

集不变条件下的服务组合演化行为研究

黄光奇¹ 蔡传青¹ 沈陈平¹ 张忠宇²

(国防科技大学信息系统工程重点实验室 长沙 410073)¹ (66105 部队司令部 保定 071051)²

摘要 从业务逻辑的一般性变更与演化需求出发,研究了服务组合中服务集合不变条件下的服务演化行为,提出了组合服务的合理演化行为集 SEBS(Sound Evolution Behavior Set),定义了一组服务工作流程结构调整的基本演化操作,证明了演化行为集 SEBS 能够保持演化后组合服务的合理性,从而能够保证在发生基于业务流程变更的服务组合演化操作后,其内部流程逻辑的合理性不变,并通过服务组合演化示例进行了实际应用说明。

关键词 业务逻辑,服务组合,演化需求,合理演化行为集,演化操作,工作流网

中图分类号 TP31 **文献标识码** A

Study on Evolution Behavior of Service Combination under Unchanged Set

HUANG Guang-qi¹ CAI Chuan-qing¹ SHEN Chen-ping¹ ZHANG Zhong-yu²

(Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China)¹

(The Department of headquarters, Army 66105, Baoding 071051, China)²

Abstract Started from the business logic of common update and evolution demand, this paper studied the evolution behavior of service combination under the condition that service set in service combination can't change. By proposing the evolution behavior set SEBS in service combination, this paper defined basic evolution operation of service work flow architecture adjustment and proved that SEBS can maintain the rationality of service combination after evolution. Thus, the rationality of inner flow logistic can't change during the operation of service combination evolution. A case study shows that it can be put into practice.

Keywords Business logic, Service combination, Evolution demand, Sound evolution behavior set, Evolution operation, Work flow net

1 概述

随着面向服务架构 SOA (Service Oriented Architecture)^[1]的快速发展和规模应用,服务技术已经成为软件领域一种得到广泛应用的关键技术。作为实现 SOA 的主要方式,Web 服务极大地推动了 SOA 在网络化软件应用领域的推广与应用。

Web 服务作为一种部署在互联网上的可编程组件,基于开放的技术规范和标准,具有完好的封装性和松散耦合的特性。其在各种异构空间信息平台的基础上,构建一个通用的与平台无关的、开发语言无关的技术层,各种不同平台上的处理资源依靠这个技术层来实施彼此的连接和集成^[2]。

在实际的网络化应用中,单个 Web 服务的功能往往比较有限,难以满足复杂多变的业务需求。当需要构造复杂的分布式应用时,须将现有的一些服务有效地整合在一起,以满足用户的需求,这种通过整合已有服务(称组件服务或伙伴服务)来向最终用户提供增值服务的技术称为服务组合^[3]。服务组合技术将各个业务单元发布的业务服务组合起来表达和执行业务流程。在开放、动态、多变的网络环境下,业务流程

经常需要变更,此时需要重新组合业务单元服务形成新的可执行流程,因此服务组合往往需要经常调整、演化以应对不断变化的内部策略和外部环境^[4,5]。

服务组合演化行为是指服务组合过程中,组合服务集合或者流程结构根据业务应用需求的变化而实施动态添加、变更与调整。一般来说,组合服务集合的变化和服务工作流程结构的调整是服务组合演化行为的两种主要形式。本文主要研究在组合服务集合不变条件下的服务工作流程结构调整问题。

文献[5]较为全面地总结了常见的服务组合演化类型,将服务的演化分为服务的浅层演化和深层演化,并提出了一个面向演化的服务生命周期方法学以应对演化带来的特殊要求。

文献[6]研究了 Web 服务组合动态演化的实例可迁移性,提出了服务组合动态演化过程的处理框架,从服务编排的角度着重研究了组合服务实例的可迁移性判断标准和相关算法。

通过对课题“面向服务的软件生产线”的研发,文献[7]从可信软件的研究领域提出了组合服务动态演化问题的解决方

本文受国防基础科研项目(B0420110607),国防预研项目(513060104)资助。

黄光奇(1972—),男,博士后,副教授,主要研究方向为 SOA、信息系统集成、CAISR 体系结构;蔡传青(1986—),男,硕士生;沈陈平(1986—),男,硕士生;张忠宇(1983—),男,硕士生。

案。其将已有的服务组合动态演化研究工作^[8,9]分为演化时间、演化操作分类以及演化带来的影响等方面。

Lohmann Niels 等人长期关注于利用 Petri 网理论对 WS-BPEL 语言及与其相关的业务过程进行的分析^[10],从而提出了大量基于 Petri 网的 WS-BPEL 分析方法和技术,包括 BPEL 到 Petri 网的转化、业务过程建模、BPEL 业务过程交互分析以及相应的工具软件等。

结合国内外相关研究成果,认为正确判断演化后组合服务的业务逻辑合理性是正确实现服务集合不变条件下的组合服务演化行为的基本问题。为了保证演化后的组合服务内部流程逻辑的正确性,结合 workflow 网合理性研究成果,本文从系统业务逻辑演化需求出发,提出了组合服务的合理演化行为集 SEBS,并定义了一组符合组合服务业务流程合理性的服务工作流程结构的基本演化操作,从而简化了演化后组合服务合理性的判别与验证,避免了传统方法在演化后的复杂验证过程。

2 服务组合问题描述

服务组合问题实质上是一个服务集合按照一定的服务工作流程组成一个组合服务的过程,可以描述为: $C = \langle S; R \rangle$,其中 $C \in Service$, $S \subset Service$, C 表示组合服务, S 表示构成组合服务 C 的原子服务集合, R 表示该集合内服务之间的组合关系。

原子服务集合 S 中原子服务的数目为 0 或 1 时,组合关系 R 将没有意义,组合服务不成立,因此,本文定义:

$$C \in CService \Leftrightarrow C = \langle \{se_0, se_1, \dots, se_n\}; R \rangle; se_i \in EService, 0 \leq i \leq n, n > 0$$

即服务 C 是组合服务当且仅当它由两个或两个以上原子服务构成。

集不变条件下的服务组合演化行为指的是,对于 $C = \langle S; R \rangle$ 而言,在原子服务集合 S 不变的条件下,集合内服务之间的组合关系 R 的调整导致组合服务 C 发生相应的变化,本文称其为组合服务 C 的 r 演化,记为 E_r 。

3 服务组合描述相关理论

3.1 工作流网

由于具备规范的形式化语义定义、图形化的表达方法以及成熟的分析理论和广泛的分析软件支持,本文采用工作流网(workflow net,简称 WF-net)进行组合服务的形式化描述及其演化的相关理论分析。工作流网实际上是一种特殊的 Petri 网,其具体的定义如下:

定义 1 工作流网(WF-net^[11])

一个 Petri 网 $WFN = (P, T, F)$ 称为工作流网,当且仅当:

- 存在一个初始库所 $i \in P$, 使得其前驱为空, 即 $\cdot i = \emptyset$;
- 存在一个终结库所 $o \in P$, 使得其后继为空, 即 $o \cdot = \emptyset$;
- 任意一个节点 $x \in P \cup T$, 都位于从 i 到 o 的一条路径上。

工作流网可用于对组合服务内部逻辑的动态描述。对于工作流网 $WFN = (P, T, F)$, 库所(place)的集合 P 用来描述服务被调用的条件。变迁(transition)的集合 T 用来描述该组合服务的构成服务 s , 实际上是其具体的一个操作。而流

关系(flow relationship)的集合 F 用来描述服务组合内各个服务间调用的逻辑关系。所有库所中令牌(token)的分布情况标识为网状态,网状态 M 用一个 $|P|$ 维向量表示,库所 p 中令牌的状态用 M_p 表示。工作流网的初始状态(终结状态)用 $M_0 (M_{end})$ 表示, $M_0 (M_{end})$ 中只有初始库所(终结库所)有标记。变迁 $t \in T$ 在状态 M 下处于使能状态,记为 $M[t >$, 当且仅当对于每一个 $(p, t) \in F$ 使得 $M_p > 0$ 。如果 t 是使能的,那么触发 t 到达的新的工作流网状态 M' , 对于任何 $(p, t) \in F$ 和 $(t, p') \in F$, 满足 $M_{p'} = M_p - 1$ 并且 $M_{p'} = M_p + 1$, 记作 $M[t > M'$ 。

定义 2 合理的工作流网(sound WF-net^[11])

工作流网 $WFN = (P, T, F, i, o, M_0)$ 是合理的, 当且仅当:

任何一个从初始状态 M_0 可达的状态 M , 都存在一个变迁序列使得 M 通往终结状态 M_{end} , 形式化描述为:

$$\forall M (M_0 \xrightarrow{*} M) \Rightarrow (M \xrightarrow{*} M_{end})$$

终结状态 M_{end} 是从初始状态 M_0 可达的唯一终止状态, 并且结束时其中至少会有一个标记, 形式化描述为:

$$\forall M (M_0 \xrightarrow{*} M \wedge M \geq M_{end}) \Rightarrow (M = M_{end})$$

这个工作流网中不存在死变迁, 形式化描述为:

$$\forall t \in T, \exists M, M's. t. M_0 \xrightarrow{*} M \xrightarrow{t} M'$$

这里将工作流网的合理性, 作为一个组合服务演化是否合理的判断依据。工作流网合理性能够保证演化后的组合服务能够顺利地终结(条件 1), 终结后其内部构成服务都已经停止运行(条件 2), 并且组合服务内任何一个构成服务都可能被调用(条件 3)。当前对应工作流网合理性的验证算法复杂度过高, 如果按照这些算法对演化后的组合服务合理性进行验证, 往往造成较大的时间开销, 而不能满足系统的实际业务要求。

定理 1^[13] 工作流网 $WFN_1 = (P_1, T_1, F_1)$ 和 $WFN_2 = (P_2, T_2, F_2)$, 其中 $T_1 \cap T_2 = \emptyset, P_1 \cap P_2 = \{i, o\}$ 并且 $t^* \in F_1$ 。那么将 WFN_1 变迁 t^* 替换为 WFN_2 构成的工作流网 $WFN_3 = (P_3, T_3, F_3)$, 从而 $P_3 = P_1 \cup P_2, T_3 = (T_1 \setminus \{t^*\}) \cup T_2$ 且 $F_3 = \{(x, y) \in F_1 \mid x \neq t^* \wedge y \neq t^*\} \cup \{(x, y) \in F_2 \mid (x, y) \cap \{i, o\} = \emptyset\} \cup \{(x, y) \in P_1 \times T_2 \mid (x, t^*) \in F_1 \wedge (i, y) \in F_2\} \cup \{(x, y) \in T_2 \times P_1 \mid (t^*, y) \in F_1 \wedge (x, o) \in F_2\}$ 。 WFN_1 和 WFN_2 是合理的, 当且仅当 WFN_3 是合理的。证明过程见文献[11]中定理 3。

3.2 组合服务业务流程描述语言

组合服务的业务逻辑指在构建组合服务时服务的编制者所要表达的业务逻辑, 具体是指组合服务内部各个原子服务之间的调用关系, 结合 WS-BPEL 业务描述语言, 如顺序(sequence)、并发(pick)、分支选择(switch)等, 在组合服务动态运行中表现为其内部调用服务的序关系。

目前作为 Web 服务组合业务流程描述语言, WS-BPEL (Web Services Business Process Execution Language)^[10] 获得了工业界广泛的采纳, 它提供了一种可以规范的业务流程和业务交互协议的方法, 并得到了主流服务组合引擎的技术支持, 具有可执行性。WS-BPEL 本质上是一种基于 XML 的服务组合流程执行语言, 其将一个组合服务的内部流程分

解为一系列的活动(activity),表1列举了BPEL的主要活动。

表1 BPEL典型活动说明

活动的标签	活动的用途
invoke	在现有的Web服务上调用一项服务操作
receive	接收来自外部实体的消息
assign	将一个值从源复制到目标
terminate	无条件中止当前流程
throw	抛出一个异常

同时WS-BPEL定义了一组关键字标签,用于对活动进行组合,从而达到描述业务流程的目的。这些关键字描述了WS-BPEL语言的结构特征,表2列举了BPEL的主要关键字。

表2 BPEL典型关键字说明

关键字标签	关键字的用途
sequence	定义一组操作的顺序序列
switch	条件分支选择语句
flow	封装一组应该并行执行的操作步骤
pick	选择语句

4 服务组合合理演化行为集

本文基于服务组合描述的相关理论,提出组合服务的合理演化行为集SEBS,从信息系统业务逻辑演化需求出发,定义了一组符合组合服务业务流程合理性的演化行为。旨在规范组合服务演化行为,进而简化对于演化后组合服务合理性的验证。这里将对应合理的工作流网的组合服务称为合理的组合服务。

对于组合服务 $C = \langle S; R \rangle$, 集不变条件下的服务组合演化行为可以定义如下:

定义3 r 演化 E_r :是指组合服务的构成域 $S = dom(C)$ 没有发生变化,而构成关系 R 发生了变化。不考虑循环调用,结合WE-BPEL语言将组合服务的基本构成关系分为如下3种:其中 $\forall a, b \in S$ 分别由变迁 t_a 和变迁 t_b 表示。

• 顺序关系(sequence):在组合服务内部,构成服务被顺序地调用,即 $\exists p_s \in P$ s. t. $p_s \in t_a \wedge p_s \in t_b$,如图1所示。

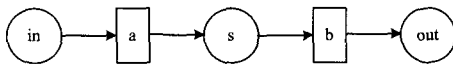


图1 顺序调用关系

• 选择关系(switch):在组合服务内部,构成服务依据条件被选择地调用,即 $\exists p_{in}, p_{out} \in P$ s. t. $p_{in} \in t_a \wedge p_{in} \in t_b \wedge p_{out} \in t_a \wedge p_{out} \in t_b$,如图2所示。

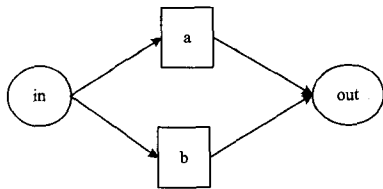


图2 选择调用关系

• 并发关系^[12](flow):在组合服务内部,构成服务同时被调用,即 $\exists p_{in}, p_{in'}, p_{out'}, p_{out} \in P \wedge t_{split}, t_{join} \in T$ s. t. $\{p_{in}, p_{in'}\} \subset t_{split} \wedge p_{in} \in t_a \wedge p_{in'} \in t_b \wedge p_{out'} \in t_a \wedge p_{out} \in t_b \wedge \{p_{out}, p_{out'}\} \subset t_{join}$,如图3所示。

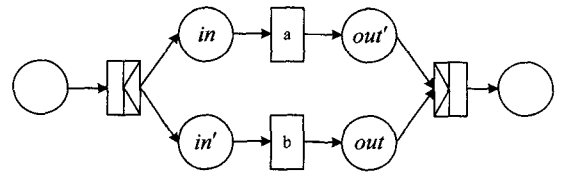


图3 并发调用关系

根据组合服务演化的需求,3种构成关系可能发生如下7种演化行为,如图4所示。

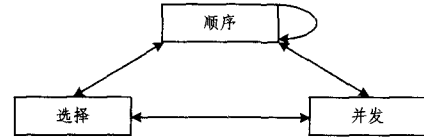


图4 7种 r 演化行为

对上面的7种 r 演化行为进行描述,设演化前后的组合服务对应工作流网分别为 $WFN^O = (P^O, T^O, F^O, i^O, o^O, M^O)$ 和 $WFN^N = (P^N, T^N, F^N, i^N, o^N, M^N)$,其中 WFN^O 是合理的工作流网, $A \setminus B$ 表示将集合 A 除掉集合 B 中的元素,则 $WFN^O \rightarrow WFN^N$ 可能在下列7种演化行为中发生:

1) 逆序演化 b_{sq2sq} :顺序执行服务 a 和服务 b 变为逆序执行,描述为:

$$T^N = T^O \wedge P^N = P^O \wedge F^N = F^O \cup \{(t_b, p_s), (p_s, t_a)\} \setminus \{(t_a, p_s), (p_s, t_b)\}$$

2) 顺序选择演化 b_{sq2sw} :顺序执行服务 a 和服务 b 变为选择执行,描述为:

$$T^N = T^O \wedge P^N = P^O \setminus \{p_s\} \wedge F^N = F^O \cup \{(p_{in}, t_b), (t_a, p_{out})\} \setminus \{(t_a, p_s), (p_s, t_b)\}$$

3) 顺序并发演化 b_{sq2fl} :顺序执行服务 a 和服务 b 变为并发执行,描述为:

$$T^N = T^O \wedge P^N = P^O \cup \{p_{in'}, p_{out'}\} \setminus \{p_s\} \wedge F^N = F^O \cup \{(t_{split}, p_{in'}), (t_a, p_{out'}), (p_{in'}, t_b), (p_{out'}, t_{join})\} \setminus \{(t_a, p_s), (p_s, t_b)\}$$

4) 选择并发演化 b_{sw2fl} :选择执行服务 a 和服务 b 变为并发执行,这里认为演化后网中的并发开始和并发结束变迁满足: $\{t_{split}, t_{join}\} \subset T^O$,从而描述为:

$$T^N = T^O \wedge P^N = P^O \cup \{p_{in'}, p_{out'}\} \wedge F^N = F^O \cup \{(t_{split}, p_{in'}), (p_{in'}, t_b), (t_a, p_{out'}), (p_{out'}, t_{join})\} \setminus \{(p_{in}, t_b), (t_a, p_{out})\}$$

上述后3种演化行为的逆行为:选择顺序演化 b_{sw2sq} 、并发顺序演化 b_{fl2sq} 和并发选择演化 b_{fl2sw} 与之相似,在此就不赘述。

命题1 r 演化行为集 $E_r = \{b_{sq2sq}, b_{sq2sw}, b_{sw2sq}, b_{sq2fl}, b_{fl2sq}, b_{sw2fl}, b_{fl2sw}\}$ 能够保持演化后组合服务的合理性。

本文以并发演化行为 b_{sw2fl} 为例,证明如下:

证明:根据选择关系构建工作流网 WFN_{sw}^O ,其中 $P_{sw}^O = \{i, o, p_{in}, p_{out}\}$, $T_{sw}^O = \{t_{split}, t_{join}, t_a, t_b\}$, $F_{sw}^O = \{(i, t_{split}), (t_{split}, p_{in}), \dots, (t_{join}, o)\}$,如图5(a)所示。同理构建工作流网 WFN_{fl}^N ,如图5(b)所示。

根据定理1中的组成方法,可得演化前工作流网 $WFN^O = WFN_{sw}^O + WFN_{other}$,其中 WFN_{other} 为演化前工作流网的剩余网,相当于定理中的 WFN_1 。由 WFN^O 是合理的,可推之 WFN_{other} 是合理的。显然网 WFN_{fl}^N 是合理的,从而可知演化后的网 $WFN^N = WFN_{fl}^N + WFN_{other}$ 也是合理的,证毕。

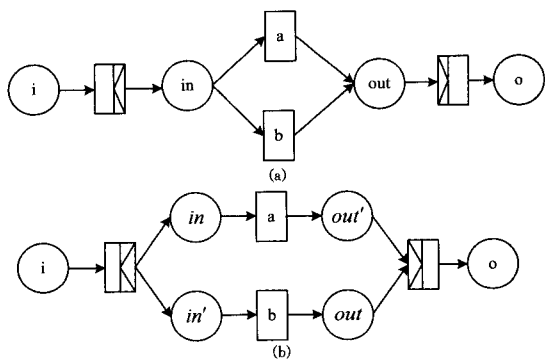


图5 r 演化行为合理性证明

5 服务组合演化示例

假设在一个防空指挥信息中心,当前方传来预警雷达探测得到的目标信息后,即开始这样的作业流程:首先,判断目标的性质,这里假设为飞机或导弹;然后,对于判断的结果进行识别,如飞机型号或导弹型号;然后,获取目标所在空域我方辖区的防空部署;最后,根据目标识别结果和我方实时部署情况,形成实时的空域态势信息。为了提高指控的效率和自动化程度,在已有服务的基础上,将上述业务流程实现为一个组合服务,其逻辑流程为:首先输入雷达探测数据,调用目标类别识别服务 Service0;然后,根据执行结果判断调用战机识别服务 Service1 或导弹识别服务 Service2;然后,调用辖区防空部署查询服务 Service3;最后,调用敌我态势综合服务 Service4,返回态势信息。

Active BPEL 是一款可执行 BPEL4WS 规范的开源流程引擎,该引擎接受 BPEL4WS 流程的定义,创建流程实例,并执行它们,其在结构上有 3 个主要的方面:引擎、流程和活动。引擎执行相匹配的 1 个或多个 BPEL 流程,流程由活动组成,并按照活动的顺序或包含 link 执行。

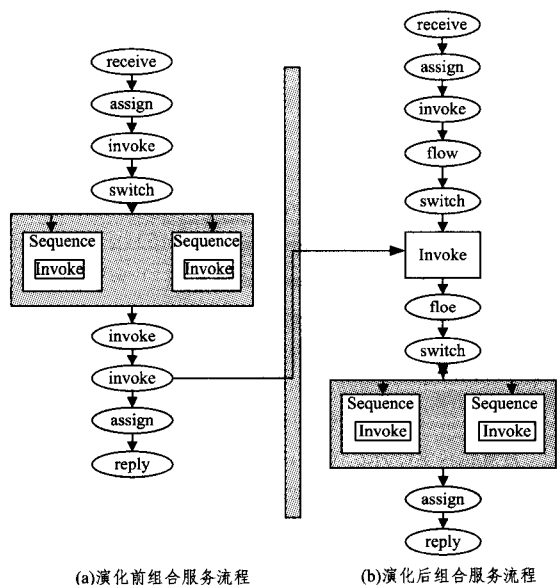


图6 组合服务的流程演化

基于 Active-BPEL 开源组合服务引擎,本文实现了该组合服务,从而形成了如图 6(a) 所示的组合服务内部流程。但在随后的使用中,业务人员发现如果在进行飞机或导弹目标识别的同时查询辖区防空部署信息,可以提高敌我态势信息

的生成速度。从而演化了原有的业务逻辑,形成了如图 6(b) 所示的组合服务。

图中,箭头表明了演化前后组合服务内部逻辑的演化。演化前:Service3 与选择执行的 Service1 和 Service2 是顺序调用关系;演化后:Service3 与选择执行的 Service1 和 Service2 变为并发调用关系。

下面分析上述组合服务业务流程演化行为的合理性,将它们转化为工作流网,如图 7 所示。

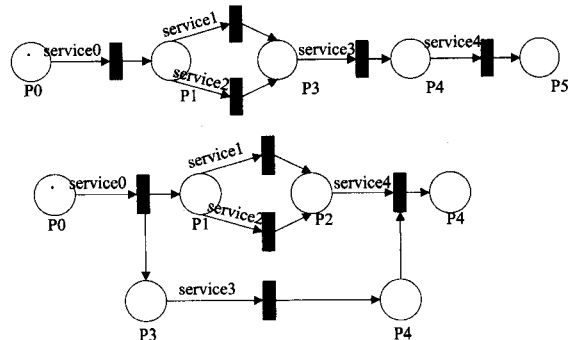


图7 演化前后的工作流网

由此可见,上述的组合服务演化实质上是 r 演化行为集中的顺序并发演化,是合理的演化行为。从而生成演化前后工作流网的状态可达图,如图 8 所示。

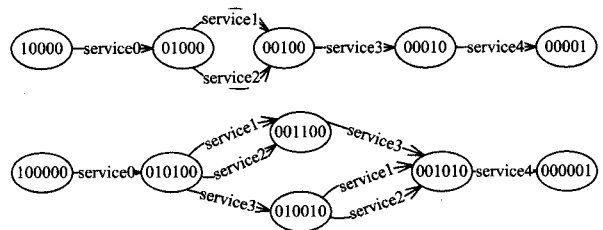


图8 演化前后的状态可达图

由服务组合演化前后工作流网的状态可达图比较可知,该演化行为是合理的,能够保持工作流程结果的一致。

结束语 在构造复杂的网络化应用中,往往需要通过服务组合的方式将功能有限的一些单个服务有效地整合在一起,支持将各个业务单元发布的业务服务组合起来表达和执行业务流程,以满足用户的业务需求。本文重点研究在组合服务集合不变条件下的服务工作流程结构调整问题。为了保证基于业务流程变更的服务组合演化操作后其内部流程逻辑的合理性不变,借鉴工作流网的相关研究成果,本文从业务逻辑的一般性变更与演化需求出发,提出了组合服务的合理演化行为集 SEBS,定义了包括逆序演化、顺序选择演化、顺序并发演化、选择并发演化、选择顺序演化、并发顺序演化和并发选择演化等 7 个符合组合服务业务流程合理性的服务工作流程结构的基本演化操作,并证明了该演化行为集能够保持演化后组合服务的合理性,最后通过服务组合演化示例进行了实际应用说明。本文的研究结果能够简化演化后组合服务合理性的判别与验证,避免传统方法在演化后的复杂验证过程。在下一步的工作中,将从组合服务集合发生变化时的演化行为及组合服务实例的动态迁移问题等方面继续深入研究。

参考文献

[1] Newcomer E, Lomow G. Understanding SOA with Web Services

- [M]. Addison Wesley Professional, 2004
- [2] 柴晓路, 梁宇奇. Web Services 技术、架构和应用[M]. 北京: 电子工业出版社, 2002
- [3] 喻坚, 韩燕波. 面向服务的计算—原理与应用[M]. 北京: 清华大学出版社, 2006
- [4] Ryu S H, Casati F, Skogsrud H, et al. Supporting the dynamic evolution of Web service protocols in service-oriented architectures[J]. *ACM Transactions on the Web*, 2008, 2(2): 1-45
- [5] Papazoglou M P. The challenges of service evolution[C]// *Proceedings of the International Conference of Advanced Information Systems Engineering*. Montpellier, France, 2008: 1-15
- [6] 宋巍, 马晓星, 吕建. Web 服务组合动态演化的实例可迁移性[J]. *计算机学报*, 2009, 32(9): 1816-1831
- [7] 曾晋, 孙海龙, 刘旭东, 等. 基于服务组合的可信软件动态演化机制[J]. *软件学报*, 2010, 21(2): 261-276
- [8] Von de Aalst W M P, Jablonski S. Dealing with workflow change: Identification of issues and solutions[J]. *Int'l Journal of Computer System Science & Engineering*, 2000, 15(5): 267-276
- [9] Andrikopoulos V, Benbernou S, Papazoglou M P. Managing the evolution of service specifications[C]// *Proc of the 20th Int'l Conf. on Advanced Information Systems Engineering*. 2008: 359-374
- [10] Lohmann N. A Feature-Complete Petri Net Semantics for WS-BPEL 2.0[J]. *Lecture Notes in Computer Science*, 2008, 4937: 77-91
- [11] Van der Aalst W M P. *Workflow Management Models, Methods, and Systems*[M]. The MIT Press Cambridge, Massachusetts London, England, 2002
- [12] 李景霞, 程久军. Web 服务组合的层次颜色 Petri 网描述模型[J]. *计算机工程*, 2010, 24(12): 39-44
-
- (上接第 349 页)
- [6] Sushit J, Samarati P, Subrahmanian V. A Logical Language for Expressing Authorizations[C]// *Proceedings of IEEE Symposium on Security and Privacy*. Oakland, Calif, USA, 1997: 94-107
- [7] Lobo J, Bhatia R, Naqvi S. A Policy Description Language[C]// *Proceedings of the Sixteenth National Conference on Artificial Intelligence Eleventh Innovative Applications of AI Conference*. Orlando, Florida, USA, 1999
- [8] Damianou N. Tools for domain-based policy management of distributed systems[C]// *Network Operations and Management Symposium*. NOMS 2002. IEEE/IFIP, 2002
- [9] Damianou N, Dulay N, Sloman M, et al. Ponder: A Language for Specifying Security and Management Policies for Distributed Systems[S]. *The language Specification Version 2.3*. Imperial College of Science Technology and Medicine, Department of Computing, London, 2001
- [10] Damianou N, Dulay N, Sloman M, et al. The Ponder Policy Specification Language[C]// *Proceedings of Policies for Distributed Systems and Networks*. HP Labs, Bristol, UK, Springer-Verlag, 2001: 18-38
- [11] Damianou N, Dulay N, Sloman M, et al. A policy deployment model for the Ponder language[C]// *Proceedings of IEEE/IFIP International Symposium on Integrated Network Management (IM'2001)*. Seattle, IEEE Press, 2001: 529-543
- [12] Hicks B, Rueda S, Clair L S, et al. A logical specification and analysis for SELinux MLS policy[C]// *Proceedings of the ACM Symposium on Access Control Models and Technologies*. 2007: 91-100
- [13] Ahn G J, Xu Wen-juan, Zhang Xin-wen. Systematic Policy Analysis for High-Assurance Services in SELinux[C]// *the 9th IEEE Workshop on Policies for Distributed Systems and Networks (POLICY)*. NY, USA, 2008
- [14] Binder D J. A logic-based security language[C]// *IEEE Symposium on Security and Privacy*. 2002: 105-113
- [15] 李星. 访问控制策略语言的研究与设计[D]. 合肥: 中国科学技术大学, 2009
- [16] OASIS. eXtensible Access Control Markup Language(XACML) Version 2.0 core specification[EB/OL]. www.oasis-open.org/committees/xacml/, 2005
- [17] 王雅哲, 冯登国, 张立武, 等. 基于多层次优化技术的 XACML 策略评估引擎[J]. *软件学报*, 2011, 22(2): 323-338
- [18] 钟勇, 秦小麟, 郑吉平, 等. 一种灵活的使用控制授权语言框架研究[J]. *计算机学报*, 2006, 29(8): 1408-1418
- [19] Becker M Y, Fournet C, Gordon A D. SecPAL: Design and semantics of a decentralized authorization language[J]. *Journal of Computer Security*, 2010, 18(4): 619-665
- [20] Gurevich Y, Neeman I. DKAL: Distributed-Knowledge Authorization Language[C]// *21st IEEE Computer Security Foundations Symposium*. 2008: 149-162
- [21] Abadi M. Logic in access control [C]// *Proceeding of the Eighteenth Annual IEEE Symposium on Logic in Computer Science*. Ottawa, Canada, 2003: 228-233
- [22] 颜学雄, 王清贤. 基于逻辑的访问控制研究[J]. *计算机科学*, 2009, 36(4): 42-46
- [23] Crampton J, Loizou G, O'Shea G. A logic of access control[J]. *The Computer Journal*, 2001, 44(2): 137-149
- [24] Dougherty D J, Fisler K, Krishnamurthi S. Specifying and reasoning about dynamic access-control policies[C]// *International Joint Conference on Automated Reasoning(IJCAR'06)*. 2006
- [25] Chaudhuri A, Naldurg P, Rajamani S K, et al. EON: Modeling and analyzing dynamic access control systems with logic programs[C]// *Proceedings of the 15th ACM Conference on Computer and Communications Security*. 2008: 381-390
- [26] 马丽, 马世龙, 陆跃飞, 等. 一种 RBAC 的描述逻辑表示方法[J]. *计算机科学*, 2010, 37(3): 29-35
- [27] 徐震, 李斓, 冯登国. 基于角色的受限委托模型[J]. *软件学报*, 2005, 16(5): 970-978
- [28] Lunt T F. *Access control policies for database systems*[M]. *Database Security II: Status and Prospects*. North-Holland, Amsterdam, 1989: 41-52
- [29] Geri S, Gottlob G, Tanca L. *Logic Programming and Databases* [M]. Beijing: Springer-Verlag, 1992