteq \dot{P}_{g1}$), 在一个业务过程执行时, 一个不含有托肯的死锁条件永远不会得到托肯, 一个含有托肯的陷阱条件永远不会失去托肯^[43].

(B) 性质合理性

上述结构正确性与业务过程模型的初始格局无关, 但格局反映了过程的动态状态, 因此, 除了结构正确性外, 还需借助业务过程模型来分析实际过程在初始格局下的动态性质. 另外, 良构的业务过程可以用合理的工作流来定义^[6, 44], 而合理性规定工作流网一定存在到达终止状态的路径、不存在死变迁且到达终止状态时仅有终止库所有托肯^[6]. 基于工作流网的合理性定义, 我们定义面向方面业务过程模型的性质合理性.

定义 9. 性质合理性. 设 $s = (i, d, J)$ 是一个横切方面, $b = (P, T; F, M_0)$ 是一个基本过程, s 织入 b 后, 在编织区域 g 保持动态性质当且仅当满足以下 4 个条件:

- (1) 如果 $\forall t_s \in d.T_s$ 且 $\exists M \in R(b, M_0), M[t_s]$;
- (2) 如果 $\forall t \in g.T_g \cap b.T, \forall \sigma \in d.T_s^*$ 且 $\forall M \in R(b, M_0)$, 有 $(M[t] \wedge M[\sigma]M') \rightarrow M'[t]$, 其中 σ 是过程通知中的活动序列;
- (3) 如果 $\forall p_g \in g.P_g$ 且 $L(p_g) = \min\{L \mid \forall M \in R(b, M_0): M(p_g) \leq L\}$, 有 $L(p_g) = 1$;
- (4) 如果 $\forall p_g \in g.P_g, \forall t_g \in g.T_g$ 且 $\forall M \in R(b, M_0)$, 不存在 $p_g \in t_g \cap M$ 且 $t_g \subseteq M$.

定义 9 中, 第 1 个条件确保横切方面织入基本过程后, 方面通知是可发生的^[45]. 第 2 个条件确保织入横切方面后基本过程是持续的^[43]. 第 3 个条件确保编织区域 g 中的所有条件满足安全性^[43]. 第 4 个条件确保编织区域 g 中不存在有冲撞的条件^[43].

(C) 行为一致性

行为一致性是在横切方面织入基本过程时, 特指编织后基本过程的活动关系与过程需求保持一致. 基于 Li^[39] 提出的软件演化过程基本块中活动间

关系, 本文面向方面业务过程模型的行为由顺序、选择、并发和迭代这 4 种关系构成, 当横切方面织入基本过程后, 不破坏基本过程中活动的原有行为关系. 下面, 对基本过程中活动的四类行为关系中的顺序行为关系和并发行为关系进行定义, 选择行为关系和迭代行为关系的定义类似, 故在此省略.

定义 10. 顺序行为关系. 设 $b = (P, T; F, M_0)$ 为一个基本过程, $t_a, t_\beta \in T, M \in R(M_0)$, 如果

- (1) $M[t_a] \text{ 但 } \neg M[t_\beta]$;
- (2) $\forall \sigma \in T^*$ 使得 $M[t_a]M'[\sigma]M'' \rightarrow M''[t_\beta]$;

则称 t_a 和 t_β 之间是顺序行为关系. 当活动序列 σ 长度为 0 时, $M[t_a]M' \rightarrow M'[t_\beta]$, 称 t_a 和 t_β 是直接顺序行为关系; 当活动序列 σ 长度不为 0 时, 称 t_a 和 t_β 是间接顺序行为关系.

定义 11. 并发行为关系. 设 $b = (P, T; F, M_0)$ 为一个基本过程, $t_a, t_\beta \in T, M \in R(M_0)$, 如果

- (1) 对于 $\forall \sigma \in T^*$, 有 $M[t_a]$ 且 $M[\sigma]$ 或者 $M[t_\beta]$ 且 $M[\sigma]$;
- (2) $M[\sigma]M_1 \rightarrow M_1[t_a] \wedge M_1[t_\beta]$ 且 $M_1[t_a]M' \rightarrow M'[t_\beta]$ 或者 $M_1[t_\beta]M'' \rightarrow M''[t_a]$, 或者 $M[t_a]M_2[\sigma]M_3 \rightarrow M_3[t_\beta]$, 或者 $M[t_\beta]M_4[\sigma]M_5 \rightarrow M_5[t_a]$;

则称 t_a 和 t_β 之间是并发行为关系. 当活动序列 σ 长度为 0 时, 有 $M[t_a]$ 且 $M[t_\beta]$, $M[t_a]M_2 \rightarrow M_2[t_\beta]$ 或 $M[t_\beta]M_4 \rightarrow M_4[t_a]$, 称 t_a 和 t_β 是直接并发行为关系; 当活动序列 σ 长度不为 0 时, 称 t_a 和 t_β 是间接并发行为关系.

基于活动间行为关系的定义, 当横切方面织入基本过程时, 行为一致性定义如下.

定义 12. 行为一致性. 一个横切方面织入基本过程保持行为一致当且仅当织入横切方面后编织区域中原基本过程的活动行为与织入前行为一致.

4.2.2 方面织入正确性准则

由于基本过程以活动为中心, 当对基本过程进行横切方面扩展或约束时, 可能的扩展和约束行为围绕活动包括 6 种情况: 在活动执行之前增加横切方面约束、在活动执行之后扩展横切方面、在两个活动之间增加横切方面约束、对某活动是否能执行增加选择扩展、对基本过程活动增加迭代执行的横切方面或者并发执行的横切方面. 然而, 基于面向方面方法的 before、after 和 around 这 3 种编织类型^[46-47]以及方面定义的 3 种切点类型(条件、活动、弧), 可以定义 10 类方面织入基本过程的操作, 分别

是条件前织入、条件后织入、条件周围织入、迭代织入、活动前织入、活动后织入、活动周围织入、并发织入、条件活动弧织入和活动条件弧织入。但是,根据方面织入正确性定义 8、9 及 11,这 10 类织入操作中条件前织入、条件后织入、活动前织入和活动后织入操作分别不满足无冲突、持续性和行为一致性,并且,这 4 类操作可以用其他满足正确性的织入操作替代。因此,为保证横切方面织入基本过程的正确性,定义如表 1 所示的 6 个方面织入正确性准则,这些准

表 1 横切方面织入准则

织入准则	图示
活动前织入准则:活动前织入的切点定义为基本过程中连接条件和活动的弧,实施条件活动弧织入,横切方面 s 织入结果是在活动 t 前顺序加入方面通知。以右图所示为例,实线框中是基本过程 b ,虚线框中是横切方面 s 的通知 d , s 的切点 $J = \{(b, (p, t), 1)\}$,方面织入类型为活动前织入。	
活动后织入准则:活动后织入与活动前织入类似,只是切点定义为连接活动和条件的弧,实施活动条件弧织入,横切方面 s 织入结果是在活动 t 之后顺序加入方面通知。如右图所示,切点 $J = \{(b, (t, p), 2)\}$,方面织入类型为活动后织入。	
活动间织入准则:活动间织入是在活动之间织入横切方面的通知,实施条件周围织入,此织入操作的结果是在切点条件前后集活动之间顺序插入横切方面的通知。如右图所示,切点 $J = \{(b, p, 3)\}$,方面织入类型为活动间织入。	
选择织入准则:选择织入以选择关系织入横切方面的通知,织入后,横切方面的通知与切点活动之间形成选择关系。需要注意的是,织入后,切点活动可能被方面通知中的活动取代,如果织入后需要满足原过程需求,方面通知中的活动必须在满足横切需求的同时包含切点活动的功能。如右图所示,切点 $J = \{(b, t, 4)\}$,方面织入类型为选择织入。	
迭代织入准则:迭代织入是将横切方面的通知迭代织入基本过程,织入后,横切方面的通知与切点活动之间形成迭代关系。需要注意的是,当切点活动 t 在织入操作前已经有迭代行为关系的活动 t' 时,迭代织入使方面通知与活动 t' 在方面织入后形成选择关系,与选择织入类似,活动 t' 可能被方面通知中的活动取代,如果需要满足原过程需求,方面通知中活动必须在满足横切需求的同时包含活动 t' 的功能。如右图所示,切点 $J = \{(b, t, 5)\}$,方面织入类型为迭代织入。	
并发织入准则:并发织入是将横切方面的通知并发织入基本过程,织入后,横切方面的通知与切点之间形成并发关系。如右图所示,切点 $J = \{(b, t, 6)\}$,方面织入类型为并发织入。	

则所对应的织入操作满足对基本过程所需要的 6 类扩展和约束行为,其中:假设横切方面 $s = (i, d, J)$,基本过程 $b = (P, T; F, M_0)$ 。

4.3 方面织入基本过程正确性检测

上一节针对横切方面织入基本过程需要保证的正确性进行定义并设计了保证正确性的方面织入准则,这属于预防性手段。除此之外,建模过程仍然存在因疏忽而导致引入错误和异常的可能性,因此,本节提出横切方面织入基本过程的错误检测及消解方法。

面向方面方法中的方面除了可以用于扩展基本过程外,还可以用于检测冲突,并且拦截匹配到的冲突。在上一节中定义的横切方面用于扩展基本过程,本节定义错误检测方面,在横切方面织入基本过程时进行错误检测。通过织入前的正确性控制,织入过程中的错误检测及消解,实现横切方面织入完全无错误保证。

面向方面方法的核心概念是连接点、切点、通知及编织机制,针对横切方面织入的错误检测,连接点定义为横切方面织入基本过程产生错误的位置,切点定义错误模式,当横切方面织入基本过程时,错误检测方面的切点拦截存在错误的连接点,并启动方面通知的执行,方面通知根据切点定义的错误模式反馈错误类型、错误匹配连接点以及引入错误的横切方面标识。单个错误检测方面实现单一错误检测,多个错误检测方面编织在一起则可以实现复杂模式的错误检测。

基于上述面向方面错误检测方法提出的连接点、切点、通知及编织机制,错误检测方面定义如下。

定义 13. 错误检测方面。一个错误检测方面是一个三元组 $s_w = (i_w, d_w, J_w)$ 。

- (1) i_w 是错误检测方面标识,用于标识不同的错误检测方面;
- (2) d_w 是通知,阻止横切方面织入并报告错误信息;
- (3) J_w 是切点,定义错误模式。

根据建模者不同的建模需求,可以灵活地定义不同的错误检测方面,实现不同类型的错误检测。本文针对横切方面织入基本过程可能引入的结构、性质和行为错误,定义如下三类错误检测方面。

(1) 结构错误检测方面

针对横切方面织入结构正确性定义 8,通过结构错误分析,我们定义如图 3(a)和图 3(b)所示的两类结构错误检测方面。

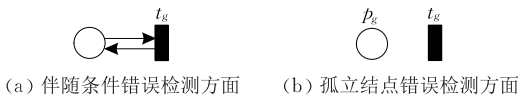


图 3 结构错误检测方面

其中:(a) 伴随条件错误检测方面遍历编织区域 g 中的活动,若一个活动 $t_g \in g$, T_g 匹配切点定义 $\cdot t_g \cap t_g \cdot \neq \emptyset$,那么这个活动 t_g 存在伴随条件错误. 此时,方面通知检测引入错误的弧,如果可以通过删除多余弧消解错误就删除引入伴随条件的弧;如果无法确定引入伴随条件的多余弧,则报告冲突位置为活动 t_g ; (b) 孤立结点错误检测方面遍历编织区域 g 中的活动和条件,若存在一个活动或条件 $\alpha_g \in g$, $P_g \cup g$, T_g 匹配切点定义 $\cdot \alpha_g \cup \alpha_g \cdot = \emptyset$,那么这个活动或条件 α_g 是孤立结点,此时,如果此孤立结点来自于横切方面,则报告产生此孤立结点的横切方面标识 i ; 如果此孤立结点来自于基本过程,则报告冲突位置为活动或条件 t_g .

另外,根据结构死锁的定义,活动间的迭代行为关系导致其前后集条件属于潜在结构死锁,由上述结构正确性分析已知,潜在结构死锁条件集中的条件一旦获得托肯,则无结构死锁,只有存在迭代关系的活动无发生权时才能判断出结构死锁存在. 因此,本文不针对死锁错误定义检测方面,对于编织区域 g 中的迭代关系活动,只需要对活动进行可发生性检测(在下面性质错误检测方面中进行介绍). 同样,根据结构陷阱的定义,织入横切方面中的通知如果永远持有托肯则导致结构陷阱. 由上述结构正确性分析已知,基本过程如果在织入横切方面后仍然是持续的,则无结构陷阱,只有横切方面织入后通知中存在无后集活动的条件导致基本过程不持续时,才能判断出结构陷阱错误存在,因此,同样不针对陷阱错误定义检测方面,而是对编织区域 g 进行持续性检测(在下面性质错误检测方面中进行介绍).

(2) 性质错误检测方面

针对横切过程方面有可能引入的四类性质错误以及相关错误控制分析,我们定义如图 4(a)和图 4(b)所示的两类性质错误检测方面.

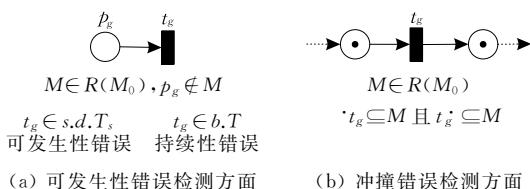


图 4 性质错误检测方面

对于图 4(a)可发生性错误检测方面,由于可发生性和持续性都属于活动有无发生权的情况,因此,统一定义可发生性错误检测方面. 当无发生权的活动在横切方面的通知中时,属于可发生性错误,而当无发生权的活动在基本过程中时,属于持续性错误. 可发生性错误检测方面的错误检测是遍历可达标识集,如果存在活动 t_g 的前集条件不在可达标识集中,说明活动 t_g 无发生权,此时,如果活动 t_g 属于横切方面中的通知,则为可发生性错误;如果活动 t_g 属于基本过程,则为持续性错误.

另外,对于图 4(b)冲突错误检测方面,如果存在 $M \in R(M_0)$, 有 $\cdot t_g \subseteq M$ 且 $t_g \cdot \subseteq M$, 那么活动 t_g 的后集条件存在冲突. 对于安全性错误,由于 $M(p_g) > 1$ 的情况与冲突一样,因此,仅检测冲突错误.

(3) 行为错误检测方面

上述 4.2.1 节定义了业务过程活动间的顺序、并发、选择及迭代行为关系,在研究行为错误时,两个活动间的间接行为关系被破坏可以体现为另外两个活动之间的直接行为关系被破坏,例如:假设活动 t_{g1} 和活动 t_{g3} 之间的间接顺序行为关系被破坏,其实也是活动 t_{g1} 和活动 t_{g2} 之间或者活动 t_{g2} 和活动 t_{g3} 之间的直接顺序行为关系遭到破坏,如图 5 所示.

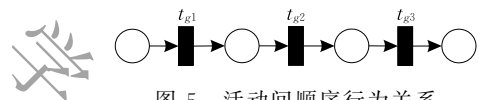


图 5 活动间顺序行为关系

因此,只需要针对直接行为关系来研究横切方面有可能引入的行为错误. 通过上述 4.2.2 节方面织入准则可知,本文定义的方面织入准则只会让活动间的直接行为关系变为间接行为关系,由行为一致性定义可知,这样的改变并不破坏原有行为关系. 另外,对于产生选择关系的织入操作,如果为了保持原有行为关系,织入横切方面的过程通知必须保持切点活动的功能.

5 面向方面业务过程建模

按照面向方面业务过程建模框架,下面定义面向方面业务过程建模的编织规则,编织规则由方面间编织规则和方面织入基本过程规则构成,方面间编织规则是将共享切点上的多个相同织入类型横切方面编织为一个横切方面,方面织入基本过程规则将横切方面织入基本过程的切点,方面编织工作由编织器完成,按照上述编织分类分别定义编织结构及编织算法.

定义 14. 方面编织规则. 方面编织规则是一个二元组 $RL = (K(W), O(W))$, 其中: W 是编织器, $K(W)$ 是编织结构, $O(W)$ 是编织算法.

(1) 编织结构 $K(W)$ 定义横切方面编织的结构关系, 编织器 W 按照不同的横切方面实施不同的结构编织;

(2) 编织算法 $O(W)$ 按照编织结构 $K(W)$ 定义方面间编织算法和方面织入基本模型算法.

基于方面编织规则, 下面按照编织操作分为方面间编织和方面织入基本过程两类, 分别提出方面间编织方法和方面织入基本过程方法, 并分别对方面织入基本模型的结构正确性、性质合理性以及行为一致性进行证明, 最后对方面间编织完整性和正确性进行证明.

5.1 方面间编织

当多个横切方面需要在同一个切点以相同织入类型织入基本过程时, 我们先把这些方面编织为一个方面, 再将编织后的方面织入基本过程. 由于同一切点上需要实施同类型织入的横切方面之间可能存在依赖关系, 基于 4.1 节中编织完整性及正确性定义, 下面对同类型方式织入同一切点的横切方面中的过程通知使用 Li 的依赖关系分析方法^[39] 进行分析, 对于存在依赖关系的活动构建活动依赖图 ADG (Activity Dependence Graph), 然后使用 Li 的转换规则^[39] 将活动依赖图 ADG 转换为过程片段, 此过程片段是多个同切点同类型方面编织后得到的横切方面中通知的 Petri 网. 对于没有依赖关系的活动, 实施方面间并发编织操作.

算法 1. 方面间编织算法 Aspect_Merging.

输入: $s_1 = (i_1, d_1, J_1), \dots, s_n = (i_n, d_n, J_n)$

输出: $s = (i, d, J), s_1 = (i_1, d_1, J_1), \dots, s_n = (i_n, d_n, J_n)$

BEGIN

分析 $s_1, \dots, s_n$ 中所有活动的数据依赖关系 E_d ;

分析 $s_1, \dots, s_n$ 中所有活动的控制依赖关系 E_c ;

IF α 个方面存在依赖关系 THEN /* $\alpha \leq n$ */

BEGIN

分析 $s_1, \dots, s_n$ 通知中所有活动的输入/输出数据集 $\text{input}(t_s), \text{output}(t_s), t_s \in s_1.d_1.T_1 \cup \dots \cup s_n.d_n.T_n$;

调用 Constructing_DG($s_1.d_1.T_1 \cup \dots \cup s_n.d_n.T_n$,

$\{\text{input}(t_s)\}, \{\text{output}(t_s)\}, E_d, E_c, \text{ADG}$);

调用 Localising_Dependences(ADG);

调用 Simplifying_ADG(ADG);

调用 Preprocessing_SADG(ADG);

调用 Constructing_Process_Segment(ADG, h)

END;

$d.P := h.P; d.T := h.T; d.F := h.F;$

FOR $\forall t \in h.T$ DO

BEGIN

/* 修改编织后方面通知中入口活动集和出口活动集 */

IF $t = \emptyset$ THEN

$d.T_e = d.T_e \cup \{t\};$

IF $t' = \emptyset$ THEN

$d.T_x = d.T_x \cup \{t\};$

END;

FOR $\gamma = \alpha + 1$ TO n DO /* 无依赖关系的方面 */

BEGIN /* 方面间并发编织 */

$d.P_s := d.P_s \cup d_\gamma.P_s;$

$d.T_s := d.T_s \cup d_\gamma.T_s \cup \{v_1, v_2\};$

FOR $t_e \in d.T_e \cup d_\gamma.T_e$ AND $t_x \in s.T_x \cup s_\gamma.T_x$ DO

$d.F_s := d.F_s \cup d_\gamma.F_s \cup \{(v_1, t_e), (t_x, v_2)\};$

$d.T_e := \{v_1\};$ /* v_1, v_2 是虚活动, 仅传递托肯 */

$d.T_x := \{v_2\}$

END;

$i := 0;$

$J := s_1.J_1 \cap s_2.J_2 \cap \dots \cap s_n.J_n;$

FOR $\gamma = 1$ TO n DO /* 修改方面切点定义 */

$s_\gamma.J_\gamma := s_\gamma.J_\gamma - J;$

END

其中, Constructing_DG 算法^[39] 将方面通知中的活动按照活动间依赖关系构建活动依赖图 ADG, 如果 ADG 中活动需要进一步分解为子活动, 则调用 Localising_Dependences^[39] 细化 ADG, 并调用 Simplifying_ADG^[39] 消除 ADG 生成过程中可能产生的冗余元素, 在调用 Preprocessing_SADG^[39] 将 ADG 预处理为单入口和单出口 ADG 后, 调用 Constructing_Process_Segment^[39] 将 ADG 转换为 Petri 网 h .

5.2 方面织入基本过程

按照 4.2.2 节方面织入准则, 横切方面的织入分为顺序织入、选择织入、迭代织入和并发织入. 由于选择织入、迭代织入和并发织入与顺序织入类似, 下面仅介绍顺序织入.

顺序织入是将横切方面织入基本过程, 使方面通知中的活动与切点活动形成顺序行为关系. 如果有多个方面需要织入同一切点且织入类型相同, 则先实施 5.1 节介绍的方面间编织操作, 然后再织入基本过程. 由 4.2.2 节的方面织入准则可知, 顺序织入定义为活动前织入、活动后织入和活动间织入.

设 $s = (i, d, J)$ 是一个横切方面, $d = (P_s, T_s; F_s, T_e, T_x)$ 是 s 的通知, $b = (P, T; F, M_0)$ 是一个基本过程, 顺序织入是将方面 s 在活动前、活动后或者活动间顺序织入 b 中. 按照 4.2.2 节的方面织入准则, 顺序织入具体由条件活动弧织入、活动条件弧织

人和条件周围织入完成. 假设弧切点为 $j_f (j_f \in J_f, J_f \subseteq J, J_f$ 是 J 中的弧切点集合), s 织入 b 后得到面向方面业务过程 $b_s = (P, T; F, M_0)$, 算法 2 给出了活动前织入算法, 活动后织入和活动间织入类似, 故在此省略.

算法 2. 顺序织入算法 Sequence_Weaving.

输入: b, s

输出: $b_s = (P, T; F, M_0)$

BEGIN

$b_s.P := \emptyset; b_s.T := \emptyset; b_s.F := \emptyset; b_s.M_0 := \emptyset;$

IF $dom(j_f.r) \in b.P$ THEN /* 活动前织入 */

BEGIN

IF $|T_e| > 1$ THEN

$b_s.P := b.P \cup s.d.P_s;$

$b_s.T := b.T \cup s.d.T_s \cup \{v\};$

/* 增加虚活动 v 传递托肯 */

$b_s.F = b.F \cup s.d.F_s - J_{f,r} \cup \{(dom(j_f.r), v)\} \cup$

$\{(v, p_{sa}) | p_{sa} \in s.d.T_e\} \cup$

$\{(p_\beta, cod(j_f.r)) | p_\beta \in s.d.T_x\}$

/* 增加方面与基本过程之间的连接弧 */

ELSE

$b_s.P := b.P \cup s.d.P_s - s.d.T_e$

$b_s.T := b.T \cup s.d.T_s;$

$b_s.F = b.F \cup s.d.F_s - J_{f,r} - inflow(s.d.T_e) \cup$

$\{(dom(j_f.r), t_{sa}) | t_{sa} \in s.d.T_e\} \cup$

$\{(p_\beta, cod(j_f.r)) | t_\beta \in s.d.T_x\}$

/* 增加方面与基本过程之间的连接弧, 其中:

$inflow()^{[29]}$ 定义了活动前所有的弧 */

$b_s.M_0 := b.M_0$

/* 织入方面后的业务过程初始标识仍为基本过程的初始标识 */

END

END

5.3 面向方面业务过程建模正确性证明

上面我们对面方面业务过程建模相关概念进行了定义并提出方面间编织算法和方面织入基本过程算法, 下面证明按照本文定义的正确性准则, 方面织入基本过程具有结构正确性、性质合理性及行为一致性, 方面间编织可保证编织完整性及正确性.

5.3.1 结构正确性

结构正确性要求横切方面织入基本过程时无结构错误, 我们假设横切方面的通知和基本过程在编织前都满足 4.2.1 节定义 8 所定义的结构正确性, 由于织入操作完成后, 可能导致结构错误的活动或条件仅存在于编织区域中, 因此, 下面证明织入操作不会在编织区域产生结构错误, 可以保证结构正确性.

定理 1. 结构正确性定理. 若基本过程和横切方面通知都通过结构正确性验证, 则按照方面织入正确性准则将方面织入基本过程后, 编织区域 g 是结构正确的.

证明. 令一个基本过程 $b = (P, T; F, M_0)$, 一个可信过程方面 $s = (i, d, J)$, 其中 $d = (P_s, T_s; F_s, T_e, T_x)$, $J_f \subseteq J$ 是弧切点集合, $j_f \in J_f$ 是弧切点集合中的一个弧切点, s 织入 b 得到编织区域 $g = (P_g, T_g; F_g)$. 根据方面编织规则定义, 编织结构分为顺序编织、选择编织、迭代编织和并发编织, 下面以顺序编织中的活动前编织为例证明编织后编织区域 g 中活动无伴随条件. 无孤立结点证明和无死锁、无陷阱证明类似, 在此省略.

根据顺序织入算法, 方面通知的入口活动集、出口活动集和基本过程切点弧的活动前后集条件改变, 假设任意 $t_g \in T_e$, 有 $t_g = dom(j_f.r) \in b.P, t_g \subseteq d.P$, 而 $b.P \cap d.P_s = \emptyset$, 因此, $t_g \cap t_g' = \emptyset$, 方面通知的入口活动无伴随条件, 同理可证方面出口活动和基本过程切点弧的活动无伴随条件. 证毕.

基于方面织入正确性准则, 在将方面织入基本过程时, 虽然在编织区域需要增加方面与基本过程之间的连接弧, 并删除多余弧, 但遵守方面织入正确性准则, 可以保证织入操作不会引入伴随条件、孤立结点、死锁和陷阱.

5.3.2 性质合理性

性质合理性要求横切方面织入基本过程时无性质错误, 同样假设横切方面通知和基本过程在编织前都满足定义 9 所定义的性质合理性, 方面织入基本过程后, 可能导致性质错误的活动或条件仅存在于编织区域中, 因此, 下面证明方面织入不会在编织区域产生性质错误. 由于横切方面通知在织入基本过程前无发生权, 因此, 要证明编织区域 g 的安全性和无冲撞性需要先证明方面通知在织入基本过程后是可发生的, 同时, 基本过程被织入方面后是持续的.

定义 15. 可发生性与持续性. 设 $s = (i, d, J)$ 是一个横切方面, $d = (P_s, T_s; F_s, T_e, T_x)$ 是方面 s 的通知, $b = (P, T; F, M_0)$ 是一个基本过程, 横切方面 s 按照方面织入正确性准则织入 p 后, 如果 p 是可发生的, b 中一定存在一个活动集合序列 $\delta_0, \delta_1, \delta_2, \dots, \delta_n$ 且存在 $M_1, M_2, \dots, M_n \subseteq b.P$, 有 $M_0[\delta_0 > M_1, M_1[\delta_1 > M_2, \dots, M_{n-1}[\delta_{n-1} > M_n$, 那么, s 织入 b 后, 其方面通知是可发生的且 b 是持续的.

定理 2. 可发生性定理. 若基本过程和横切方面通知中的活动是可发生的, 则按照方面织入正确

性准则将方面织入基本过程后, 方面通知中活动是可发生的。

证明. 令一个基本过程 $b=(P, T; F, M_0)$, 一个横切方面 $s=(i, d, J)$, 其中: $d=(P_s, T_s; F_s, T_e, T_x)$, $J_f \subseteq J$ 是弧切点集合, $j_f \in J_f$ 是弧切点集合中的一个弧切点。

为了证明 s 织入 b 后, s 的通知是可发生的, 下面分两个步骤进行论证。首先, 证明方面 s 的通知的入口活动集 T_e 有发生权, 在此基础上, 证明通知中所有其他活动都有发生权。

(1) 横切方面 s 的入口活动集 T_e 中的活动有发生权。

下面证明顺序织入中的活动前织入保证 T_e 中活动有发生权, 其他织入操作的证明类似, 在此省略。

根据顺序织入算法, 若 $dom(j_f) \in b.P$, 因为基本过程 b 是可发生的, 则一定存在 M' , 使 $dom(j_f.r) \in M'$, 那么, $M'[T_e]$, T_e 有发生权。

(2) 如果 T_e 有发生权, 证明 s 的通知中所有其他活动都有发生权。

每一个横切方面的过程通知都是使用白盒或黑盒建模方法^[39]实现。当应用白盒方法建模时, 方面通知中的一个活动 t_i 细化为一个基本块, 基本块由活动的顺序、并发、选择和迭代共 4 种结构构成。当应用黑盒方法建模时, 活动 t_i 细化为一个过程包, 隐藏其内部结构。Li^[39] 已经证明了接口一致性, 也就是说, 如果活动 t_i 有发生权, 那么细化后的基本块和过程包中的所有活动都有发生权, 且保持细化前的格局。也就是说, 横切方面的通知初始时 $T_e = T_x = d.T_s$, 上述论证已经证明了 T_e 有发生权, 此时, 过程通知没有其他活动, 过程通知是可发生的。当对方面通知进行细化操作后, 由 Li 的接口一致性定理^[39]可知, 方面通知仍然是可发生的。当多个横切方面基于方面间编织算法生成一个新的横切方面时, 按照方面间编织算法, 方面通知中的活动间存在顺序、选择或并发关系, 下面证明方面间编织算法让并发编织后的活动有发生权, 顺序和选择编织关系类似, 在此省略。

设 s_a 和 s_β 是两个横切方面, d_a 和 d_β 分别是它们的通知, 其中 $t_{s1}, \dots, t_{sm} \in d_a.T_x$, $t'_{s1}, \dots, t'_{sn} \in d_\beta.T_e$, 假设 $M_a[d_a.T_e]M_{a1}, \dots, M_{ay}[d_a.T_x]M'_a$ 且 $M_\beta[d_\beta.T_e]M_{\beta1}, \dots, M_{\beta z}[d_\beta.T_x]M'_\beta$ 。对于方面间并发关系编织, 按照算法 1, 如果将 d_a 和 d_β 并发编织后织入基本过程, 那么 v_1 有发生权, 令方面织入基本过程所处格局为 M , 那么 $M[v_1]M_a \cup M_\beta$ 因为 $M_a[d_a.T_e]M_{a1}, \dots,$

$M_{ay}[d_a.T_x]M'_a$ 且 $M_\beta[d_\beta.T_e]M_{\beta1}, \dots, M_{\beta z}[d_\beta.T_x]M'_\beta$, 有 $M'_a \cup M'_\beta[v_2]$, 并发编织后的过程通知中所有活动是可发生的。

证毕。

可发生性定理保证横切方面织入后基本过程的格局一定能让方面中的活动获得发生权, 即横切方面的通知是可发生的。下面的持续性定理则保证横切方面织入后基本过程中的活动仍有发生权, 即横切方面的织入不影响基本过程的持续性。

定理 3. 持续性定理. 若基本过程和横切方面通知中的活动是可发生的, 则按照方面织入正确性准则将方面织入基本过程后的基本过程是持续的。

证明. 令一个基本过程 $b=(P, T; F, M_0)$, 一个横切方面 $s=(i, d, J)$, 其中 $d=(P_s, T_s; F_s, T_e, T_x)$, $J_f \subseteq J$ 是弧切点集合, $j_f \in J_f$ 是弧切点集合中的一个弧切点。假设 s 在弧切点 $j_f \in J_f$ 处织入 b , $M \in R(M_0)$ 。对于横切方面的活动前织入, 根据算法 2, 如果在 s 织入 b 之前有 $M[t](t=cod(j_f.r))$, 那么, s 织入 b 之后, 有 $M[\sigma]M'$ 且 $M'[t]$, 基本过程是持续的。

其它顺序织入、选择织入、迭代织入和并发织入的证明类似。

证毕。

可发生性和持续性是横切方面织入基本过程应基本保持的合理性性质, 不满足这两条基本性质, 横切方面就不能织入基本过程。但是, 仅满足基本性质是不够的, 横切方面的织入还需要满足安全性和无冲撞性, 下面基于可发生性和持续性定理证明编织区域保持安全性, 无冲撞性证明类似。

定理 4. 安全性定理. 若基本过程和横切方面的通知都保持安全性, 则按照方面织入正确性准则将方面织入基本过程后编织区域 g 保持安全性。

证明. 令一个基本过程 $v=(P, T; F, M_0)$, 一个横切方面 $s=(i, d, J)$, 其中 $d=(P_s, T_s; F_s, T_e, T_x)$, $J_f \subseteq J$ 是弧切点集合, $j_f \in J_f$ 是弧切点集合中的一个弧切点, s 织入 b 得到编织区域 $g=(P_g, T_g; F_g)$, 下面用反证法证明。

假设 g 不满足安全性, 则存在条件 $p_g \in g.P_g$, 有 $L(p_g) > 1$ 。根据算法 2, 方面织入时只有方面通知入口条件的前集发生改变, 若 g 不满足安全性, 那么, 存在 $p_g \in T_e$, 有 $L(p_g) > 1$, 由于 $(T_e) = dom(j_f)$, 若 $L(p_g) > 1$, 必然有 $L(cod(j_f.r)) > 1$, 而 $cod(j_f.r) \in b.P$, 与基本过程 b 满足安全性矛盾, 因此, g 满足安全性。

证毕。

可发生性定理与持续性定理保证横切方面织入基本过程后横切方面通知中的活动可发生, 同时, 基

本过程是持续的,在此基础上,安全性定理保证织入方面后的业务过程在执行过程中,所有活动执行条件满足活动对资源条件的需求并能保证活动的正常执行。

5.3.3 行为一致性

同样,我们假设基本过程和横切方面通知的行为与过程规约定义的行为是一致的,当我们将横切方面织入基本过程后,方面织入不能使基本过程和方面通知的行为与过程规约定义的行为不一致。因为方面织入后,可能产生行为不一致的活动仅存在于编织区域中,因此,下面我们证明顺序织入中的活动前织入不会在编织区域产生行为不一致错误。

定理 5. 行为一致性定理. 若基本过程和横切方面的通知都满足过程规约定义的行为,则按照方面织入正确性准则将方面织入基本过程后,编织区域行为保持一致性。

证明. 令一个基本过程 $b = (P, T; F, M_0)$, 一个横切方面 $s = (i, d, J)$, 其中 $d = (P_s, T_s; F_s, T_e, T_x)$, $J_f \subseteq J$ 是弧切点集合, $j_f \in J_f$ 是弧切点集合中的一个弧切点。根据顺序织入中的活动前织入算法,方面织入只会改变切点弧相关活动的顺序和迭代行为关系,下面证明编织后的顺序行为关系与编织前顺序行为关系保持一致,迭代行为关系证明类似。

假设横切方面织入前,基本过程的切点弧前活动 $dom(j_f, r)$ 与活动 $t \in b.T$ 之间是直接顺序行为关系,有 $M[dom(j_f, r)] \text{ 但 } \rightarrow M[t]$, 而 $M[dom(j_f, r)] M' \rightarrow M'[t]$, 那么,织入横切方面后,根据织入算法,此时, $M[dom(j_f, r)] \rightarrow M[t]$, 对于 $\forall \sigma \in d.A^* \forall \sigma \in d.T^*$, $M[dom(j_f, r)] M' \rightarrow M'[\sigma] \wedge \rightarrow M'[t]$, 但 $M'[\sigma] M'' \rightarrow M''[t]$, 根据顺序行为关系定义 10 可知, $dom(j_f, r)$ 与活动 $t \in b.T$ 之间仍是顺序行为关系。如果 $dom(j_f, r)$ 和 t 在横切方面织入前是间接顺序行为关系,同理可证。证毕。

当横切方面织入基本过程时,在织入位置虽然会引入新的活动,但新引入的活动不会破坏基本过程中活动间的行为关系,即不会将原有顺序关系改变为选择、迭代或者并发关系,也不会将选择、迭代或者并发关系改变为其他关系,能够保证行为一致性。

5.3.4 方面间编织完整性与正确性

根据方面间编织完整性定义,编织完整性可以通过保证所有织入的横切方面具有发生权且在共享切点上所有存在数据依赖关系的横切方面按照数据依赖关系顺序融合来实现。上述可发生性定理已经证明横切方面具备发生权,因此,下面证明共享切点

上所有存在数据依赖关系的横切方面如果按照数据依赖关系顺序融合,则保证了编织完整性。为描述简洁,针对数据依赖关系,下面将决定其他横切方面的横切方面称为被依赖方面,而依赖于其他横切方面的横切方面称为依赖方面。

定理 6. 完整性定理. 设 $S = \{s_1, \dots, s_n\}$ 是一个共享切点的横切方面集合,若 S 中所有方面织入基本过程,其中所有存在数据依赖关系的横切方面都按照依赖关系顺序融合,那么编织是完整的。

证明. 设横切方面集合 $S = \{s_1, \dots, s_n\}$ 中任意两个横切方面 s_1 和 s_2 中活动 t_{s_1} 和 t_{s_2} 之间是数据依赖关系 $t_{s_1} \theta^d t_{s_2}$, 根据方面间编织算法, s_1 和 s_2 以顺序关系编织后织入基本过程, s_1 和 s_2 的通知在织入后都是可发生的(上述定理 2 证明了方面通知的可发生性),那么,存在格局 M 和 M' , 有 $M[t_{s_1}] \text{ 但 } \rightarrow M[t_{s_2}]$, $M[t_{s_1}] M' \rightarrow M'[t_{s_2}]$, 在格局 M , t_{s_1} 有发生权, t_{s_2} 没有发生权,但 t_{s_1} 发生后进入格局 M' , 此时, t_{s_2} 获得发生权以及需要的资源,意味着 t_{s_2} 正常执行且不会丢失其应有的影响,因此,编织是完整的。证毕。

同样,根据方面间编织正确性定义,编织正确性可以通过保证所有织入基本过程的横切方面具有发生权且共享切点上所有存在控制依赖关系的横切方面按照控制依赖关系实施选择编织来实现。上述可发生性定理已经证明横切方面的活动具备发生权,对于存在控制依赖关系的横切方面,如果被控制方面未按照控制关系织入基本过程,那么被控制方面将造成错误的方面影响,等同于被控制方面错误的织入基本过程,此时编织是不正确的。因此,存在控制依赖关系的横切方面按照控制依赖关系正确地实施选择编织后织入基本过程,可以保证编织的正确性。

定理 7. 正确性定理. 设 $S = \{s_1, \dots, s_n\}$ 是一个共享切点的横切方面集合,若 S 中所有方面织入基本过程,其中所有存在控制依赖关系的横切方面都按照控制依赖关系实施选择编织,那么编织是正确的。

定理 7 的证明与定理 6 类似,为描述简洁,在此省略。

5.4 方面织入基本过程正确性检测

通过面向方面业务过程建模正确性证明可知,当横切方面织入基本过程时,如果按照定义及设计的方面织入准则进行建模,则可以保证过程模型结构正确性、性质合理性和行为一致性。但在实际建模过程中,仍然存在因疏忽而导致引入错误和异常的可能性。因此,下面针对 4.3 节提出的方面织入基本

过程正确性检测方法,在面向方面建模过程中进行织入正确性检测。

设 $g=(P_g,T_g;F_g)$ 是横切方面 $s=(i,d,J)$ 织入基本过程 $b=(P,T;F,M_0)$ 产生的一个编织区域,其中: $d=(P_s,T_s;F_s,T_e,T_x)$ 是 s 的通知。针对横切方面织入有可能引入的 4 类性质错误,下面给出性质错误检测方面和性质错误检测算法,结构错误检测与行为错误检测类似,在此省略。

按照 4.3 节错误检测方面定义 13,定义持续性错误检测方面 $s_w=(i_w,d_w,J_w)$,其中: i_w = “Persistent” 定义方面标识,方面通知定义为 d_w = “可能出现基本过程持续性错误,请查看”,方面切点定义为 $J_w=\{\exists p_g \notin M, p_g \in T\}$ 。可发生性错误、安全性错误及冲撞错误方面定义类似。基于错误检测方面定义,算法 3 给出错误检测算法。

算法 3. 性质正确性检测算法 Property_Detection.

输入: s, b, g

输出: 如果 g 不满足方面织入性质正确性,输出错误提示

BEGIN

 计算 g 的可达标识集 $R(M_0)$;

 FOR $\forall M \in R(M_0)$ DO

 BEGIN

 FOR $\forall p_g \in P_g$ DO

 BEGIN

 IF p_g 满足 J_w 中所有条件 THEN

 /* 错误检测方面切点 J_w 定义了错误模式 */

 RETURN $d_w + p_g +$ “附件节点”

 /* 检测到错误,给出提示 */

 END;

 FOR $\forall t_g \in T_g$ DO

 BEGIN

 IF t_g 满足 J_w 中所有条件 THEN

 RETURN $d_w + t_g +$ “和” + $p_g +$ “附近节点”

 END

 END

END

根据 4.3 节定义的错误检测方面,算法 3 提出的方面织入错误检测算法能够根据错误检测方面定义的错误模式找出并定位可能出现的织入错误,并以消息提示方式显示错误定位与错误类型,建模者可以根据提示消息对定位到的错误位置,按照错误类型进行错误处理。

5.5 面向方面业务过程建模工具

按照面向方面业务过程建模框架,为支持面向方面业务过程建模,我们基于 Petri 网建模开源工具 PIPE^① 设计并实现了一个面向方面业务过程建模工具 TPAT(Trustworthy business Process Aided Tool),TPAT 使用插件技术将横切方面织入基本过程,并在织入过程中对织入错误进行检测,实现面向方面扩展后的 TPAT 模块间关系如图 6 所示。

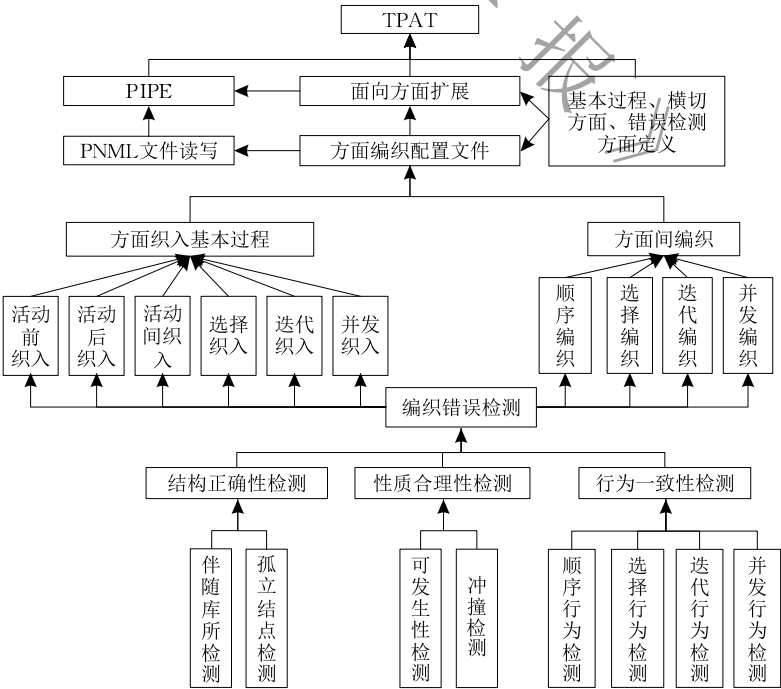


图 6 TPAT 模块设计

① PIPE2 (Platform Independent Petri Net Editor Beta). <http://pipe2.sourceforge.net/>

TPAT 由两大核心组件构成,分别是基本过程与方面定义组件和面向方面扩展组件,其中面向方面扩展组件由方面间编织、方面织入基本过程和编织错误检测 3 个核心模块构成,图 7 展示了使用面向方面业务过程建模工具 TPAT 的主要功能。

面向方面业务过程建模总体分两个主要阶段:第 1 阶段是定义模型基本元素,即在新建面向方面业务过程建模工程中定义基本过程、横切方面和错误检测方面(如图 7 界面 1 所示)以及相关参数(如

图 7 界面 2 所示),定义完成后,调用在 PIPE 中扩展的插件(如图 7 界面 3 所示),按照方面切点定义,将方面间编织和方面织入基本过程的相关配置信息写入方面编织配置文件(如图 6 所示).在调用 PIPE 展示面向方面业务过程模型时,按照错误检测方面定义,基于 4.3 节提出的织入正确性检测方法和 5.4 节提出的对应算法,对编织进行正确性检测,如果检测出错误,如图 7 界面 4 所示,过程模型存在冲突,则给予错误提示(如图 7 界面 5 所示),否则生成并输出面向方面业务过程模型。

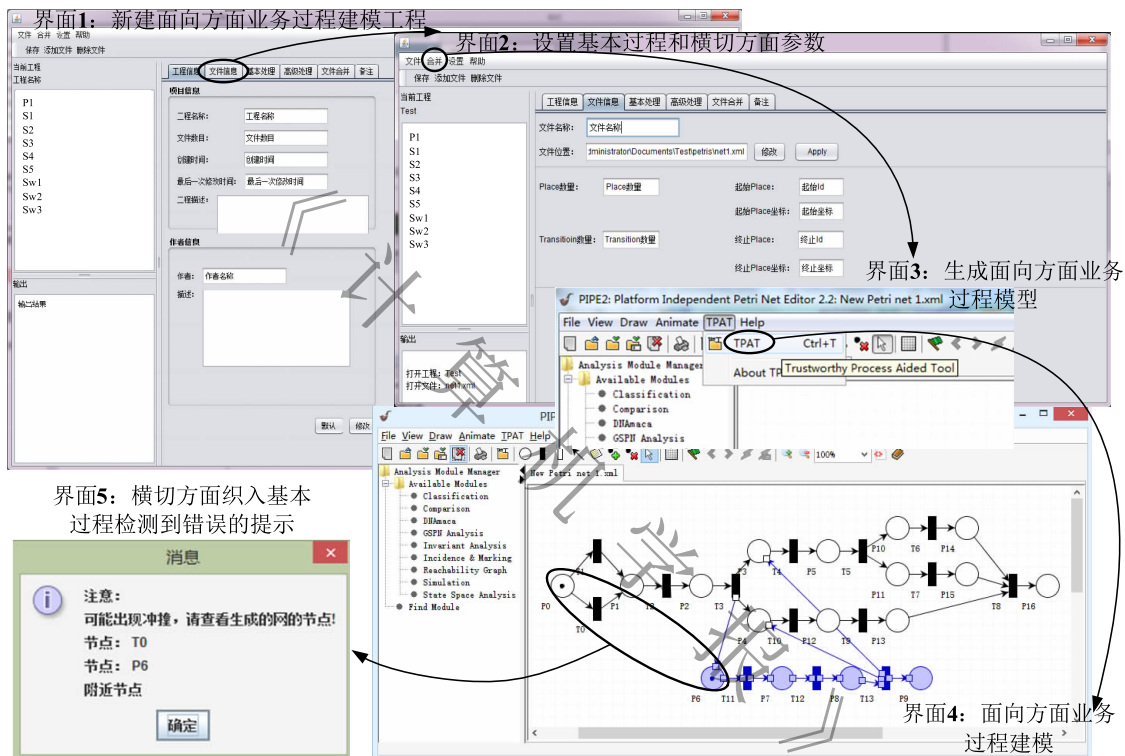


图 7 面向方面业务过程建模工具

6 案例分析

基于面向方面业务过程建模方法及正确性控制与检测分析,下面对一个真实的银行业务过程,投资交易过程,进行面向方面的建模与分析,这个交易过程数据来源于一个银行的 1000 余个分行^[48]。

投资交易过程描述了银行的投资活动,交易过程起始于交易员提交投资申请,如果投资交易金额没有超过限额则直接提交,如果金额超过限额,则需要责任交易员和部门经理对申请进行确认,如果申请被否则将申请归档,并结束交易过程,如果申请得到确认,则继续投资交易,交易员填写交易单并提交给责任交易员和部门经理签字并归档.在交易单

归档后,根据交易单,银行结算部门生成结算单,并提交交易员、责任交易员和部门经理签字.同时,银行结算部门生成结算数据,结算数据由部门经理生成交易订单后提交后台部门,同时提交行间交易机构,在分别处理完交易后归档,并连同结算单生成货币兑换凭据,将兑换凭据归档后完成整个交易过程.使用 Petri 网语言建模这个过程得到的过程模型如图 8 所示。

在这个银行交易业务过程中,存在着大量的验证、签署、控制、约束和归档等横切活动,这些横切活动用以保证基本交易活动的安全及可追溯.然而,众多相同的横切活动分散在基本过程的不同位置,并与基本过程的活动交织在一起,增加了过程模型的复杂度,不利于业务过程的维护和重用.将这些横切

活动在面向方面方法中封装为横切方面并与基本业务过程分离,分离后投资交易过程的基本业务过程和横切方面如图 9 和图 11 所示,横切方面编织入基本过程可得到的过程模型如图 10 所示. 图 9 是基本过程,基本过程定义为 $b = \{P, T; F, M_0\}$, 其中:

$$P = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}\},$$

$$T = \{JDOpenthePosition, JDDDeal, JDFillDealSlip, SDReceiveNT300Swift, SDProvideSwiftDraft, SDSendtoGM, SDSendSwift, GMOrdertoBackOffice, GMOrdertoDealingRoom, BOERegisterVoucher\},$$

$$F = \{(p_1, JDOpenthePosition), (JDOpenthePosition, p_2), (p_2, JDDDeal), (JDDDeal, p_3), (p_3, JDFillDealSlip), \dots, (BOERegisterVoucher, p_{13})\},$$

$$M_0 = \{p_1\}.$$

为描述简洁,基本过程中部分弧的定义被省略了,这些省略的弧定义与已给出的弧定义类似.

图 11 是横切方面,为描述横切方面与基本过程的关系,图 9 中的六边形代表横切方面,横切方面在基本过程的位置大致描述了这些横切方面的切点位置. 以横切方面 s_1 为例, s_1 定义为 $s_1 = \{i, d, J\}$, 其中

$$i = 1$$

$$d = \{P_s, T_s; F_s, T_e, T_x\}$$

$$J = \{(b, JDOpenthePosition, 1)\}$$

$$P_s = \{p_{s1}, p_{s2}\}$$

$$T_s = \{GMConfirm\},$$

$$F_s = \{(p_{s1}, GMConfirm), (GMConfirm, p_{s2})\},$$

$$T_e = T_x = \{GMConfirm\}$$

其它横切方面定义类似.

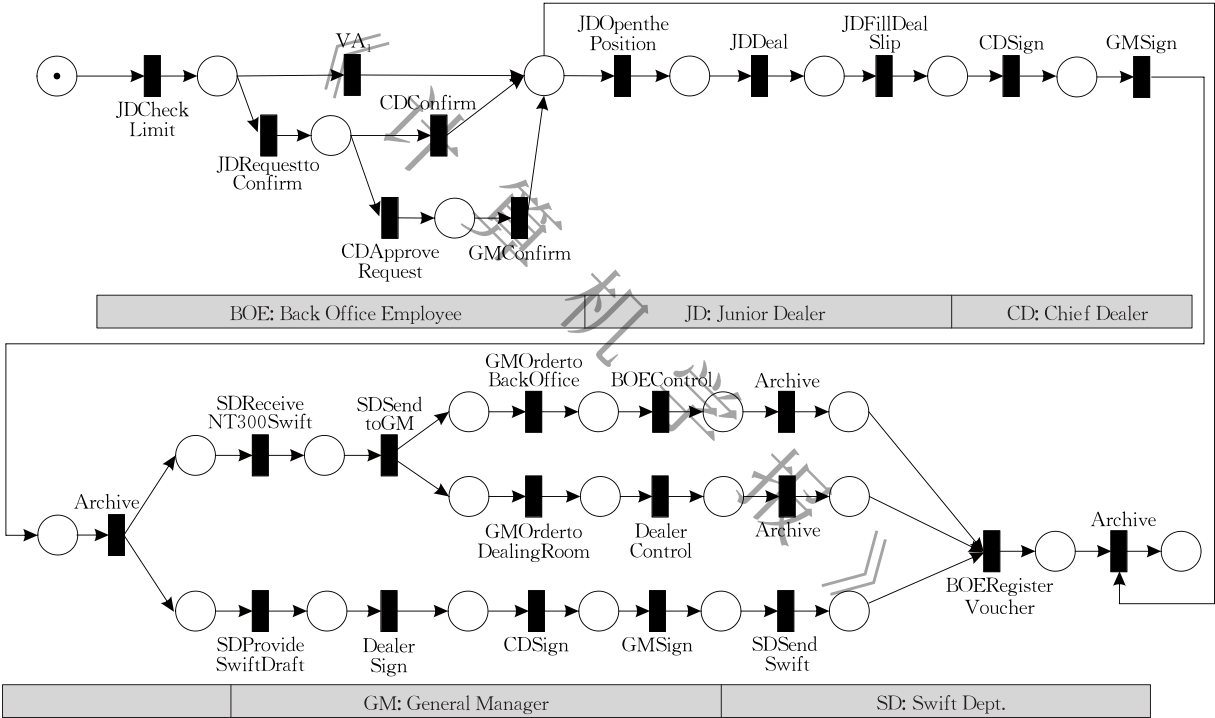


图 8 银行交易业务过程

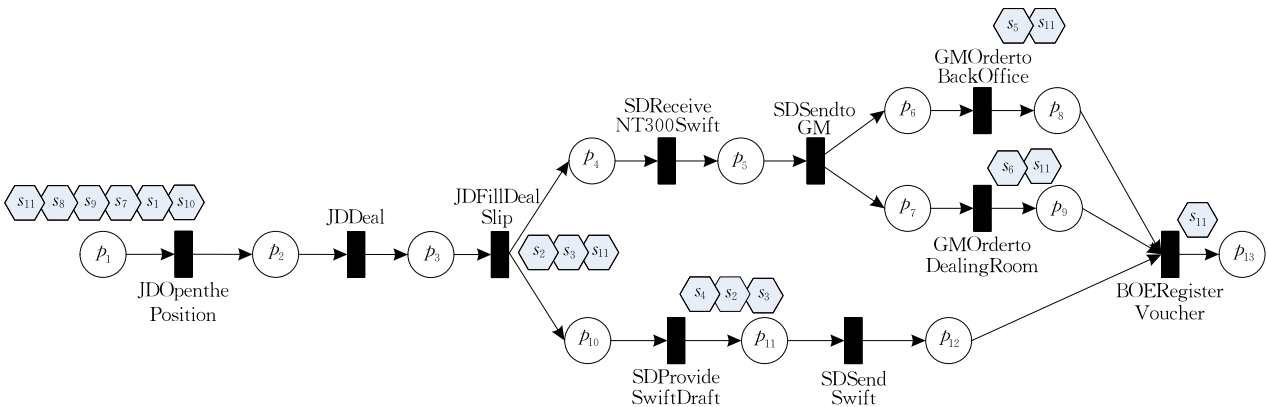


图 9 基本过程

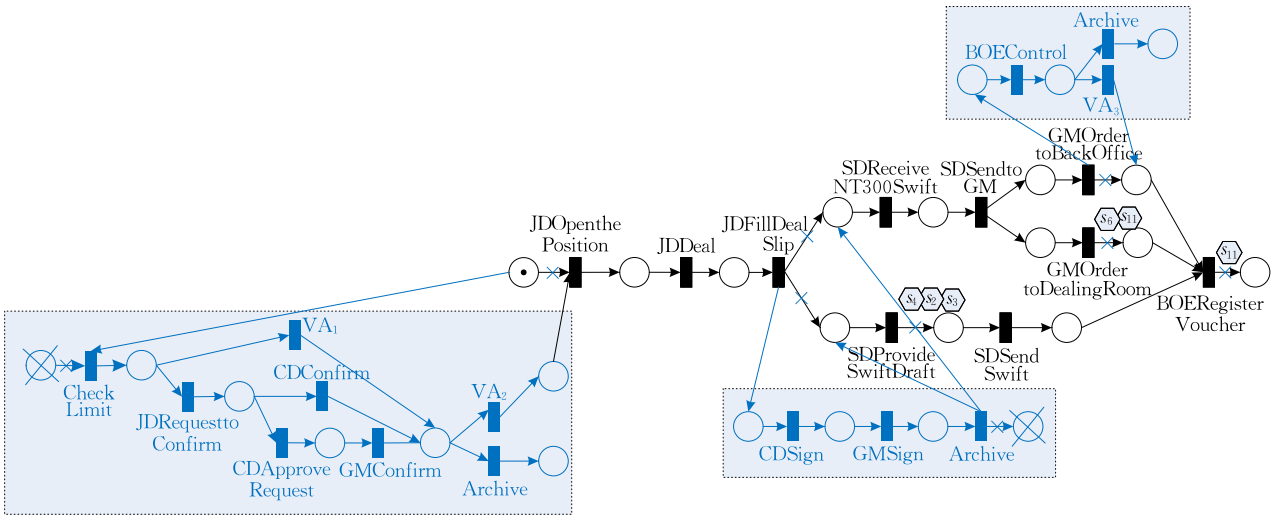


图 10 面向方面业务过程模型

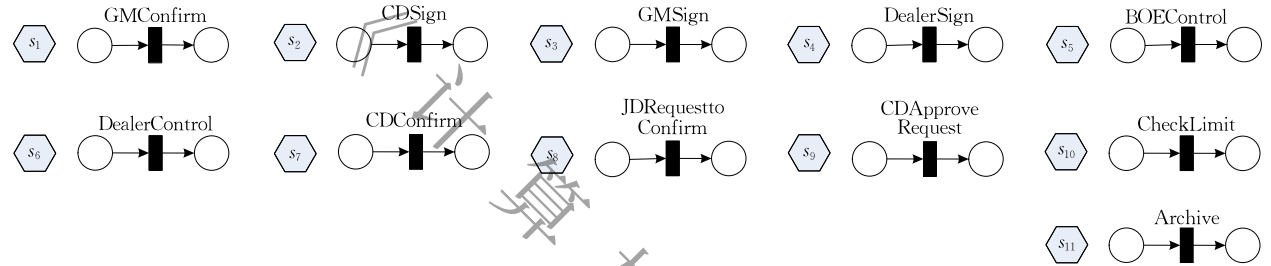


图 11 横切方面

通过对比图 8 和图 9,基本过程与横切方面的分离使过程模型清晰且复杂度明显降低,更重要的是,有利于基本过程和横切方面的重用,对有不同需求的业务过程按照各自规模和需求重用不同的基本过程或编织不同的横切方面提供了方便.例如:如果需要对货币汇兑交易过程进行建模,此交易过程处理不同货币之间的转换,交易起始于后台员工填写汇兑交易表,提交部门经理,如果部门经理否决,则交易结束并归档,如果部门经理同意,汇兑交易表将提交给交易员填写交易单,后续过程执行与投资交易具有同样的过程.由于与投资交易过程采用了大部分相同的基本过程和横切方面,因此,可以重用投资交易过程模型大部分的基本过程和横切方面.另外,如果对银行汇票交易过程进行建模,此交易过程是银行签发汇票并兑付的过程,汇票签发起始于支付者提交签发申请并提供银行账户信息,银行对支付者进行测试汇票验证,如果未通过,则申请被否决并在申请归档后结束签发过程,如果验证通过则生成银行汇票信息,提交交易员填写交易单,后续基本过程同样采用与投资交易一样的过程,但由于汇票可以延时兑付并且需要支付者的信用担保,因此,与投资交易过程相比,横切方面除了需要安全和日志

记录外,还需要追踪和审计的横切方面.因此,汇票交易过程建模相对货币汇兑交易过程,可以重用较少的投资交易基本过程,同时需要增加投资交易过程模型不需要的追踪和审计横切方面.再如,银行为国际贸易提供的信用证交易过程,同样与上述 3 个交易的基本过程相似,但横切方面主要以保证安全性为主.总之,在定义了基本过程和横切方面后,银行业务过程可以通过不断重用相似基本过程和横切方面进行面向方面业务过程建模.

下面,以投资交易过程为例进行面向方面业务过程建模.首先,根据横切方面定义,我们将同一切点上相同编织类型的多个横切方面按照方面间依赖关系编织为一个横切方面,然后按照方面织入准则将横切方面织入基本过程,并在织入过程中进行方面编织正确性检测.

6.1 方面间编织

如图 9 所示,所有横切方面(图中用六边形表示)在基本过程的 7 个切点位置织入基本过程,其中有 6 个切点存在多个横切方面需织入,按照面向方面业务过程建模,首先将同一切点上相同织入类型的横切方面实施方面间编织.基于方面间编织算法,首先分析这些同切点同织入类型横切方面间的依赖

关系,在切点 JDFillDealSlip,SDProvideSwiftDraft, GMOrdertoBackOffice 和 GMOrdertoDealingRoom 上的横切方面仅有数据依赖关系,在 BOEFillPositionSheet 和 JDRequesttoConfirm 上的横切方面间既有数据依赖关系也有控制依赖关系。

下面以切点 JDFillDealSlip 和 JDOpenthePosition 上横切方面间编织为例,由于 JDFillDealSlip 切点上的横切方面 s_2, s_3 和 s_{11} 之间仅有数据依赖关系(如图 12 所示,其中 data 表示数据依赖关系),按照方面间编织算法, s_2, s_3 和 s_{11} 按照数据依赖关系顺序编织为如图 13 所示的 Petri 网. 与此类似, JDOpen- thePosition 切点上横切方面 $s_1, s_7, s_8, s_9, s_{10}$ 和 s_{11} 之间按照数据依赖关系和控制依赖关系(如图 14 所示,

其中 cntr 表示控制依赖关系)编织的结果如图 15 所示。

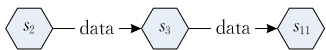


图 12 JDFillDealSlip 切点上横切方面依赖关系



图 13 JDFillDealSlip 切点上横切方面编织结果

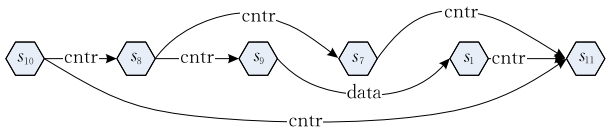


图 14 JDOpenthePosition 切点上横切方面依赖关系

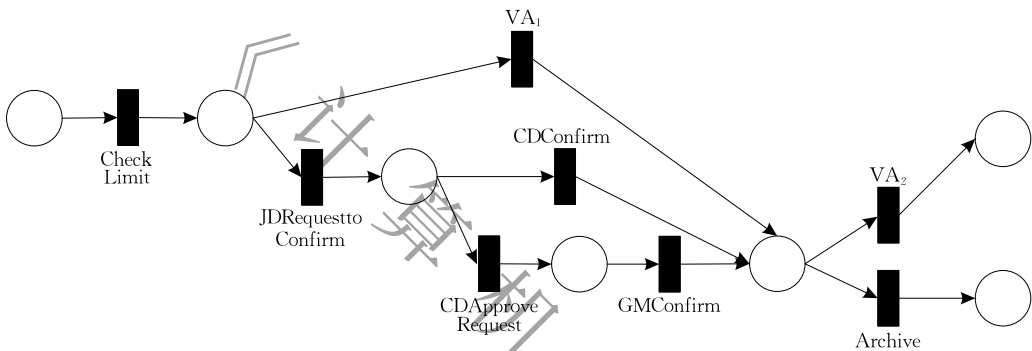


图 15 JDOpenthePosition 切点上横切方面编织结果

在所有切点上有相同织入类型的方面编织后, 下面将编织后方面织入基本过程。

6.2 方面织入基本过程

当横切方面织入基本过程时,需要按照方面织入正确性准则进行织入,否则方面织入会引入错误或导致异常,例如:在基本过程(图 16 中实线框内

Petri 网)切点 JDOpenthePosition,图 15 所示编织后方面以活动前方式织入,如果不按照方面织入正确性准则,编织结果如图 16 所示,横切方面中的活动无发生权,基本过程也不满足持续性. 如果按照方面织入正确性准则,如图 17 所示的编织结果可以保证方面中的活动有发生权且基本过程满足持续性。

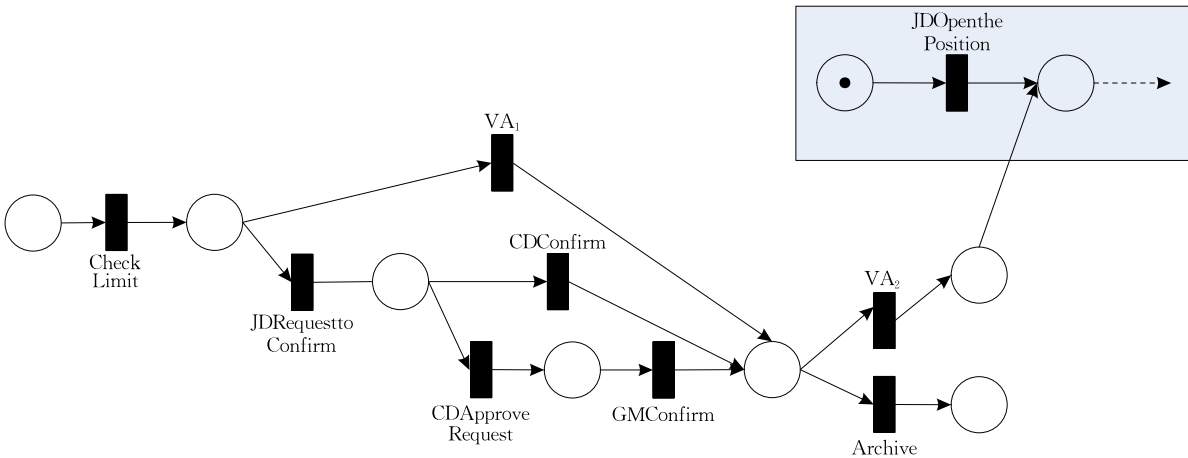


图 16 不按照方面织入正确性准则织入横切方面

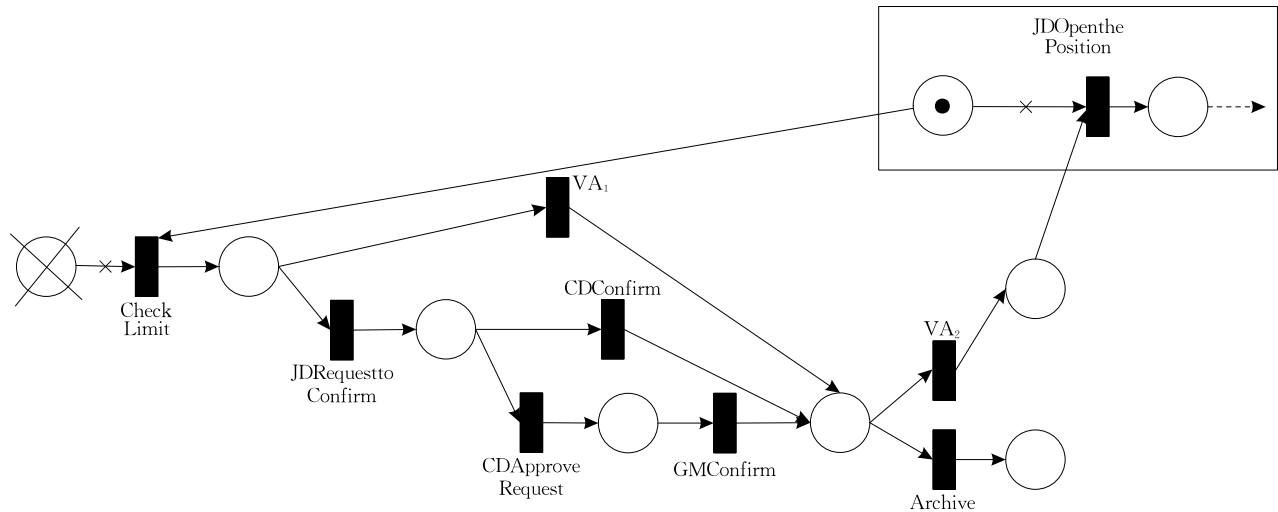


图 17 按照方面织入正确性准则织入横切方面

在方面织入基本过程中,如果错误检测方面检测到方面织入错误,则弹出如图 18 所示的错误提示窗口,指出错误位置,待所有可能引入的错误解决后,图 10 给出了在 3 个切点位置织入方面的结果,其他切点织入结果类似.

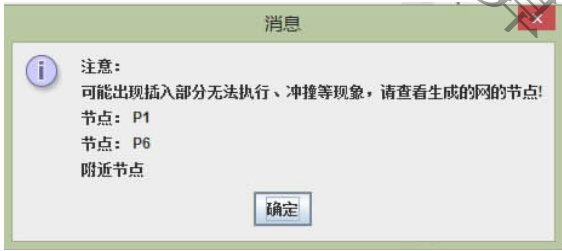


图 18 织入错误提示

案例分析结果表明,面向方面业务过程建模有 3 个优势:一是分离后的基本过程与横切方面有利于重用和定制,减少业务过程模型库中相似重复的过程模型存储,解决大量相似模型管理复杂的问题;二是有利于业务过程的柔性建模,可以按照不同业务应用领域需求灵活织入不同横切方面实现按需裁剪,方便业务过程改进;三是在业务过程建模过程中,使用正确性准则与正确性检测可以有效保证建模得到的业务过程模型的正确性,减少后期验证修复过程模型产生的成本、人员及时间损失.

6.3 性能评估

通过第 5 节的证明与本节案例分析,所提出的方法可以达到预期目标,然而,在实际使用过程中,分离横切活动与核心活动,可以增加多大的过程模型重用比例?分别对基本过程、横切方面通知和编织区域进行正确性检测,是否会影响业务过程建模的整体性能与效率?面向方面业务过程建模是否会增

加过多的代码量?下面,通过度量重用比例、正确性检测时间性能和增加的代码量这 3 项指标,评估面向方面业务过程建模方法的性能与工作量.

为统计重用比例,下面对银行业务过程案例中的 4 个过程模型进行重用比例分析,如表 2 所示.

表 2 重用率

业务过程模型	基本过程重用活动数量	基本过程重用比例/%	横切方面重用数量	横切方面重用比例/%
$b_{s1}b_{s2}$	9	90	10	95
$b_{s1}b_{s3}$	7	58	10	83
$b_{s1}b_{s4}$	7	60	9	90
$b_{s2}b_{s3}$	7	58	10	88
$b_{s2}b_{s4}$	7	60	9	95
$b_{s3}b_{s4}$	7	48	9	84

b_{s1} 、 b_{s2} 、 b_{s3} 和 b_{s4} 分别是投资交易过程、货币汇兑交易过程、银行汇票交易过程和信用证交易过程.表 4 所示重用率是 4 个业务过程两两比较的结果.由于所有业务过程中涉及银行结算的基本过程都是可重用的,而结算过程包含 7 个核心活动,因此,基本过程中可重用的活动数量至少为 7,其中:投资交易过程和货币汇兑交易过程除了初始活动不同外,其他 9 个核心活动组成的业务过程是一致的,因此,重用比例最高.

横切方面主要分 4 类:安全横切方面、日志横切方面、追踪横切方面和审计横切方面,根据 4 个业务过程的需求,投资交易过程和货币汇兑交易过程需编织安全横切方面和日志横切方面,银行汇票交易过程需编织所有横切方面,而信用证交易过程仅需编织安全横切方面,因此,横切方面重用比例按此计算得到结果如表 2 所示.

当然,并不是所有银行业务的基本过程都可以

重用,例如:贷款业务过程和本案例中的四个业务过程均没有相同的核心活动,无法重用基本过程,但是,部分横切方面是可以重用的,因此,面向方面业务过程建模可以提高重用率.然而,为实现面向方面业务过程建模,需要增加额外代码量存储横切方面定义及方面编织配置信息,因此,下面分析需要增加的代码量.

如表 3 所示,如果仅对投资交易过程进行面向方面建模,那么,面向方面投资交易过程模型 b_{s1} 的代码增加量为 33.8%,如果对所有 4 个银行业务过程进行面向方面建模,代码增加量降低为 29.6%,此时,代码增加量的降低是因为模型重用.因此,采用面向方面方法对业务过程进行建模会增加一定的代码量,但随着模型的不断重用,代码增加量会逐渐减少.

表 3 增加代码量				
业务过程模型	评估指标	非面向方面业务过程建模方法	面向方面业务过程建模方法	
		代码量	增加代码量	增长量/%
b_{s1}	代码量(LOC)	1597	815	33.8
$b_{s1}b_{s2}$	代码量(LOC)	7516	3161	29.6
$b_{s3}b_{s4}$	XML Tags 数量	50	10	16.7

另外,与非面向方面业务过程建模相比,如果要完成过程模型的正确性检测,首先需要对基本过程和横切方面通知进行正确性检测,在此基础上,再对编织区域进行正确性检测,如表 4 给出的面向方面建模方法与非面向方面建模方法正确性检测时间性能的比较.

表 4 检测时间				
业务过程模型	非面向方面业务过程模型检测	面向方面业务过程模型检测		增长时间/%
		基本过程与横切方面检测	编织区域检测	
		执行时间/ms	执行时间/ms	
b_{s1}	2801.7	1510.2	1259.1	-3.30
b_{s2}	2787.6	1503.0	1171.7	-4.05
b_{s3}	3125.4	1779.8	1024.2	-10.28
b_{s4}	2560.1	1477.9	991.7	-3.54

通过表 4 的检测时间性能比较可以看出,由于汇票交易过程模型 b_{s3} 的规模要相对大于其他过程模型,因此,在进行正确性检测时,检测时间较优于非面向方面业务过程模型,主要原因是面向方面方法分解了业务过程模型,使其复杂度降低,其它 3 个过程模型因为规模相对较小,检测时间性能没有明显差异,但如果能够重用过程模型,重用的过程模型将大大减少检测时间.

7 总结与展望

传统的业务过程模型将横切关注点活动与业务基本活动交织建模,导致业务过程模型复杂、逻辑行为不清晰,而随着业务工作的愈加复杂化以及不断积累的过程模型,大量相似过程模型储存下来,难于管理.面向方面方法通过分离关注点,将不同功能的活动分开建模,可以根据具体应用领域需要,重用已有过程模型并实现业务过程模型的柔性建模.基于此思想,本文提出面向方面业务过程建模方法,本文的主要贡献如下:

(1)采用关注点分离的思想,定义面向方面业务过程元模型,将不同关注点的过程活动模块化分离,在此基础上,提出面向方面编织方法,可以灵活地根据不同的模型需求将模块化过程模型编织为一个过程模型,不仅增强业务过程建模方法的易理解性和易用性,而且,分离的过程模块还可以在不同建模需求业务过程中实现重用,方便业务过程模型的动态变更.

(2)定义横切方面编织正确性,设计方面间编织正确性准则和方面织入基本过程正确性准则,并使用面向方面方法定义业务过程建模正确性检测方法,通过在业务过程建模过程中使用正确性准则,并进行正确性检测保证在建模完成后得到的业务过程模型是正确的.

本文的主要创新工作主要体现为:

(1)为业务过程建模领域提出完整的面向方面业务过程建模与正确性保证解决方案.

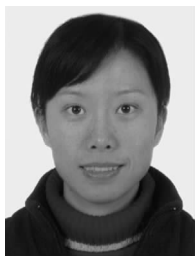
(2)基于面向方面思想提出错误检测方面用于过程模型错误检测,可以方便建模者通过错误检测方面灵活定义检测需求,并且根据实际业务需要使用错误检测方面进行过程模型的错误检测.

今后,在两个方面还需进一步开展相关研究工作:首先,横切关注点活动与基本业务过程的分离是面向方面业务过程建模的基础,面向已有业务过程模型,识别横切关注点活动并实现这些横切活动的分离是下一步需要研究的工作之一;其次,基于基本 Petri 网定义面向方面业务过程模型时,横切方面存在活动相同但执行者不同的情况,对这些横切方面仍然需要进一步研究更高抽象层次的模块化方法,进一步细化模型复用粒度.

参 考 文 献

- [1] Dijkman R, La Rosa M, Reijers H A. Managing large collections of business process models: Current techniques and challenges. *Computers in Industry*, 2012, 63(2): 91-97
- [2] Odgers B, Thompson S. Aspect-oriented process engineering (ASOPE)//*Proceedings of the Workshop on Object-Oriented Technology*. London, UK; Springer-Verlag, 1999: 295-299
- [3] Reichert M, Weber B. Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies. Berlin Heidelberg, Germany: Springer-Verlag, 2012
- [4] Yu Yang, Tang Yong, Pan Mao-Lin, et al. Temporal workflow process model and its soundness verification. *Journal of Software*, 2010, 21(6): 1233-1253(in Chinese)
(余阳, 汤庸, 潘茂林等. 时态工作流过程模型及其合理性验证. *软件学报*, 2010, 21(6): 1233-1253)
- [5] Barkaoui K, Ayed R B. Uniform verification of workflow soundness. *Transactions of the Institute of Measurement and Control*, 2011, 33(1): 133-148
- [6] van der Aalst Wil M P. Business process management: A comprehensive survey. *ISRN Software Engineering*, 2013, article ID 507984, 37 pages
- [7] Clempner J. Verifying soundness of business processes: A decision process Petri nets approach. *Expert Systems with Applications*, 2014, 41(11): 5030-5040
- [8] Clempner J. An analytical method for well-formed workflow/Petri net verification of classical soundness. *International Journal of Applied Mathematics and Computer Science*, 2014, 24(4): 931-939
- [9] Song Wei, Ma Xiao-Xing, Hu Hao, et al. Dynamic evolution of processes in process-aware information systems. *Journal of Software*, 2011, 22(3): 417-438(in Chinese)
(宋巍, 马晓星, 胡昊等. 过程感知信息系统中过程的动态演化. *软件学报*, 2011, 22(3): 417-438)
- [10] van der Aalst W M P. The application of Petri nets to workflow management. *Journal of Circuits, Systems, and Computers*, 1998, 8(1): 21-66
- [11] Sun P, Jiang C J. Analysis of workflow dynamic changes based on Petri net. *Information and Software Technology*, 2009, 51: 284-292
- [12] Weber B, Reichert M, Rinderle-Ma S. Change patterns and change support features—Enhancing flexibility in process-aware information systems. *Data & Knowledge Engineering*, 2008, 66(3): 438-466
- [13] Bachmendo B, Unland R. Aspect-based workflow evolution // *Proceedings of the Workshop on Aspect-Oriented Programming and Separation of Concerns*. Lancaster, UK, 2001: 21-27
- [14] Cappelli C, Santoro F M, do Prado Leite J C S, et al. Reflections on the modularity of business process models. *Business Process Management Journal*, 2010, 16(4): 662-687
- [15] Santos F J N, Cappelli C, Santoro F M, et al. Aspect-oriented business process modeling: Analyzing open issues. *Business Process Management Journal*, 2012, 18(6): 964-991
- [16] Yu E. Towards modeling and reasoning support for early-phase requirements engineering//*Proceedings of the 3rd IEEE International Symposium on Requirements Engineering*. Annapolis, USA, 1997: 226-235
- [17] Amyot D, Ghanavati S, Horkoff J, et al. Evaluating goal models within the goal-oriented requirement language. *International Journal of Intelligent Systems*, 2010, 25: 841-877
- [18] do Prado Leite J C S, Santoro F M, Cappelli C, et al. Ownership relevance in aspect-oriented business process models. *Business Process Management Journal*, 2016, 22(3): 566-593
- [19] Pourshahid A, Mussbacher G, Amyot D, Weiss M. Toward an aspect-oriented framework for business process improvement. *International Journal of Electronic Business (IJEB)*, Inderscience Publishers, 2010, 8(3): 233-259
- [20] Pourshahid A, Mussbacher G, Amyot D, Weiss M. Requirements for a modeling language to specify and match business process improvement patterns//*Proceedings of the 3rd International Model-Driven Requirements Engineering Workshop (MoDRE 2013)*. Rio de Janeiro, Brazil, 2013: 10-19
- [21] Amyot D, Mussbacher G. User requirements notation: The first ten years, the next ten years. *Journal of Software*, 2011, 6(5): 747-768
- [22] Charfi A, Mezini M. Aspect-oriented workflow languages. On the move to meaningful Internet systems 2006: CoopIS (Cooperative Information Systems), DOA (Distributed Objects and Applications), and ODBASE (Ontologies, DataBases, and Applications of Semantics). *Lecture Notes in Computer Science* 4275. Montpellier, France, 2006: 183-200
- [23] Charfi A, Mezini M. AO4BPEL: An aspect-oriented extension to BPEL. *World Wide Web Journal: Recent Advances on Web Services (Special Issue)*, 2007, 10(3): 309-344
- [24] Charfi A, Schmeling B, Mezini M. Transactional BPEL processes with AO4BPEL aspects//*Proceedings of the 5th IEEE European Conference on Web Services*. Halle, Germany, 2007: 149-158
- [25] Charfi A, Müller H, Mezini M. Aspect-oriented business process modeling with AO4BPMN//Kühne T, Selic B, Gervais M-P, Terrier F eds. *Modelling Foundations and Applications*. *Lecture Notes in Computer Science* 6138. Berlin Heidelberg, Germany: Springer-Verlag, 2010: 48-61
- [26] Witteborg H, Charfi A, Collell D C, et al. Weaving aspects and business processes through model transformation//Villari M, et al, eds. *Service-Oriented and Cloud Computing*. *Lecture Notes in Computer Science* 8745. Berlin Heidelberg, Germany: Springer-Verlag, 2014: 47-61
- [27] Park R, Choi H, Lee D, et al. Knowledge-based AOP framework for business rule aspects in business process. *ETRI Journal*, 2007, 29(4): 477-488

- [28] Jalali A. Static weaving in aspect oriented business process management//Proceedings of the 34th International Conference on Conceptual Modeling. Stockholm, Sweden, 2015: 548-557
- [29] Jalali A, Wohed P, Ouyang C, et al. Dynamic weaving in aspect oriented business process management//Proceedings of the Move to Meaningful Internet Systems Conferences. Berlin Heidelberg, Germany, 2013: 2-20
- [30] Jalali A, Maggi F M, Reijers H A. Enhancing aspect-oriented business process modeling with declarative rules//Proceedings of the 34th International Conference of Conceptual Modeling. Lecture Notes in Computer Science 9381. Stockholm, Sweden, 2015, 10: 108-115
- [31] Chen Xin, Huang Chao, Zhang Yi-Fan, et al. A tool for analyzing interference problems in aspect-oriented designs. Journal of Software, 2016, 27(3): a10(in Chinese)
(陈鑫, 黄超, 张一帆等. 一个面向方面设计中干涉问题的分析工具. 软件学报, 2016, 27(3): a10)
- [32] Roubtsova E E, Aksit M. Extension of Petri nets by aspects to apply the model driven architecture approach//Proceedings of the 1st International Workshop on Aspect-Based and Model-Based Separation of Concerns in Software-Systems (ABMB). Nuremberg, Germany, 2005, 1: 1-15
- [33] Xu D X, Nygard K E. Threat-driven modeling and verification of secure software using aspect-oriented Petri nets. IEEE Transactions on Software Engineering, 2006, 32(4): 265-278
- [34] Guan L, Li X, Hu H, Lu J. A Petri net-based approach for supporting aspect-oriented modeling. Frontiers of Computer Science in China, 2008, 2(4): 413-423
- [35] Molderez T, Meyers B, Janssens D, Vangheluwe H. Towards an aspect-oriented language module: Aspects for Petri nets//Proceedings of the 7th Workshop on Domain-Specific Aspect Languages. Potsdam, Germany, 2012, 3: 21-26
- [36] Sun C, Zhang X, Shang Y, Aiello M. Integrating transactions into BPEL service compositions: An aspect-based approach. ACM Transactions on the Web, 2015, 9(2): 1-31
- [37] Yu J, Sheng Q Z, Swee J K Y, et al. Model-driven development of adaptive Web service processes with aspects and rules. Journal of Computer and System Sciences, 2015, 81(3): 533-552
- [38] García-Borgoñón L, Barcelona M A, García-García J A, et al. Software process modeling languages: A systematic literature review. Information and Software Technology, 2014, 56(2): 103-116
- [39] Li T. An Approach to Modelling Software Evolution Processes. Berlin, Germany: Springer-Verlag, 2008
- [40] Kniesel G. Detection and resolution of weaving interactions//Rashid A, Ossher H eds. Transactions on Aspect-Oriented Software Development. Berlin Heidelberg: Springer, 2009: 135-186
- [41] van der Aalst W M P. Exterminating the dynamic change bug: A concrete approach to support workflow change. Information Systems Frontiers, 2001, 3(3): 297-317
- [42] Yuan Chong-Yi. Petri Net Principles and Applications. Beijing: Publishing House of Electronics Industry, 2005(in Chinese)
(袁崇义. Petri网原理与应用. 北京: 电子工业出版社, 2005)
- [43] Wu Zhe-Hui. Introduction to Petri Net. Beijing: China Machine Press, 2006(in Chinese)
(吴哲辉. Petri网导论. 北京: 机械工业出版社, 2006)
- [44] van der Aalst W M P, van Hee K M, ter Hofstede A H M, et al. Soundness of workflow nets: Classification, decidability, and analysis. Formal Aspects of Computing, 2011, 23(3): 333-363
- [45] Dai Fei. An Approach to Verifying Software Evolution Process Models Based on EPMM [Ph. D. dissertation]. Yunnan University, Kunming, 2011(in Chinese)
(代飞. 基于 EPMM 的软件演化过程模型验证[博士学位论文]. 云南大学, 昆明, 2011)
- [46] Kiczales G, Hilsdale E, Hugunin J, et al. An overview of aspect//Proceedings of the 15th European Conference on Object-Oriented Programming. Budapest, Hungary, 2001: 327-353
- [47] Filman R E, Elrad T, Clarke S, Aksit M. Aspect-Oriented Software Development. Boston, USA: Addison-Wesley Professional, 2004: 10
- [48] Jalali A. Foundation of Aspect Oriented Business Process Management [M. S. dissertation]. Stockholm University, Stockholm, Sweden, 2011



ZHANG Xuan, born in 1978, Ph.D., associate professor. Her research interests include business process, software process and requirements engineering.

WANG Xu, born in 1976, Ph.D., associate professor. His research interests include business process and financial risk.

LI Tong, born in 1963, Ph.D., professor, Ph.D. supervisor. His research interests include software process and formal method.

CHEN Qing-Yi, born in 1981, Ph.D., assistant professor. His research interests include business process and dispatch of resources in Cloud.

LIU Jun-Hui, born in 1980, Ph.D., assistant professor. His research interests include business process and software engineering.

Background

As it becomes increasingly common for organizations to work in a process-oriented manner, single organizations may be dealing with collections of hundreds or thousands business process models. Consequently, more and more researches are done on the topic of reusable and flexible process modeling. Separation of concerns is an import topic in business process modeling that aims to reduce complexity, increase the reusability and enhance the maintainability of business process models. Some works have been done in aspect-oriented business process modeling. Most of them focused on defining the concepts and only a few of them presented modeling and correctness verification methods. To our knowledge, there has been no consideration of aspect weaving correctness. In this paper, aspect weaving correctness is defined. Weaving correctness criteria between multi-aspects and between

aspects and basic processes are designed. In addition, a detection method for errors or mistakes in process modeling is proposed to ensure error-free business process modeling.

This work is supported by the National Natural Science Foundation of China project “Research on Trustworthy Software Process Supporting Evolution” and “Research on Software Process Improvement Based on Requirement Changes”. In these two projects, the approaches of extending software process to support trustworthy software modeling and software process improvement are studied. Additionally, we have extensively studied the aspect-oriented business process, and several works has been published, such as the aspect-oriented method for business processes improvement and aspect tracing in aspect-oriented business process modeling.

