

面向组合式软件功能正确性和非功能满足性的统一建模方法

肖芳雄¹ 许波¹ 夏国恩² 李国祥² 闵华清¹

(华南理工大学软件学院 广州 510006)¹ (广西财经学院信息与统计学院 南宁 530003)²

摘要 在 Internet 动态、开放、异构、多变的环境下,组合式软件不仅要满足功能正确,非功能(时间、代价、概率等)属性的满足也日益成为一个重要的问题。在组合式软件的设计阶段,建立组合式软件功能和非功能的统一模型并验证其功能正确性和非功能满足性,是确保组合式软件可信性的重要手段。在 UML 序列图的基础上进行时间、代价、概率属性的扩展,提出了一种可对组合式软件的交互行为及其时间、代价、概率属性统一建模的属性序列图,该属性序列图以两层模式进行建模,其中基本属性序列图可对涉及时间、代价属性的具体交互场景建模,高层属性序列图通过概率合成基本属性序列图可形成全局场景。给出了针对 BPELAWS 的建模实例以说明所提建模方法的有效性。

关键词 组合式软件,功能正确性,非功能满足性,统一建模

中图分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.12.035

Unified Modeling Method of Functional and Non-functional Aspects for Composite Software

XIAO Fang-xiong¹ XU Bo¹ XIA Guo-en² LI Guo-xiang² MIN Hua-qing¹

(School of Software Engineering, South China University of Technology, Guangzhou 510006, China)¹

(School of Information and Statistics, Guangxi University of Finance and Economics, Nanning 530003, China)²

Abstract Non-functional aspects such as time, cost and probability, are becoming more and more important for composite software in the dynamic, open, heterogeneous and changeable environment of Internet. Constructing unite functional and non-functional models of composite software in design phase and verifying functional and non-functional aspects based on the models, are effective to assure dependability for the kind of software. In this paper, a novel property sequence diagram (PSD) was proposed by extending traditional UML sequence diagram with abstract time, cost and probability. PSD has two levels. The low level of PSD is extended with time and cost, and is used to model basic and detail interact scenarios of composite software. While the high level is extended with probability, and is used to construct full scenario by synthesizing the low levels. An example was illustrated to show the effectiveness of purposed methods.

Keywords Composite software, Functional aspect, Non-functional aspect, Unite modeling

1 引言

随着 Internet 的发展,出现了一类基于 Internet 的新软件形态,这类软件依据一定规范,组合 Internet 上的软件资源以构建软件应用,如网构软件^[1]、组合式软件^[2]、云服务组合^[3]等。与传统的软件形态相比,这类 Internet 环境下的组合式软件具有一些共同的特点:通过组合 Internet 上的软件资源(如主体化构件、Web 服务、云服务等,为便于叙述,以下将这些软件资源统一简称为服务)来构建软件;被组合的软件资源由第三方不同机构开发。在 Internet 动态、开放、异构、演化的环境下,组合式软件的逻辑功能不仅要满足正确,非功能属性的满足也日益成为一个重要的问题。

软件的属性包括功能属性和非功能属性。功能属性一般

指软件应符合业务逻辑和交互逻辑。非功能属性涉及的范围较广,如文献[4]提出 Web 服务的非功能属性包括响应时间、服务价格、可用性、信誉度、可靠性等。文献[5]提出 Web 服务的非功能属性包括性能、可靠性、可用性、可达性、安全性、完整性、规范性、鲁棒性等方面的需求。本文从软件抽象建模和验证的角度出发,主要考虑时间、代价、概率 3 种抽象属性,一是因为不可能在一个模型中覆盖所有的非功能属性;二是因为响应时间、性能等非功能属性可抽象为时间,服务价格、费用、带宽、能耗等非功能属性可抽象为代价,可靠性、可用性、鲁棒性等非功能属性可通过概率进行计算。

在组合式软件的设计阶段,为组合式软件建立模型并分析和验证其功能正确性和非功能满足性,是提高组合式软件可信性的重要手段。目前组合式软件的功能属性和非功能属

到稿日期:2015-02-04 返修日期:2015-03-24 本文受国家自然科学基金(61262002),中国博士后科学基金(2014M562177),广西自然科学基金(2014GXNSFAA118378),广西高等学校科学技术研究项目(2013ZD060),广西高等学校优秀中青年骨干教师培养工程,广西高等学校高水平创新团队及卓越学者计划,广西财经学院数量经济学自治区级重点实验室资助。

肖芳雄(1971-),男,博士,副教授,主要研究方向为软件工程、云计算、大数据,E-mail:fx_xiao@hotmail.com;许波(1982-),男,博士生,讲师,主要研究方向为智能机器人、计算智能、云计算;夏国恩(1977-),男,博士,教授,主要研究方向为数据挖掘和商务智能;李国祥(1984-),男,讲师,主要研究方向为数据智能处理;闵华清(1958-),男,博士,教授,主要研究方向为智能机器人、计算智能,E-mail:seanmot@foxmail.com(通信作者)。

性一般分别用不同的方法和工具进行建模和分析。比如,系统设计人员常使用 UML 对软件功能进行图形化建模和分析,研究人员常使用形式化方法(如自动机、Petri 网、进程代数等)对软件功能进行形式建模和验证,性能及可靠性分析人员常使用马尔科夫模型分析性能和可靠性。由于需要分别建立功能模型和非功能模型,并且两者建立的阶段常常不同步,通过功能验证的系统可能并不满足非功能属性要求,需要重新进行设计或调整,从而增加了开发成本,同时延长了开发周期。

实际上,软件的非功能属性依附于软件的功能行为,随着功能行为的变化而表现出相应的属性。相对于使用不同模型分别建模功能属性和非功能属性,把非功能属性的建模分析与功能行为的建模分析结合在一个统一的模型中,能使得非功能属性的建模分析更准确,同时加快建模速度,减小建模开销。作者在这方面做了一些尝试,在传统进程代数基础上提出功能与时间、代价、概率相结合进行建模分析的进程代数,称之为时间代价概率进程代数(Timed Price Probability Process Algebra, TPPPA)^[6],用以对组合式软件的功能与时间、代价、概率的形式化统一建模和验证。

组合式软件的一个显著特征是服务之间的并发和组合,虽然进程代数具有以自然的方式描述并发和组合的优势,但作为一种形式化方法,其具有严格的数学基础和形式符号,主要为研究人员所使用,不易为最终用户、需求分析人员和系统设计人员所掌握和使用,限制了形式化方法的产业化推广和使用。而 UML 作为一种可视化的通用建模语言,得到了业界大多数软件厂商的支持,易于被最终用户、需求分析人员和系统设计人员所接受。但是,UML 目前主要用于功能行为的建模分析,尚未能同时支持对时间、代价、概率的建模;此外,UML 作为一种图形化建模工具,缺乏形式化的精确语义,无法直接支持形式化分析与验证,难以验证模型的正确性。为此,本文在 UML 基础上进行时间、代价、概率的扩展,以支持功能与时间、代价、概率的统一建模;并将在以后的进一步工作中给出扩展后的 UML 到时间代价概率进程代数的转换方法,进而支持基于时间代价概率进程代数的功能与时间、代价、概率的统一验证。

本文第 2 节描述基于 UML 序列图进行时间、代价、概率扩展而得到的属性序列图;第 3 节给出基于属性序列图的 BPEL4WS 活动建模方法;第 4 节为相关工作;最后为结束语。

2 属性序列图及其建模方法

2.1 属性序列图构造

本文提出对 UML 序列图进行非功能属性扩展,原因在于组合式软件的特征之一是参与组合的服务组件之间的消息交互,而 UML 序列图是描述对象之间消息交互的建模视图。此外,UML 的动态行为视图有活动图、状态图、协作图和序列图等。如前所述,UML 活动图从业务流程的角度建模组合式软件,由于其活动建模的粒度大,容易忽略一些影响组合服务非功能属性的关键行为,因此本文不选择活动图进行扩展。UML 状态图描述系统的状态,对于组合式软件的实施者而言,服务往往由第三方提供,其状态往往不得而知,因此,不选择状态图进行扩展。UML 协作图和 UML 序列图都可以描述组件之间的消息交互,建模粒度较活动图小,更适合捕

获影响组合服务非功能属性的关键行为。其中 UML 协作图关注消息交互对象的结构组织,而 UML 序列图关注消息交互对象的交互次序,更适合捕获影响组合服务非功能属性的与交互次序相关的关键行为。因此,本文选择 UML 序列图进行扩展。此外,选择 UML 序列图进行扩展的原因还在于 UML 序列图支持场景的建模。

由于 UML 是在抽象的层面上建模,当把时间、代价、概率等属性扩展进 UML 序列图时,这些属性也应属于抽象属性,可以假定这些属性的参数值已经存在。因此,时间、代价、概率等参数如何收集、在何处收集、由谁收集等问题,不是本文关注的重点,但为了更全面地阐述本文的方法和原理,简要介绍参数的获取:(a)时间参数的获取:服务之间交互的时间参数获取一般可采取 3 种方法:一是可依据服务提供者对响应时间的历史日志数据来估计;二是服务的使用者对服务响应时间的测量;三是建模者根据对服务的经验来估计。(b)代价参数的获取:需要说明的是,本文中代价是一个抽象的概念,泛指对资源的消费。本文中资源也是一个抽象的概念,泛指软件在完成的过程中消费的各种资源,包括但不限于处理器时间、内存、外存、功耗、带宽等。本文假设参与组合式软件各服务组件提供服务时会消耗两种资源:一是服务组件之间消息交互的带宽资源,本文为着重说明非功能建模原理和方法,对模型做了简化,假设不考虑消息的大小和通道的种类等,收发一个消息所占用的带宽资源的代价都是 1;二是各服务组件的计算资源,假设计算资源的价格以该服务组件的单位时间使用费用表示,可取为该服务组件在 UDDI/WS-DL 上声明的价格。当提供持续服务时,将产生计时服务代价。(c)概率参数的获取:由于目前主流的组合式软件执行语言如 WS-BPEL、WS-CDL 等的概率信息体现在选择结构向不同分支迁移上,向不同分支迁移的概率往往取决于具体参与选择的服务,因此需要针对具体流程来分析不同分支迁移的概率,概率参数的获取与时间参数的获取类似,一是可根据服务组件的历史日志数据来估计向不同分支迁移的概率,如对于订票服务,可根据历史日志数据来统计订票成功和失败的概率;二是根据建模者对服务的经验来估计向不同分支迁移的概率。

由于组合式软件的建模和设计中,不同参与者对系统的关注点不同,比如有些参与者关注组合服务与用户的交互场景,而有些参与者关注内部服务组件间的交互场景。为支持不同参与者对系统不同部分场景的建模,本文基于 UML 序列图(Sequence Diagram, SD)提出两层建模架构的改进:一是用于对系统局部具体场景进行建模的基本属性序列图(basic Property Sequence Diagram, bPSD),bPSD 在 UML 序列图基础上为交互行为扩展时间和代价属性;二是用于把各个 bPSD 合成为系统全局场景的高层属性序列图(high-level Property Sequence Diagram, hPSD),hPSD 是根据概率把各个 bPSD 联系起来的有向图。下面结合时间、代价、概率的扩展来说明 bPSD 和 hPSD。

(1) 时间的扩展

由于 SD 是关注交互行为的视图,因此把时间与交互行为联系起来。组合式软件中服务组件之间的交互行为可以分为两种:一是服务组件之间的消息交互行为;二是服务组件交互期间在某种状态下的延时行为。下面分别讨论这两种行为的时间扩展。

(a)对于服务组件之间消息交互行为的时间扩展,本文假设服务组件之间的消息通信是同步通信,为便于描述和说明,本文假设发送消息和接收消息所需的时间都是一个时间步(Timed Step,比如1秒)。因此,为发送消息动作和接收消息动作都扩展一个时间属性,其属性值为1;由于时间是与具体的行为联系起来的,因此时间的扩展是在 bPSD 层面进行的。BPEL4WS(Business Process Execution Language for Web Services)是目前业界主流的用于构建组合软件的语言,图1以 BPEL4WS 中的 Invoke 动作为例,给出消息交互行为时间扩展的示意。

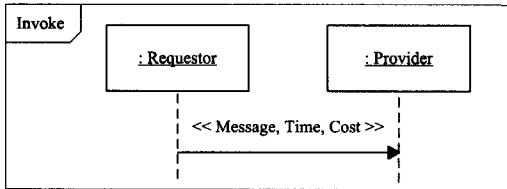


图1 bPSD中消息交互行为的时间和代价扩展

从图1可以看到,序列图中的消息线上使用了构造型<< Message, Time, Cost >>,其中 Message 表示交互消息,Time 表示消息交互所耗时间, Cost 表示消息交互的代价。如前所述,本文假定发送消息和接收消息所需的时间都是一个时间步,为消息交互行为扩展的时间属性值为1,因此,Time 的值为1。对于消息交互行为的 Cost 属性值,将在下文讨论。

(b)对于服务组件交互期间在某种状态下的延时行为,这里所说的状态指的是与代价相联系的状态,比如收费语音服务在连接建立之后、持续通话期间的状态,在这种状态下,收取的费用是按某个费率与延续时间长短成正比的。对于延时行为,扩展的时间属性值取实际的延时时间的离散值。取离散值的原因:一是符合大多数收费服务的计时情况,大多数收费服务的计时是离散值,比如收费语音服务的计时最小单位是秒,不会出现小于1秒的小数;二是简化时间处理,降低时间处理的复杂度。图2以 BPEL4WS 中的 Wait 动作为例,给出延时行为时间扩展的示意。

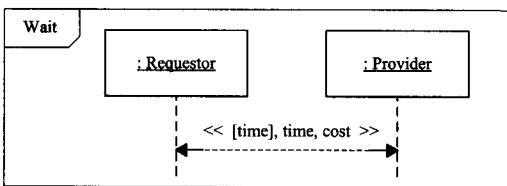


图2 bPSD中延时行为的时间和代价扩展

从图2可以看到,序列图中的消息线上使用了构造型<< [time], Time, Cost >>,其中 [time] 表示延时行为,Time 表示实际的延时时间, Cost 表示与延时相关的代价。对于延时行为的 Cost 属性值,将在下文讨论。

(2)代价的扩展

本文中代价是一个抽象的概念,泛指对资源的消费。本文中资源也是一个抽象的概念,泛指软件在完成的过程中消费的各种资源,包括但不限于处理器时间、内存、外存、功耗、带宽等。本文假设组合式软件的服务组件提供服务时会消耗两种资源:一是服务组件之间消息交互的带宽资源,本文为着重说明建模原理和方法,对模型做了简化,假设不考虑消息的大小和通道的种类等,收发一个消息所占用的带宽资源的代价都是1;二是各服务组件的计算资源,假设计算资源

的价格以该服务组件的单位时间使用费用表示,比如,对服务,可取为该服务组件在 UDDI/WSDL 上声明的价格。当提供持续服务时,将产生计时服务代价。由此,对于图1和图2中的代价属性,现可解释如下:

(a)对于服务组件之间消息交互行为的代价扩展,由于假设收发一个消息所占用的带宽资源的代价都是1,因而图中的 $Cost=1$ 。

(b)对于服务组件交互期间在某种状态下的延时行为,假设交互的两个服务组件的单位时间使用费用分别为 c, d , 延时时间为 t , 则 $Cost=t * (c+d)$ 。

(3)概率的扩展

UML 序列图支持建模结构化活动,结构化活动包括顺序、循环和选择结构等。其中,选择结构用于表示活动之间的不确定选择,用 Alt 符合来表示,如图3所示。

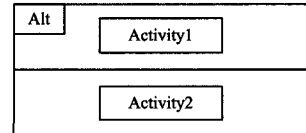


图3 基本 SD 的选择结构

从图3可以看出,UML 序列图的选择结构没有明确标注活动选择之间的概率。此外,当存在活动之间顺序、选择、循环等复杂的结构关系时,这样的结构图不能清晰地表示出活动之间的复杂结构关系。

本文提出的 hPSD 是一个扩展了概率标注的有向图,如图4所示,其初始节点表示系统的初始状态,其余的每个节点表示一个活动,两个节点之间有向边上的小数表示两个活动之间的转移概率。当某个节点有多条有向边引出时,这些有向边上的概率值之和为1。当只有一个有向边引出时,该有向边上的概率值1常常不标注。可以看出,hPSD 支持 bPSD 的合成。相比 UML 序列图的结构化活动表示,hPSD 更能表示出活动之间复杂的顺序、选择、循环等结构关系。需要说明的是,为了保持 hPSD 建模的简洁性,即为了在一个 hPSD 中表达所有 bPSD 的关系,本文没有把 UML 序列图的并发结构引入到 hPSD 中。对于并发行为的处理,可以在 bPSD 中沿用 UML 序列图的表达方式。

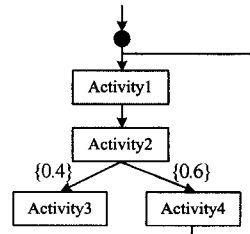


图4 hPSD 的顺序、选择和循环结构

3 基于属性序列图的建模方法

目前,基于 BPEL4WS 的组合式软件是业界主流的组合式软件构建方式,本节以基于属性序列图的 BPEL4WS 活动建模为例,给出基于属性序列图的建模方法。

BPEL4WS 是基于流程的服务组合描述语言标准。BPEL4WS 通过流程将服务组合起来,其中的每一步称作活动,BPEL4WS 流程通过活动来控制服务的交互实现组合逻辑。BPEL4WS 流程中的活动分为两类:基本类(Primitive)、结构化类(Structured)。

3.1 基本活动建模

BP EL4WS 基本活动的 PSD 模型如图 5 所示。

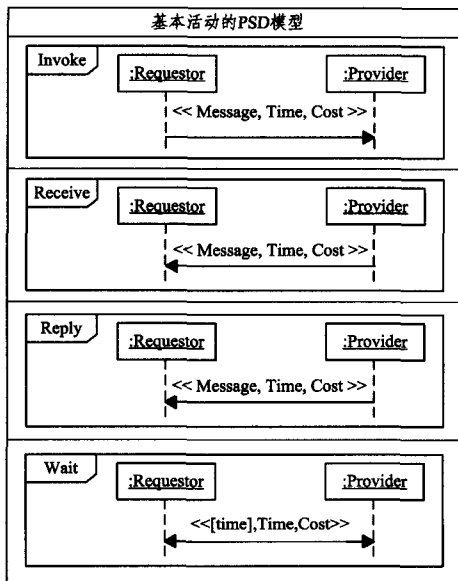


图 5 BP EL4WS 基本活动的 PSD 模型

需要说明的是, BP EL4WS 基本活动是在 bPSD 层面建模的, 其中, Invoke, Receive, Reply 的行为是消息交互行为, 使用 bPSD 中的消息交互行为的构造型来建模; Wait 的行为是延时行为, 使用 bPSD 中的延时行为的构造型来建模。另外, 由于 PSD 是关注组件之间交互行为的视图, 而 BP EL4WS 基本活动中的 Terminate, Empty, Assign, Throw 是控制流程内部逻辑的活动, 因此没有给出这些活动的 PSD 模型。

3.2 结构活动建模

BP EL4WS 结构化活动的 PSD 模型如图 6 所示。

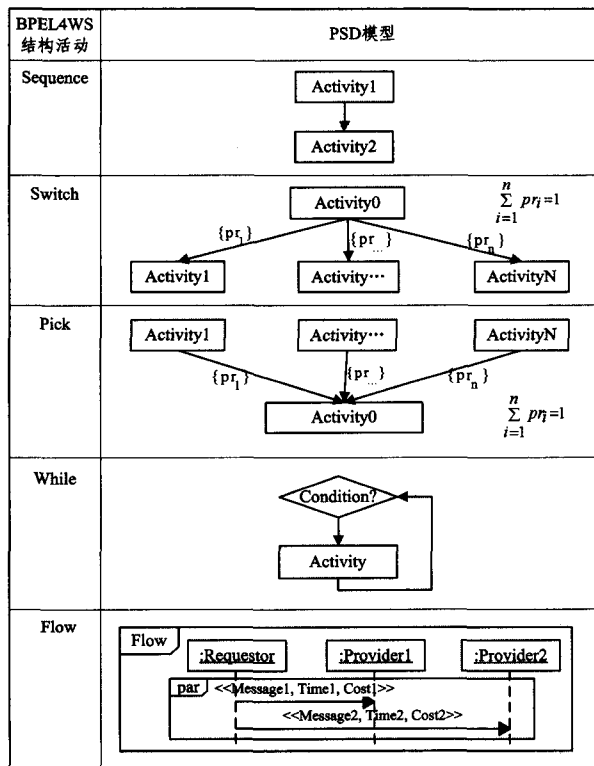


图 6 BP EL4WS 结构化活动的 PSD 模型

需要说明的是, BP EL4WS 结构化活动是在 hPSD 层面建模的。另外, 结构化活动 Flow 表示组成 Flow 的基本活动的

并发执行。如前所述, 为了保持 hPSD 建模的简洁性, 即为了在一个 hPSD 中表达所有 bPSD 的关系, 本文没有把 UML 序列图中的并发结构引入到 hPSD 中。对于并发行为的处理, 在 bPSD 中沿用了 UML 序列图的表达方式。

4 相关工作

在 UML 的非功能属性扩展建模方面, 已有一些标准的 UML 非功能属性建模框架 (Profile), 例如 SPT^[7]、QoS&FT^[8]、MARTE^[9]。其中, SPT 是 UML 1.4 版中一个用于性能、可调度性、时间等非功能属性建模的框架; QoS&FT 是 UML 中一个通用的非功能属性描述框架; MARTE 是在 UML 2.0 版中引入的一个非功能属性建模框架, 是对 SPT 的改进。在 UML 时间扩展建模方面, 文献[10]使用 UML 序列图建模时间约束的业务流程活动, 并用于 WS-BPEL 流程活动的建模, 再把建立的带时间约束的 UML 序列图转换为 WS-BPEL 流程活动, 从而得到带时间约束的 WS-BPEL 流程描述。文献[11]使用 UML Profile 建模时间约束的业务流程活动, 并用于 WS-CDL 活动的建模, 再把建立的带时间约束的 UML 模型转换为 WS-CDL 流程活动, 从而得到带时间约束的 WS-CDL 流程描述。在 UML 的概率扩展建模方面, 文献[12]通过在 UML 图中标注概率信息, 然后通过构造转移图, 将标注了概率信息的 UML 模型转换为 Markov 链进行可靠性分析; 文献[13]利用构造型、标记值、约束等 UML 扩展建模机制对构件进行 QoS 信息的标注, 从而能够对构件系统的 QoS 进行建模和分析。文献[14]提出了一个支持可靠性建模的框架, 通过在该框架中标注可靠性信息, 支持可靠性分析。文献[15]提出在 UML 用例图、顺序图和部署图中标注概率信息, 然后转化为贝叶斯模型, 进而支持对可靠性进行分析。文献[16]提出在 UML 类图和对象图中标注 QoS 信息, 然后转化为 Petri 网, 进而支持可靠性和可用性的分析和评价。文献[17]提出在 UML 消息序列图中标注概率信息, 并通过局部场景建模和全局场景合成来对系统可靠性进行分析。但上述工作暂时未能解决组合式软件功能与非功能统一建模的问题, 主要表现在: 未能同时支持时间、代价、概率的建模; 没有同时考虑多角色建模和模型合成, 例如在组合服务的建模中, 有的角色关注组合服务与用户交互的部分, 而有的角色关注服务组件之间交互的部分, 这些不同的角色分别对系统不同的局部进行建模, 所得到的局部模型需要合成为一个全局的模型, 但已有工作对这方面的考虑不多。

结束语 在 Internet 动态、开放、异构、多变的环境下, 组合式软件的非功能属性正日益成为一个不可避免的重要问题, 在组合式软件的分析 and 设计阶段, 把非功能属性的建模分析与功能行为的建模分析结合在一个统一的模型中, 能使得非功能属性的建模分析更准确, 同时加快建模速度, 减小建模开销。本文针对目前功能属性和非功能属性建模不充分的现状, 在 UML 序列图基础上提出了支持功能和时间、代价、概率的 PSD, 支持组合式软件功能与非功能的统一建模。

下一步的工作是研究和实现属性序列图到 TPPPA 的自动转换, 并研究和实现包括属性序列图建模及到 TPPPA 转换的统一建模和验证工具 (Universal Modeling and Verification, UMV), 以支持 PSD 建模、PSD 到 TPPPA 转换、TPPPA 分析验证等整个流程。

参考文献

- [1] 吕建, 马晓星, 陶先平, 等. 网构软件的研究与进展[J]. 中国科学 (E 辑), 2006, 36(10): 1037-1080
Lv Jian, Ma Xiao-xing, Tao Xian-ping, et al. Research and progress on Internetworking[J]. Science in China (Series E), 2006, 36(10): 1037-1080
- [2] Hamid R, Nezhad M, Stephenson B, et al. Virtual Business Operating Environment in the Cloud: Conceptual Architecture and Challenges [C]//OOER. 2009: 501-514
- [3] Garcia G, Octavio J. Agent-based cloud service composition [J]. Applied Intelligence, 2013, 38(3): 436-464
- [4] Zeng Liang-zhao, et al. QoS-Aware Middleware for Web Services Composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327
- [5] W3C. QoS for Web Services: Requirements and Possible Approaches [EB/OL]. <http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/>
- [6] 肖芳雄, 李燕, 黄志球, 等. 基于时间概率代价进程代数的组合式软件建模和分析[J]. 计算机学报, 2012, 35(5): 918-936
Xiao F X, Li Y, Huang Z Q, et al. Modeling and Analyzing Web Services Composition Using Timed Probabilistic Priced Process Algebra[J]. Chinese Journal of Computer, 2012, 35(5): 918-936
- [7] Object Management Group. UML profile for schedulability, performance and time [EB/OL]. <http://www.omg.org/spec/SPTP/5>
- [8] Object Management Group. UML profile for modeling QoS and fault tolerance characteristics and mechanisms [EB/OL]. <http://www.omg.org/spec/QFTP/>
- [9] Object Management Group. UML profile for modeling and analysis of real-time and embedded systems version 1.0 [EB/OL]. <http://www.omg.org/spec/MARTE/1.0/>
- [10] Cambronerio M E. Using UML Diagrams to Model Real-Time Web Services[C]//Proceedings of the Second International Conference on Internet and Web Applications and Services. 2007: 24-30
- [11] Cambronerio E M. RT-UML for modeling Real-Time Web Services[C]//Proceedings of the 3rd IEEE International Conference on Service Computing Workshops. Illinois, USA: IEEE Press, 2006: 131-139
- [12] 柳毅, 麻志毅, 何啸, 等. 一种从 UML 模型到可靠性分析模型的转换方法[J]. 软件学报, 2010, 21(2): 287-304
Liu Yi, Ma Zhi-yi, He Xiao, et al. Approach to Transforming UML Model to Reliability Analysis Model[J]. Journal of Software, 2010, 21(2): 287-304
- [13] Corellessa V, Pompei A. Towards a UML profile for QoS: A contribution in the reliability domain[J]. ACM SIGSOFT Software Engineering Notes, 2004, 29(1): 197-206
- [14] Bernardi S, Merseguer J, Petriu D C. Adding dependability analysis capabilities to the MARTE profile[C]//Czarnecki K, Ober I, Bruel J M, et al., eds. Proc. of the 11th Int'l Conf. on Model Driven Engineering Languages and Systems. Berlin: Springer-Verlag, 2008: 736-750
- [15] Cortellessa V, Singh H, Cukic B. Early reliability assessment of UML based software models[C]//Balsamo S, ed. Proc. of the 3rd Int'l Workshop on Software and Performance. New York: ACM Press, 2002: 302-309
- [16] Majzik I, Pataricza A, Bondavalli A. Stochastic dependability analysis of system architecture based on UML Models[C]//de Lemos R, Gacek C, Romanovsky A, eds. Architecting Dependable Systems. LNCS 2677, Berlin, Heidelberg: Springer-Verlag, 2003: 219-244
- [17] Rodrigues G, Rosenblum D, Uchitel S. Using scenarios to predict the reliability of concurrent component-based software systems [C]//Cerioli M, ed. Fundamental Approaches to Software Engineering, FASE 2005. Berlin: Springer-Verlag, 2005: 111-126

(上接第 151 页)

最小程度的交互情况下解决过程之间差异的问题。事实上, 许多编辑距离的概念也能被用于过程差异的检测。例如, Li 等^[10]提出了高层修改操作作为度量过程模型之间距离的基础。上述方法虽然能够计算过程之间差异的距离, 却未给出具有层次性的修改操作。于是, Küster 等^[8]提出一种基于对应关系和单输入单输出分解的方法, 并以一种用户友好的方式检测和解决差异。在此基础上, 他们又提出了一种无需修改日志, 应用复合修改操作和层次性分解解决过程模型之间差异的方法^[2]。该方法虽然能够解决过程模型之间差异的问题, 但并不适用于数据感知过程之间差异的检测和解决。

结束语 本文提出了一种检测和解决数据感知过程之间差异的方法。该方法通过比较修改前后两个数据感知过程检测出 3 类差异, 根据每一类差异都能获得对应的复合修改操作。应用定点活动和迭代计算的方式确定所有复合修改操作中的位置参数, 从而很好地解决了这些差异。下一步将在本文的基础上尝试解决数据感知过程的合并问题, 并且进一步研究数据感知过程修改操作的语义和语法上的矛盾性问题。

参考文献

- [1] Weber B, Rinderle S, Reichert M. Change Patterns and Change Support Features in Process-Aware Information Systems[C]//Proceeding of CAiSE'07. LNCS 4495: 574-588
- [2] Küster J M, Förster A, Engels G. Detecting and Resolving Process Model Differences in the Absence of a Change Log[C]//Proceeding of the 6th BPM. LNCS 5240, 2008: 244-260
- [3] Dijkman R. Diagnosing Differences between Business Process Models[C]//Proceeding of the 6th BPM. 2008, LNCS 5240, 2008: 261-277
- [4] Fan S K, Zhao J L, Dou W C, et al. A framework for transformation from conceptual to logical workflow models[J]. Decision Support Systems, 2012, 54(1): 781-794
- [5] WS-BPEL 2.0 Specification (2007) [OL]. <http://docs.oasisopen.org/wsbpel/2.0/wsbpel-v2.0.pdf>
- [6] Song W, Ma X X, Cheung S C, et al. Refactoring and publishing WS-BPEL processes to obtain more partners[C]//Proceeding of ICWS, 2011: 129-136
- [7] Pottinger R A, Bernstein P A. Merging models based on given correspondences[C]//Proceeding of the 29th VLDB, 2003: 862-873
- [8] Küster J M, Förster A, Engels G. Process Merging in Business-Driven Development; IBM Research Report, No. 3703 [R]. Switzerland: IBM Zurich Research Laboratory, 2008
- [9] Herrmann C, Krahn H, Rumpe B, et al. An Algebraic View on the Semantics of Model Composition[C]//Proceeding of the 3rd ECMDA-FA, 2007. LNCS 4530: 99-113
- [10] Li C, Reichert M, Wombacher A. On measuring process model similarity based on high-level change operations[C]//Proceeding of Conceptual Modeling. LNCS 5231, Barcelona, 2008: 248-264