

Web 服务流程的结构范式及其判定算法

胡 强¹⁾ 杜军威¹⁾ 杜玉越²⁾

¹⁾(青岛科技大学信息科学技术学院 山东 青岛 266061)

²⁾(山东科技大学信息科学与工程学院 山东 青岛 266590)

摘 要 服务组合是一种利用已有 Web 服务构建新的增值服务的有效方法,然而当前缺乏对组合产生的服务流程进行结构健壮性层次划分和判定的准则.针对上述问题,提出 Web 服务流程结构范式,从路径的可达性和结构冗余性两个角度将结构范式划分为四个层次.基于逻辑 Petri 网建模服务流程为服务网,将服务流程结构范式的判定转化为对服务网组成结构的分析.在服务网的基础上构建控制流网,从而将服务流程中的控制流和数据流进行分离.定义了控制流网语言,并提出一种通过约简求解控制流网语言的方法.通过对控制流网语言中字符序列组成结构特征的分析,获取服务网的结构性质,完成对服务流程结构范式层次的判定.给出了服务流程结构范式的定义、判定定理以及算法,并通过实例验证了文中方法对流程结构范式进行判定的可行性和有效性.

关键词 Web 服务;结构范式;逻辑 Petri 网;服务网;健壮性

中图法分类号 TP311 DOI号 10.3724/SP.J.1016.2015.00178

The Structure Normal Form of Web Service Processes and Its Testing Algorithm

HU Qiang¹⁾ DU Jun-Wei¹⁾ DU Yu-Yue²⁾

¹⁾(School of Information Science and Technology, Qingdao University of Science and Technology, Qingdao, Shandong 266061)

²⁾(College of Information Science and Engineering, Shandong University of Science and Technology, Qingdao, Shandong 266590)

Abstract Service composition is an efficient way to construct a new and value-added service from the existing Web services. However, there are still no rules to evaluate the level of structure soundness in the service process which is generated by service composition. To resolve such problem, the structure normal form of service processes was proposed in this paper. From the accessibility of the paths and structural redundancy, the level of structure soundness of a service process was divided into four levels in the structure normal form. The service process was modeled as a service net based on logic Petri nets, and the evaluation of the structure normal form was converted to analyze the structure features of the service net. Control flow net was built based on the service net and it separates the control flow and data flow in a service process. The language of control flow net was defined and the method to obtain it based on reduction and recursion was also provided. By analyzing the structure properties of the component character sequences in the language of control flow net, the structure properties of the service net are easily obtained, and then the level of the structure normal form of the service process can be evaluated. The corresponding definitions, judgment theorems and algorithm of the structure normal form were presented. Some application examples are given to show the feasibility and efficiency of the algorithm in testing the structure normal form of the service processes.

Keywords Web service; structure normal form; logic Petri net; service net; soundness

收稿日期:2013-05-06;最终修改稿收到日期:2014-07-08. 本课题得到国家“九七三”重点基础研究发展规划项目基金(2010CB328101)、国家自然科学基金(61170078,61273180)、教育部高等学校博士学科点专项科研基金(20113718110004)、山东省自然科学基金项目(ZR2012FL17)、青岛市科技计划基础研究项目(13-1-4-116-jch)、山东科技大学科研创新团队支持计划项目(2011KYTD102)资助. 胡 强,男,1980 年生,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为 Petri 网的建模与分析技术、服务计算. E-mail: huqiang200280@163.com. 杜军威,男,1974 年生,博士,副教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为软件的形式化建模与验证. 杜玉越(通信作者),男,1960 年生,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为计算机支持协同工作、形式化技术、Petri 网、服务计算、实时系统、Web 服务技术、工作流. E-mail: yydu001@163.com.

1 引言

随着云计算、物联网以及移动互联网等新一代信息技术的成熟与应用的普及,软件开发和交付模式呈现出明显的服务化特征.作为一种成熟的服务计算实现模式,Web 服务是 SOA 模式软件开发的首选.Web 服务在网络化软件的构建中存在两种使用方式:原子 Web 服务调用和组合 Web 服务流程.

通过组合产生的服务流程可以应对一些复杂的服务请求,对于给定的服务请求和一组原子服务,采用不同组合方法可以得到不同组合模式的服务流程,这些服务流程在结构的分支连通性、冗余性以及逻辑正确性等方面存在差异,本文将上述反映流程结构差异的特征称为服务流程的结构健壮性.

现有研究工作中通常以工作流网的健壮性^[1]为基础,提出一种新的约束规则,从流程中路径的可达性方面对结构健壮性进行评价.然而,在一些特定的应用场景下,并不要求服务流程中所有的服务路径均可正确执行,只要某些特定服务路径可达即可.因此,仅以服务流程中路径的可达性来定性地判定一个服务流程结构的健壮性是不合理的.服务流程的结构健壮性不仅要考察流程中路径的可达性,也需要考虑流程结构中是否存在冗余等特征.因此,本文提出对流程结构健壮性进行层次划分的思想,从路径可达性和结构冗余性两个角度将流程的结构健壮性划分为不同的层次级别.

通过建立流程的结构健壮性层次划分准则及判定方法,可以有效地对服务流程的结构质量和可用性进行评价,进而有助于改进服务组合的合理性.借鉴关系数据库中的范式概念,将 Web 服务流程的结构健壮性层次划分准则定义为流程的结构范式.

为了对服务流程的结构健壮性进行层次划分和判定,需对流程进行形式化建模,目前常用于 Web 服务建模的形式化工具主要有 Petri 网^[2-4]、随机 Petri 网^[5-6]、自动机^[7-8]、描述逻辑^[9]、进程代数^[10]以及 LTS(Labeled Transition System)^[11]等. Petri 网作为一种图形化的建模方式,直观简洁且具有良好的数学推理分析方法,但在描述和分析复杂服务流程时存在状态空间爆炸问题.进程代数则过于符号化,直观性稍差;描述逻辑在服务描述及组合中的推理过程较为繁琐.自动机以及 LTS 与 Petri 网类似,但数学推理和分析能力稍显不足.

逻辑 Petri 网(Logic Petri Net, LPN)是在普通 Petri 网的某些变迁中添加逻辑表达式而得到的一

种高级 Petri 网,其描述能力等价于带抑制弧的 Petri 网^[12].通过合理设置逻辑表达式,可有效地缩减建模时产生的网规模,且具有良好的建模结构性,适合对 Web 服务流程建模^[13].本文采用逻辑 Petri 网作为流程描述工具,将 Web 服务流程建模为服务网,基于服务网中的可触发变迁序列结构性质,分析 Web 服务流程的结构特征,进而划分相应结构范式,并给出判定算法.

2 相关研究

现有的成果中以服务流程的结构健壮性为主题的研究相对较少,研究者通常对流程结构的某些局部性质进行挖掘和验证.例如 Benatallah 等人^[14]通过构造完全交互树来判定业务流程之间的互操作是否匹配. Xiong 等人^[15]利用 Petri 网对服务流程组成服务之间的交互行为做了分析,提出了若干种交互失配模式及其判定算法. Tan 等人^[16]利用随机 Petri 网定义了抽象工作流网,对 Web 服务流程建模,提出了一组流程结构化简规则,并对流程性能进行分析和评估. Mansour 等人^[17]对组合服务流程的事务性和可靠性进行了研究,并提出了相应的容错解决方案. Mateescu 等人^[18]从软件实体复用和组合的角度,利用状态图生成软件实体之间的会话适配器,并将其应用到 Web 服务交互协调过程中.

邓水光等人^[19]将 Web 服务之间的交互行为描述为 π 演算进程,利用 π 演算的操作语义对服务行为的兼容性进行推演和验证.李喜彤等人^[20]采用颜色 Petri 网建模 Web 服务流程,并利用通信可达树对组合后的服务流程的逻辑正确性进行验证.宋巍等人^[21]综合考虑服务流程中的控制依赖、数据依赖、交互依赖以及异步调用依赖,基于程序依赖图分析了 BPEL 服务流程的可替换性.雷丽晖等人^[22]采用扩展自动机验证服务流程执行过程中可能存在死锁或活锁等逻辑错误.

在流程结构健壮性方面, Clempner^[23]提出了一种基于轨迹跟踪的方法去检测工作流网或业务流程中健壮性的规则. Du 等人^[24]利用 Petri 网对流程的组成服务之间的交互行为做了分析,并给出一种新的流程合理性判定标准. Martens^[25]提出了一种基于工作流网定义的弱健壮性概念,并在弱健壮性概念的基础上给出流程可用性的判定准则,进而判断服务的交互兼容性和可替换性. Groefsema 等人^[26]在对业务过程模型的控制流进行分析时,将流程的健壮性定义为流程中不存在死锁和同步缺失. van

der Aalst 等人^[27]在对业务流程的健壮性进行评价时,提出弱良好、 k 良好以及最大 k 良好 3 个级别,并结合 workflow 健壮性的定义给出了不同级别流程健壮性的判定方法,该工作虽然对健壮性进行了层次划分,但所涉及到的 3 个层次的健壮级别仅仅是 workflow 健壮性 3 个条件的不同表述。

上述研究大多是选取了流程结构中的某一性质进行分析或验证,特别是在健壮性方面鲜有分层评价的工作。为了合理评价流程的健壮性,本文从流程中的路径可达性和结构冗余性两个角度分析 Web 服务流程的结构健壮性,借鉴已有工作中在流程结构性验证和健壮性分析方面取得的成果,建立起 Web 服务流程的结构健壮性层次划分准则及其形式化判定方法。

3 Web 服务流程建模

3.1 逻辑 Petri 网简介

采用逻辑 Petri 网描述的系统模型结构良好且规模较小,文献^[28]中给出了一个应用实例,相比传统 Petri 网建模,网模型中变迁、库所以及可覆盖性树中的结点分别缩减了 57.1%、19.2% 和 50.0%。在前期研究中,已将逻辑 Petri 网应用于实时协作系统^[29-30]、E-Commerce 工作流^[31]、网上股票交易系统^[32]以及 Web 服务流程的建模^[13]与分析。下面给出 Petri 网、逻辑表达式以及逻辑 Petri 网等相关概念的定义。

定义 1. Petri 网是一个四元组集合 $\Sigma = (P, T; F, M)$, 其中 P 是库所集合, T 是变迁集合, 且 $P \cap T = \emptyset$; $F \subseteq (P \times T) \cup (T \times P)$ 称为 Petri 网的流关系, 映射 $M: P \rightarrow \{0, 1, 2, \dots\}$ 是 Petri 网的标识。

若 $\forall p \in t: M(p) \geq 1$, 则称 t 在 M 是下使能的, 记为 $M[t]$; 若 $M[t]M'$, 则对于 $\forall p \in P, M'(p)$ 可通过以下公式计算:

$$M'(p) = \begin{cases} M(p) - 1, & p \in t - t' \\ M(p) + 1, & p \in t' - t \\ M(p), & \text{其他} \end{cases}$$

定义 2. $\Sigma = (P, T; F, M)$ 为一 Petri 网, $x \in P \cup T$ 是 Σ 中的一个结点, $\cdot x = \{y \mid (y, x) \in F\}$ 称为 x 的前集; $x \cdot = \{y \mid (x, y) \in F\}$ 称为 x 的后集。特别定义操作 π 获取网中结点的前集和后集, 令 $\pi_1(x) = \cdot x$, $\pi_2(x) = x \cdot$ 。

定义 3. $P = \{p_1, p_2, \dots, p_n\}$ 是一个布尔变量集合, $f = \bar{p}_{i_1} \Delta_{i_1} \bar{p}_{i_2} \cdots \Delta_{i_{j-1}} \bar{p}_{i_j} \Delta_{i_j} \cdots \Delta_{i_{m-1}} \bar{p}_{i_m}$ 是集合 P

上的逻辑表达式, 其中 $\Delta_{i_j} \in \{\vee, \wedge\}$, \vee 与 \wedge 分别为逻辑析取与合取运算符, \bar{p}_{i_j} 称为 f 的组成元项, $\bar{p}_{i_j} = p_{i_j}$ 或 $\neg p_{i_j}$, $p_{i_j} \in P$, \neg 为逻辑非运算符, $1 \leq i_j \leq n$, 记为 $f = \delta(\Delta, \bar{P})$, Δ 和 \bar{P} 分别为 $\Delta_{i_1} \cdots \Delta_{i_{j-1}} \Delta_{i_j} \cdots \Delta_{i_{m-1}}$ 和 $\bar{p}_{i_1} \bar{p}_{i_2} \cdots \bar{p}_{i_j} \cdots \bar{p}_{i_m}$ 的顺序序列, 并将 f 的所有组成元项的集合标记为 Ω_f 。

定义 4. $P = \{p_1, p_2, \dots, p_n\}$ 和 $Q = \{q_1, q_2, \dots, q_n\}$ 为两个布尔变量集合, λ 为一个映射函数: $\lambda(P) \rightarrow Q$. $f = \delta(\Delta, \bar{P})$ 为 P 上的逻辑表达式, 令 $P' \subseteq \Omega_f$, 对 $\forall p \in P'$, 利用 $\lambda(p)$ 替换 p , 得到逻辑表达式 f' , f' 称为 f 的函变替换, 记为 $f' = \Psi(f, P', \lambda)$ 。

例如, 假设 $P = \{p_1, p_2, p_3, p_4\}$ 与 $Q = \{q_1, q_2, q_3, q_4\}$ 为布尔变量集合, 且存在映射 λ 满足 $\forall p_i \in P: \lambda(p_i) = q_i (1 \leq j \leq n)$. 令 $f = p_1 \wedge (\neg p_2 \vee p_3)$, $P' = \{p_2, p_3\}$, 则 $f' = \Psi(f, P', \lambda) = p_1 \wedge (\neg q_2 \vee q_3)$ 。

定义 5. 设 $LN = (P, T, F, I, O)$, $LPN = (LN, M)$ 称为一个逻辑 Petri 网, 其中:

(1) P 是一个有限库所集合;

(2) $T = T_D \cup T_I \cup T_O$ 是一个有限变迁集合, 且 $T \cup P \neq \emptyset$, $\forall t \in T_I \cup T_O: t \cap t' = \emptyset$, 其中

① T_D 为一个传统 Petri 网变迁集合;

② T_I 为一个逻辑输入变迁集合; $\forall t \in T_I, t$ 的触发受其输入库所集上的逻辑表达式 $f_I(t)$ 的约束, 即 $f_I(t)$ 中的谓词变量包含 t 的所有输入库所, 称 $f_I(t)$ 为逻辑输入表达式;

③ T_O 为一个逻辑输出变迁的集合; $\forall t \in T_O, t$ 触发后的结果受其输出库所集上的逻辑表达式 $f_O(t)$ 的约束, 即 $f_O(t)$ 中的谓词变量包含 t 的所有输出库所, 称 $f_O(t)$ 为逻辑输出表达式;

(3) $F \subseteq (P \times T) \cup (T \times P)$ 是一个弧的集合;

(4) I 是一个从逻辑输入变迁到逻辑输入表达式的映射, 即 $\forall t \in T_I, I(t) = f_I(t)$;

(5) O 是一个从逻辑输出变迁到逻辑输出表达式的映射, 即 $\forall t \in T_O, O(t) = f_O(t)$;

(6) $M: P \rightarrow \{0, 1\}$ 是一个标识函数;

(7) 变迁发生规则:

① 对 $\forall t \in T_D$, 变迁发生规则与普通 Petri 网中变迁发生规则相同;

② 对 $\forall t \in T_I$, 若 $f_I(t) |_M = .T.$, 即 t 的输入库所集在标识 M 下满足 t 的逻辑输入表达式 $f_I(t)$, 则 $M[t]$, 并且, 若 $M[t]M'$, 则 $\forall p \in \cdot t: M'(p) = 0$; $\forall p \in t \cdot: M(p) = 0, M'(p) = 1$; 且 $\forall p \notin \cdot t \cup t \cdot: M'(p) = M(p)$;

③ 对 $\forall t \in T_O$, 若 $\forall p \in \cdot t: M(p) = 1$, 则 t 是使能

的,并且,若 $M[t]M'$, 则 $\forall p \in t: M'(p) = M(p) - 1$; $\forall p \notin t \cup t: M'(p) = M(p)$; 且 $\forall p \in t: f_o(t) \mid_{M'} = .T.$, 即 t 的输出库所集在标识 M' 下满足 t 的逻辑输出表达式 $f_o(t)$.

逻辑 Petri 网通过附加在逻辑变迁中的逻辑表达式控制网中的流程转移,为了与普通变迁进行区别,在逻辑变迁上添加 I 或者 O 表示其变迁类型,图 1(a)为一个逻辑输入变迁, (b)为一个逻辑输出变迁.图 1 中 f_1, f_2, f_3 以及 f_4 为逻辑 Petri 网的部分子网流程, (a)中的逻辑输入变迁标注 $(p_1 \vee p_2) \wedge p_3$, 表示流程 f_1 和 f_2 之一与 f_3 同时完成后即可触发后续流程 f_4 ; (b)中的逻辑输出变迁标注 $p_2 \wedge \neg p_3 \vee \neg p_2 \wedge p_3$, 表示当流程 f_1 执行完毕后,可以触发且只能触发流程 f_2 与 f_3 之一.

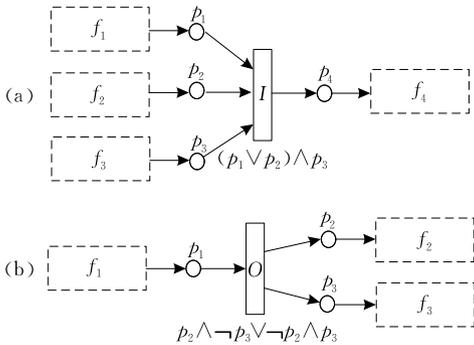


图 1 逻辑变迁示意图

3.2 服务网及相关概念

借鉴工作流网的建模方法^[1],采用逻辑 Petri 网描述服务流程,将 Web 服务流程建模为服务网,下面给出服务网的定义.

定义 6. 服务网^[13]为一个标记 LPN,表示为 $SN = (LPN, i, o, \varphi, \Gamma)$, 其中:

(1) LPN 是一个逻辑 Petri 网,用于描述 Web 服务流程的过程模型,具有唯一起始库所 i 和终止库所 o ;

(2) $P = P_D \cup P_I \cup P_C$, P_D 为外部数据库所集合,用于与外部服务进行数据交互; P_I 为内部数据库所集合,用于服务流程内部的数据交互; P_C 为控制库所集合,用于描述服务流程的转移;

(3) $T = T_s \cup T_I \cup T_O$, T_s 为服务变迁集合,表示服务流程的所有组成 Web 服务, T_I 和 T_O 为逻辑输入和输出变迁集合,用于控制服务流程中的业务流;

(4) M 称为 SN 的标识,令 M_0 为初始标识,且 $M_0(i) = 1, \forall p \in P - \{i\}: M_0(p) = 0$;

(5) $\varphi: T \rightarrow \Gamma$ 是一个映射函数,其中 Γ 表示流程中的变迁名称集合.

为了简化服务网模型,当服务网中的两个服务变迁之间同时存在控制库所和内部数据库所时,将二者合并表示为控制库所.图 2 为一个服务网,描述了电子商务中买卖双方交易流程,为了清晰描述服务流程,将流程中的服务变迁和数据库所进行编号,具体含义参见表 1 及表 2.

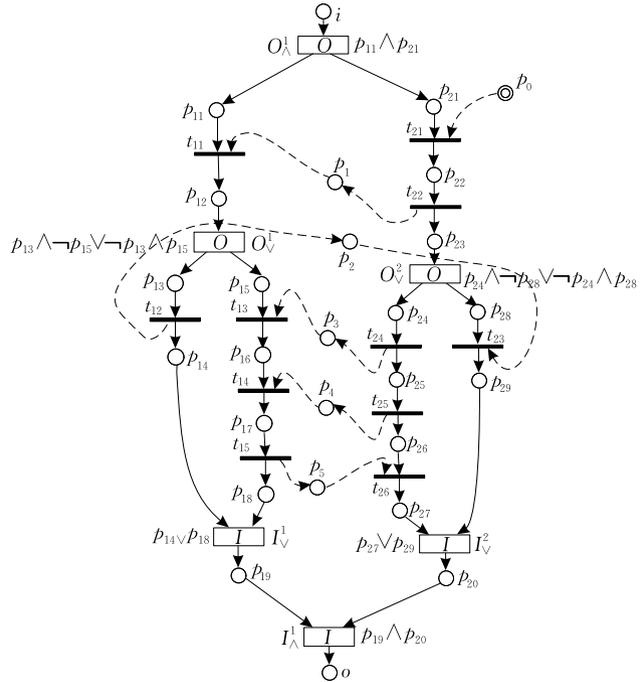


图 2 电子商务交易流程服务网

表 1 服务变迁编号及含义

变迁 ID	Web 服务名称	含义
t_{11}	s_query	卖家接受查询
t_{12}	s_query_fail	卖家进行查询失败处理
t_{13}	s_book	卖家接受商品订购
t_{14}	s_pay	卖家接受货款支付
t_{15}	s_delivery	卖家发货
t_{21}	b_login	买家登录
t_{22}	b_query	买家查询商品
t_{23}	b_query_fail	买家查询失败处理
t_{24}	b_book	买家订购商品
t_{25}	b_pay	买家支付货款
t_{26}	b_accept	买家接收商品

表 2 数据库所编号及含义

库所 ID	库所名称	含义
p_1	q_info	商品查询信息
p_2	query_f	查询失败信息
p_3	order	商品订单
p_4	pay	支付数据
p_5	delivery	商品配送数据
p_0	b_login	买家登录信息(外部数据)

服务网中的库所 $p_{11} \sim p_{19}$ 以及 $p_{20} \sim p_{29}$ 为控制库所,表示流程内部操作状态之间的转换;库所 $p_1 \sim p_5$ 由虚线连接,表示内部数据库所及其数据转

移关系, 库所 p_0 为外部数据库所, 表示服务流程需要从外部接受数据, 为了区别于其他库所, 以双圈进行标识.

将服务网中的外部数据库所集合删除后得到的网结构称为服务内网, 将服务网的外部数据库所和内部数据库所同时删除, 得到的网结构称为控制流网, 详见定义 7 和定义 8. 服务内网描述了服务流程之间的操作转移及其内部数据交互, 控制流网则仅保留了服务流程中的操作转移关系, 图 3 为图 2 中服务网所对应的控制流网.

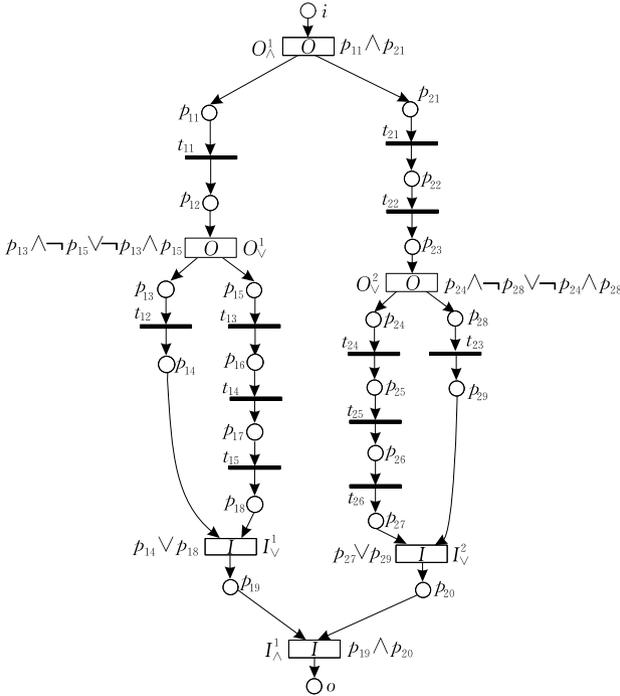


图 3 电子商务交易流程控制流网

定义 7. 若 $SN = (LPN, M, i, o, \varphi, \Gamma)$ 为一个服务网, 令 $LPN' = (P', T', F', I, O)$, 其中 $P' = P_c \cup P_t, T' = T, F' = F - \{P_D \times T \cup T \times P_D\}$, 则称 $SIN = (LPN', i, o, \varphi, \Gamma)$ 为服务网 SN 的服务内网.

定义 8. 若 $SN = (LPN, M, i, o, \varphi, \Gamma)$ 为一个服务网, 令 $LPN'' = (P'', T'', F'', I, O)$, 其中 $P'' = P_c, T'' = T, F'' = F - \{P_D \cup P_t\} \times T \cup T \times \{P_D \cup P_t\}$, 则称 $CFN = (LPN'', i, o, \varphi, \Gamma)$ 为服务网 SN 的控制流网.

根据 Web 服务业务流程执行语言及工作流^[1]相关知识可知, 任意复杂的服务流程可由顺序、并行、选择以及循环 4 种基本结构组合而成. 令 t_1 和 t_2 为两个原子 Web 服务, 由上述 4 种基本结构表示的服务网如图 4 所示. 本文将由两个原子服务组成的顺序、选择以及并行和一个原子服务组成的循环结构型的服务网称为原子型服务网.

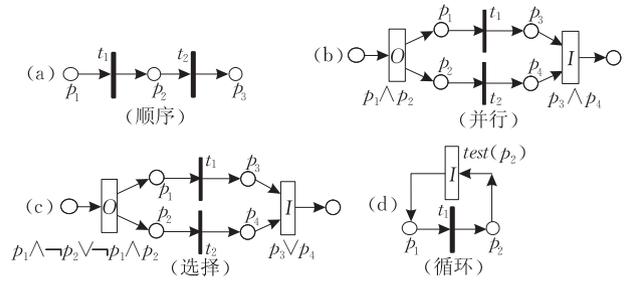


图 4 服务网的 4 种基本结构

并发流程中的逻辑输入和输出变迁对应于工作流^[1]中路由构件 or-split 和 or-join, 选择流程中的逻辑输入和输出变迁对应于路由构件 and-split 和 and-join, 上述逻辑变迁类型分别记为 $O_\wedge, I_\wedge, O_\vee$ 和 I_\vee . 为规范流程建模, 下面给出逻辑表达式为标准型及协同表达式的定义.

定义 9. $F = \{f_1, f_2, \dots, f_n\}$ 是一组服务网, t_o 和 t_i 是连接 F 的逻辑变迁, $\forall f_j: \pi_1(f_j, i) = \{t_o\}$ 且 $\forall f_j: \pi_2(f_j, o) = \{t_i\} (1 \leq j \leq n)$. 将以下形式的逻辑表达式定义为对应类型逻辑变迁的逻辑表达式的标准型.

- (1) $O_\vee: O(t_o) = f_1.i \wedge \neg f_2.i \wedge \neg \dots \wedge \neg f_n.i \vee \neg f_1.i \wedge f_2.i \wedge \neg \dots \wedge \neg f_n.i \vee \dots \vee \neg f_1.i \wedge \neg f_2.i \wedge \neg \dots \wedge f_n.i$;
- (2) $O_\wedge: O(t_o) = f_1.i \wedge f_2.i \wedge \dots \wedge f_n.i$;
- (3) $I_\vee: I(t_i) = f_1.o \vee f_2.o \vee \dots \vee f_n.o$;
- (4) $I_\wedge: I(t_i) = f_1.o \wedge f_2.o \wedge \dots \wedge f_n.o$.

给定服务网 SN , 定义运算符 ∂ 来获取逻辑变迁的类型, 对任意 $t \in SN, T_I \cup SN, T_O, \partial(t) \in Lg, Lg = \{O_\vee, O_\wedge, I_\vee, I_\wedge\}$.

定义 10. $F = \{f_1, f_2, \dots, f_n\}$ 是一组服务网, t_o 和 t_i 是连接 F 的逻辑变迁, $\forall f_j: \pi_1(f_j, i) = \{t_o\}$ 且 $\forall f_j: \pi_2(f_j, o) = \{t_i\} (1 \leq j \leq n)$. 若 $\partial(t_o) = O_\vee \wedge \partial(t_i) = I_\vee$ 或 $\partial(t_o) = O_\wedge \wedge \partial(t_i) = I_\wedge$, 则称 $O(t_o)$ 和 $I(t_i)$ 为协同表达式, 记为 $O(t_o) \diamond I(t_i)$.

为了便于后文描述, 引入符号 \blacklozenge 表示两个逻辑表达式为非协同逻辑表达式. 若 $O(t_o)$ 和 $I(t_i)$ 为非协同表达式, 记为 $O(t_o) \blacklozenge I(t_i)$.

定义 11. SN 为一服务网, $\forall t_o \in SN, \exists t_i: O(t_o) \diamond I(t_i)$ 且 $\forall t_i \in SN, \exists t_o: I(t_i) \diamond O(t_o)$, 则称 SN 为良构流程.

良构流程的概念被应用于工作流^[1]和 Web 服务流程^[20]建模中, 良构是指两个流程若以 or-split 开始分支, 则必须以 or-join 进行流程汇集; 若以 and-split 开始分支, 则必须以 and-join 进行流程汇集, 并且匹配的逻辑变迁具有协同表达式.

图 5 为两种非良构流程,其中图 5(a)流程中的逻辑变迁没有成对匹配出现,图 5(b)流程中的逻辑变迁 t_{o1} 与 t_{i1} 虽成对出现,但二者标注的逻辑表达式为非协同表达式。

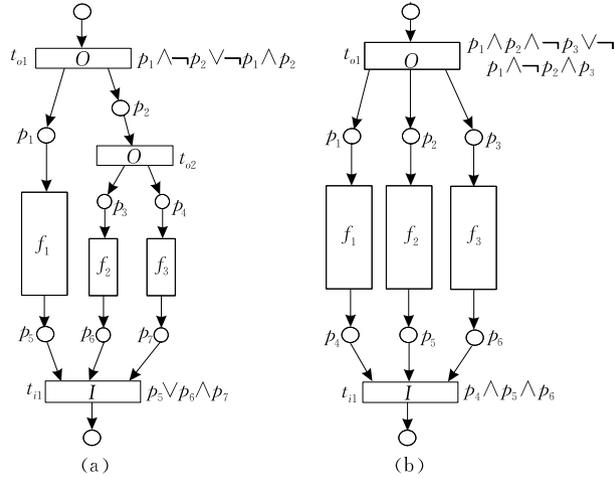


图 5 非良构的服务流程示例

4 Web 服务流程的结构范式

4.1 结构范式的定义

由于需要协调和管理参与组合的各个 Web 服务,在组合得到的服务流程中容易存在逻辑结构混乱、路径不可达、数据流交互死锁以及结构冗余等结构不合理的问题。然而,并非所有存在问题的流程均没有利用价值,为了对流程结构的合理性做出更为科学的定量评价,本文提出将已有研究工作中的结构健壮性进行层次划分,建立不同级别的健壮性判定准则。

结构范式是在服务网的基础上对流程结构健壮性层次划分准则的一种形式化表达。结构范式可以用来评价服务流程的结构合理性,进而有助于改进服务组合的合理性和可靠性。

在服务流程的建模过程中,服务内网描述了删除外部数据请求的服务流程,可以用来判定服务流程的逻辑结构和内部数据交互是否存在缺陷,本文从流程中的路径可达性和结构冗余性两个角度将 Web 服务流程结构范式划分为四个层次。

为了便于描述,令 SP 表示一个 Web 服务流程, SN 、 SIN 与 CFN 分别为 SP 的服务网、服务内网和控制流网, G_f 表示 SN 的终止状态,即 G_f 为 SN 对应 $M(o)=1$ 标识状态。

定义 12. 若 SIN 中存在变迁序列 σ ,使得 $M_0[\sigma]G_f$,则称 SP 满足第一范式,记为 1NF。

定义 13. 若 SP 满足 1NF,且 SIN 是无阻塞

的,则称 SP 满足第二范式,记为 2NF。

定义 14. 若 SP 满足 2NF,且 SIN 中无等价子控制流网,则称 SP 满足第三范式,记为 3NF。

定义 15. 若 SP 满足 3NF,且 SIN 中无重复子控制流网,则称 SP 满足第四范式,记为 4NF。

满足第一范式的服务流程中至少存在一条从起始服务至终止服务的连通服务流程路径,从而可以对外提供服务,这是服务流程结构合理的最低层次;第二范式要求服务流程是无阻塞的,即所有潜在的服务流程路径都是可以到达并正确地执行,无阻塞服务网参见定义 16。

第三范式要求流程中不存在等价服务流程,等价服务流程(定义 17)是指在流程的选择和并行结构中存在两段功能等价的子服务流程。等价服务流程的出现造成了服务网结构的冗余,并可引发服务网语义表达错误。例如,一个服务网的选择结构中出现了两个等价子流程,选择结构会在不同条件下分别执行等价流程中的一个,而这两个流程的功能相同,因此该选择结构是无意义的。

第四范式要求在流程中不存在重复服务流程,重复服务流程是指两段完全相同的服务流程,为了流程模型表达的简介,重复服务流程应该被抽象映射为一个子流程结点,有关重复流程的定义可参见定义 18。

定义 16. $\langle SN, G_f \rangle$ 是无阻塞的,当且仅当 $\forall M' \in R(M_0), \exists \sigma \in T^*: M'[\sigma]G_f$ 。

定义 17. $CFN_1 = (LPN_1, M_1, i_1, o_1, \varphi_1, \Gamma_1)$ 和 $CFN_2 = (LPN_2, M_2, i_2, o_2, \varphi_2, \Gamma_2)$ 为两个控制流网,称 CFN_1 和 CFN_2 为等价控制流网当且仅当存在映射 τ 使得如下条件成立:

- (1) $\forall x \in P_1 \cup T_1 \Leftrightarrow \tau(x) \in P_2 \cup T_2$;
- (2) $\forall (x, y) \in F_1 \Leftrightarrow (\tau(x), \tau(y)) \in F_2$;
- (3) $\forall t \in T_1, \tau(t) \in T_2; I(\tau(t)) = \Psi(I(t), \Omega_{I(t)}, \tau)$;

(4) $\forall t \in T_{O_1}, \tau(t) \in T_{O_2}; O(\tau(t)) = \Psi(O(t), \Omega_{O(t)}, \tau)$;

- (5) $\forall t \in T_{S_1}, \tau(t) \in T_{S_2}; \varphi_1(t) \Leftrightarrow \varphi_2(\tau(t))$;

若 CFN_1 和 CFN_2 为等价控制流网,则记为 $CFN_1 \sim CFN_2$ 。

定义 18. $CFN_1 = (LPN_1, M_1, i_1, o_1, \varphi_1, \Gamma_1)$ 和 $CFN_2 = (LPN_2, M_2, i_2, o_2, \varphi_2, \Gamma_2)$ 为两个控制流网,对给定正整数 n ,称 CFN_1 和 CFN_2 为重复控制流网当且仅当如下条件成立:

- (1) $LPN_1 = LPN_2$;
- (2) $i_1 = i_2 \wedge o_1 = o_2$;

(3) $\Gamma_1 = \Gamma_2 \wedge \varphi_1 = \varphi_2$;

(4) $|LPN_1.T_s| \geq n$;

定义 18 中(4)要求服务网过程模型中的服务数目大于指定的整数 n , 是对重复流程中包含的服务数目做约束, 即只有大于 n 时才认为两个服务流程是重复流程; 在小于或等于 n 时, 即使两个流程片段完全相同也不被认作重复流程。

4.2 结构范式的判定

4.2.1 控制流网语言

服务流程的结构范式是根据服务内网中流程路径结构性进行划分的, 为了判定服务流程的结构范式, 须获取服务内网中的流程路径, 而服务内网中的流程路径对应于控制流网中从起始库所到终止库所的可触发变迁序列。

将 Petri 网的变迁集 T 映射到某个字母表 Γ (每个变迁对应一个字母), 该 Petri 网的所有可能发生的变迁序列映射到 Γ^* 的字符串的集合称为 Petri 网的语言, 从 Petri 网语言可以方便地获取所有可触发变迁序列. 传统的 Petri 网语言求解方法通常是通过遍历 Petri 网的覆盖性树或可达标识图获得. 当 Petri 网规模较大时, 其对应的覆盖性树或可达标识图中的状态过多, 存在状态空间爆炸问题, 因此求解 Petri 网语言的复杂度过高。

为避免上述问题, 本文采取合并化简的方法求解控制流网语言(CFN-L), 从而获取所有可触发变迁序列, 然后再将内部数据库所加入到控制流网, 判定这些可触发变迁序列的可达性及其结构性, 进而判定对应服务流程的结构范式. 一个控制流网 CFN 的 CFN-L 表示为 $L(CFN)$, 同时将 $L(CFN)$ 对应的控制流网记为 $N_{sp}(L(CFN))$. CFN-L 求解约简规则如下:

从起始库所 i 开始搜索 CFN, 令 t_1 为当前处理变迁, $t_2 = \pi_2(\pi_2(t_1))$:

(1) 若 $t_1 \in CFN.T_s, t_2 \in CFN.T_s$, 则约简为 $t_1 t_2$.

(2) 若 $t_1 \in CFN.T_s, t_2 \in CFN.T_o$, 则 t_2 为逻辑输出变迁, 假设其关联子流程 Sp_1, Sp_2, \dots, Sp_m , 令 $L(Sp_i) = s_i$, 约简为 $t_1 t_2 s_1 \parallel / \otimes s_2 \parallel / \otimes \dots s_m$, 其中 s_i 与 s_{i+1} 之间的连接符 \parallel 或 \otimes 由 $O(t_2)$ 中的逻辑运算符映射而得, \parallel 对应 \wedge , \otimes 对应 \vee .

(3) 若 $t_1 \in CFN.T_s, t_2 \in CFN.T_1$, 则 t_2 为逻辑输入变迁, 假设其关联子流程 Sp_1, Sp_2, \dots, Sp_m , 且 $L(Sp_i) = s_i$ 求解完毕, 约简为 $s_1 \parallel / \otimes s_2 \parallel / \otimes \dots s_m t_2$, 其中 $t_1 \in s_j, 1 \leq j \leq m$.

(4) 若 $t_1 \in CFN.T_1, t_2 \in CFN.T_s$, 则约简为 $t_1 t_2$.

(5) 若 $\pi_2(t_1) = o$, 则约简完毕。

(6) 由逻辑变迁关联的子流程采取规则(1)~(5)递归约简, 若在约简过程中选择和并发结构是非原子型的, 对其分支流程产生的 CFN-L 添加括号以便增加控制流网语言的可读性。

例如, 令 CFN_1 表示图 3 中控制流网, 则 $L(CFN_1) = O_\wedge^1(t_{11} O_\vee^1 t_{12} \otimes (t_{13} t_{14} t_{15}) I_\vee^1) \parallel (t_{21} t_{22} O_\vee^2 t_{23} \otimes (t_{24} t_{25} t_{26}) I_\vee^2) I_\wedge^1$.

4.2.2 相关结构判定定理

定义 19. $\Pi_{X \rightarrow Y}: X^* \rightarrow Y^*$ 称为从 X 到 Y 投影, 当且仅当 $\forall \sigma \in X^*, \Pi_{X \rightarrow Y}(\sigma)$ 是将 X - Y 中所有字符从 σ 删除后得到的字符串^[31].

定义 20. 字符串 $\sigma \subseteq L(CFN)$, 若存在字符 $x, y \in \mathcal{S}(\sigma): \partial(x) \in \{O_\vee, O_\wedge\} \wedge \partial(y) \in \{I_\vee, I_\wedge\} \wedge O(x) \diamond I(y)$, 则将 x 和 y 从 σ 中同时删除, 上述操作称为逻辑熔合, 记为 $\Delta(\sigma)$.

定理 1. ϵ 为空字符串, 给定控制流网 CFN 与字符集合 $\Sigma = \varphi(CFN.T_i \cup CFN.T_o)$, 若 $\Delta(\Pi_{\Gamma \rightarrow \Sigma}(L(CFN))) = \epsilon$, 则 CFN 是良构的。

证明. 假设 CFN 是非良构, 则 $\exists t_o \in SN, \forall t_i: O(t_o) \blacklozenge I(t_i)$ 或者 $\exists t_i \in SN, \forall t_o: I(t_i) \blacklozenge O(t_o)$. 令 $\sigma = \Pi_{\Gamma \rightarrow \Sigma}(L(CFN))$, 则 $\exists x \in \mathcal{S}(\sigma)$, 对 $\forall y \in \mathcal{S}(\sigma)$, 不存在 $O(x) \diamond I(y)$, 所以 x 无法被逻辑熔合, 即 $\Delta(\Pi_{\Gamma \rightarrow \Sigma}(L(CFN)))$ 至少包含字符 x , 这与 $\Delta(\Pi_{\Gamma \rightarrow \Sigma}(L(CFN))) = \epsilon$ 矛盾, 因此假设不成立, CFN 是良构的. 证毕。

引理 1. 顺序型控制流网无阻塞。

证明. 假设 F 是由一组服务变迁 $\{t_1, t_2, \dots, t_n\}$ 组成的顺序型控制流网, 且 $F.i = \pi_1(t_1), \forall t_i: \pi_2(t_i) = \pi_1(t_{i+1}), 1 \leq i \leq n-1$. 令 $M_0(F.i) = 1, \forall M \in R(M_0)$, 若 $M(t_j) = 1, 1 \leq j \leq n$, 则 $\exists \sigma = t_{j+1} t_{j+2} \dots t_n$, 使得 $M[\sigma]G_f$. 因此 F 是无阻塞的.

证毕。

引理 2. 良构原子选择型控制流网无阻塞。

证明. F 是由一组服务变迁 $\{t_1, t_2, \dots, t_n\}$ 与逻辑变迁 t_o 及 t_i 组成的良构原子选择型控制流网, 即 $\forall t_j: \pi_1(t_j.i) = \{t_o\}, \forall t_j: \pi_2(t_j.o) = \{t_i\}$, 且 $O(t_o) = t_1.i \wedge \neg t_2.i \wedge \dots \wedge \neg t_n.i \vee \neg t_1.i \wedge t_2.i \wedge \dots \wedge \neg t_n.i \vee \dots \vee \neg t_1.i \wedge \neg t_2.i \wedge \dots \wedge t_n.i, I(t_i) = t_1.o \vee t_2.o \vee \dots \vee t_n.o$.

令 $M_0(F.i) = 1, M_0[t_o]M'$, 若 t_o 触发后执行 t_p , 则 $M'[t_p]M'', M''(t_p.o) = 1$, 而 $\pi_2(t_p.o) = \{t_i\}$ 且 $I(t_i) = t_1.o \vee t_2.o \vee \dots \vee t_n.o$, 由 LPN 触发规则可知 $M''[t_i]G_f$, 这里 $1 \leq p \leq n$. 由 p 取值的任意性可知, F 是无阻塞的. 证毕。

引理 3. 良构原子并行型控制流网无阻塞.

证明. 方法类似引理 2, 不再叙述. 证毕.

引理 4. 原子循环型控制流网无阻塞.

证明. F 是由服务变迁 t_1 逻辑变迁 t_i 组成的原子循环型控制流网, 且 $F.i = \pi_1(t_1) \wedge F.o = \pi_2(t_1)$. F 中只存在两种标识: $M_0(F.i) = 1$ 与 $G_f = M(F.o) = 1$, 且 $M(F.o) = 1 \wedge test(F.o) = 1$ 时, $G_f[t_i]M_0$, 而 $M_0[t_1]G_f$, 因此 F 是无阻塞的. 证毕.

定理 2. 良构控制流网无阻塞.

证明. 若 F 是良构原子选择型控制流网, 由引理 2 知 F 是无阻塞的, 即若 $M(F.o) = 1$, 则存在 $\sigma \in F.T^*$, 使得 $M[\sigma] > G_f$, 因此 F 等价于一个服务变迁 t . 同理, 原子并行型与原子循环型控制流网也等价于一个服务变迁. 将良构控制流网中的上述 3 种结构利用服务变迁进行等价替换, 最终可获得一个顺序结构的控制流网, 由引理 1 可知, 该控制流网是无阻塞的, 定理得证. 证毕.

定义 21. 若服务网中的变迁 t 连接了发送数据的库所 p , 则 t 称为库所 p 的发送变迁, 记作 $s(p)$; 若服务网中的变迁 t 连接了接受数据的库所 p , 则 t 称为库所 p 的接受变迁, 记作 $r(p)$;

定义 22. SN 是一个服务网, C 称为从节点 n_1 到节点 n_k 的一条路径当且仅当在 SN 中存在节点序列 $\langle n_1, n_2, \dots, n_k \rangle$, 使得 $(n_i, n_{i+1}) \in F, 1 \leq i \leq k$. 用 $\&(C)$ 表示路径 C 的字母表, 即 $\&(C) = \{n_1, n_2, \dots, n_k\}$.

引入符号 $<$ 表示两个字符存在后继关系, 在 C 中, n_j 称为 n_i 的后继, 记作 $n_i <_C n_j$, 当且仅当存在 $n_i, n_{i+1}, \dots, n_j \in \&(C)$, 且 $(n_i, n_{i+1}), (n_{i+1}, n_{i+2}), \dots, (n_{j-1}, n_j) \in F$.

从上述定义和 CFN-L 的构造规则可知, 一个服务网的 CFN-L 展开得到的语言表达式集合对应于该服务网从起始库所 i 到终止库所 o 的所有路径.

定理 3. 假设一个服务内网中存在 n 条从起始库所 i 到终止库所 o 的路径, 任取其中的 m 条路径, 编号为 C_1, C_2, \dots, C_m , 若两条路径 C_i 与 C_{i+1} 之间存在内部库所 p , 使得 $s(p) \in C_i \wedge r(p) \in C_{i+1}$, 将这些库所命名为集合 $P_i^j (1 \leq i < m)$, 在集合 P_i^j 到 P_i^m 中任取一个库所, 分别记为 $p_1 p_2 \dots p_m$, 若存在 $j (1 \leq j \leq m)$ 使得 $C_j : s(p_j) <_{C_j} r(p_{(m+j-1) \% (m+1)})$, 则 C_1, C_2, \dots, C_m 中无环.

证明. C_1, C_2, \dots, C_m 中每条路径中无环, 若存在环路, 则必定是由 C_1, C_2, \dots, C_m 路径的一部分与 $p_1 p_2 \dots p_m$ 及其连接弧构成, 因此路径 $C_i, p_i, r(p_i), s(p_{i+1})$ 以及 p_{i+1} 之间的流关系必须同向. 由于 $p_i <$

$r(p_i)$ 且 $s(p_{i+1}) < p_{i+1}$, 只有 $r(p_i) <_{C_j} s(p_{i+1})$, 路径 $C_i, p_i, r(p_i), s(p_{i+1})$ 以及 p_{i+1} 之间才存在同向流, 即 $\forall C_i : r(p_i) <_{C_j} s(p_{i+1})$ 才存在环. C_1, C_2, \dots, C_m 中若有一条路径不满足上述规则便不存在环, 即存在 $j (1 \leq j \leq m)$ 使得 $C_j : s(p_j) <_{C_j} r(p_{(m+j-1) \% (m+1)})$, 定理得证. 证毕.

定理 4. 服务内网 SIN 是良构并且无环, 则 $\langle SIN, G_f \rangle$ 是无阻塞的.

证明. 构建 SIN 的控制流网 CFN , 因为 SIN 良构, 所以 CFN 良构, 因此 CFN 是无阻塞的.

由于 $SIN = CFN \cup P_D \times T \cup T \times P_D$, 即在 CFN 加入了内部数据库所及其流程关系. 对一个处于使能状态下的变迁, 增加一个带有托肯的库所将不会影响其使能及触发, 因此, 若 SIN 内部数据库均能得到托肯, 则 SIN 中所有变迁的触发性等同 CFN , 即无阻塞的. 根据资源请求死锁判定条件, 如果不存在环路等待, 则不会出死锁, 因此如果 SIN 中无环结构, 则所有数据库所均能获取托肯, 此时 $\langle SIN, G_f \rangle$ 是无阻塞的. 证毕.

定义 23. CFN_1 和 CFN_2 为两个控制流网, 令 $L(CFN_1) = x_{11} x_{12} x_{13} \dots x_{1n}$, $L(CFN_2) = x_{21} x_{22} \dots x_{2i} \dots x_{2n}$, 当且仅当存在映射 θ 使得如下条件成立时, $L(CFN_1)$ 和 $L(CFN_2)$ 称为等价控制流网语言, 记为 $L(CFN_1) \simeq L(CFN_2)$:

- (1) $\forall t \in T_1 \Leftrightarrow \theta(t) \in T_2$;
- (2) $\varphi_1(t) \Leftrightarrow \varphi_2(\theta(t))$;
- (3) $x_{1i} = \varphi_1(t) \Leftrightarrow x_{2i} = \varphi_2(\theta(t))$;
- (4) $x_{1i} = \parallel \Leftrightarrow x_{2i} = \parallel$ or $x_{1i} = \otimes \Leftrightarrow x_{2i} = \otimes$.

定理 5. CFN_1 和 CFN_2 为两个控制流网, 若 $L(CFN_1) \simeq L(CFN_2)$, 则 $CFN_1 \sim CFN_2$.

证明. 令 $L(CFN_1) = x_{11} x_{12} \dots x_{1i} \dots x_{1n}$, $L(CFN_2) = x_{21} x_{22} \dots x_{2i} \dots x_{2n}$. 因为 $L(CFN_1) \simeq L(CFN_2)$, $\forall \sigma_1 = x_{1j} x_{1j+1} \dots x_{1m}, \sigma_2 = x_{2j} x_{2j+1} \dots x_{2m}$ 满足 $\sigma_1 \simeq \sigma_2$.

若 σ_1 与 σ_2 为顺序结构或者循环结构, 不涉及逻辑表达式, 由于 $\sigma_1 \simeq \sigma_2$ 可得: (1) $\forall t \in T_1 \Leftrightarrow \theta(t) \in T_2$; (2) $\varphi_1(t) \Leftrightarrow \varphi_2(\theta(t))$; (3) $x_{1i} = \varphi_1(t) \Leftrightarrow x_{2i} = \varphi_2(\theta(t))$; 令 $\tau = \theta$, 由定义可知 $N_{s_p}(\sigma_1) \sim N_{s_p}(\sigma_2)$.

若 σ_1 与 σ_2 为原子选择结构, 令 $\sigma_{11} \sigma_{12} \dots \sigma_{1p}$ 与 $\sigma_{21} \sigma_{22} \dots \sigma_{2p}$ 为二者的组成分支, 由于 $\sigma_{1k} \simeq \sigma_{2k}$ 且是顺序结构或者循环结构的, 因此 $N_{s_p}(\sigma_{1k}) \sim N_{s_p}(\sigma_{2k})$ 且 $\tau = \theta$. 由定义 23 可知, σ_1 与 σ_2 中的字符 \parallel 和 \otimes 对应出现, 而 \parallel 与 \otimes 对应于流程中的逻辑表达式中的 \vee 与 \wedge , 即逻辑表达式连接了等价流程, 因此满足 $\forall t \in T_1, \tau(t) \in T_2 : I(\tau(t)) = \Psi(I(t), \Omega_{I(t)}, \tau)$ 和 $\forall t \in$

$T_{O_1}, \tau(t) \in T_{O_2}; O(\tau(t)) = \Psi(O(t), \Omega_{O(t)}, \tau)$, 由定义 17 可知 $N_{sp}(\sigma_1) \sim N_{sp}(\sigma_2)$.

同理证明原子并发结构也满足 $N_{sp}(\sigma_1) \sim N_{sp}(\sigma_2)$. 根据服务网的建模规则和控制流网语言的求解方法可知, 对任意结构的 σ_1 与 σ_2 , 若 $\sigma_1 \simeq \sigma_2$, 则 $N_{sp}(\sigma_1) \sim N_{sp}(\sigma_2)$, 因此定理得证. 证毕.

定义 24. σ 为字符串, $\forall x, y \in \mathcal{S}(\sigma), \sigma' \sqsubseteq \sigma$, 若 $\sigma' = x \cdot \bar{\sigma} \cdot y$, 称 $\bar{\sigma}$ 为字符串 σ 基于字符 x 与 y 的切片, 记为 $\rho(\sigma, x, y)$.

定义 25. σ 所有基于字符 x 与 y 的切片组成一个字符串集合, 称为切集, 记为 $\sigma/(x, y)$.

定义 26. σ 为字符串, 字符集合 $C \subset \mathcal{S}(\sigma), \sigma$ 被 C 中包含的字符分割为若干字符串, 由这些字符串组成的集合称为 σ 关于 C 的割集, 记为 $\mathfrak{R}(\sigma, C)$.

例如, $\sigma = \text{"abpcdapmbqcdpabc"}$, 则 $\sigma/(p, q) = \{\text{cdapmbqcdpabc}, \text{cdapmb}, \text{mbqcdpabc}, \text{mb}, \text{abc}\}$. 令 $C = \{p, q\}$, 则 $\mathfrak{R}(\sigma, C) = \{\text{ab}, \text{cda}, \text{mb}, \text{cd}, \text{abc}\}$.

为了判定 CFN-L 中是否存在等价流程, 首先利用 CFN-L 中的匹配逻辑变迁对进行切片, 获取其中的并发或选择子流程结构对应 CFN-L, 然后再利用字符集 $C = \{\parallel, \otimes\}$ 去求解切片的割集, 获取流程结构中的具体子流程的 CFN-L, 从而可以根据定理 5 进行是否存在等价流程的判定.

将控制流网 $CFN = (LPN, M, i, o, \varphi, \Gamma)$ 的匹配逻辑变迁对组成集合定义为 $T_{match} = \{(t_1, t_2) \mid t_1 \in T_O, t_2 \in T_I, O(t_1) \diamond I(t_2)\}$. 根据 CFN-L 的生成规则, 在进行切片割集求解时只能对不包含在括号内的字符 \parallel 和 \otimes 进行割运算, 为区别定义 26, 以符号 $\overline{\mathfrak{R}}(\sigma, C)$ 进行表示.

例如, $L(CFN_1) = O_{\lambda}^1(t_{11} O_{\vee}^1 t_{12} \otimes (t_{13} t_{14} t_{15}) I_{\vee}^1) \parallel (t_{21} t_{22} O_{\vee}^2 t_{23} \otimes (t_{24} t_{25} t_{26}) I_{\vee}^2) I_{\lambda}^1$, 则 CFN_1 的匹配逻辑变迁对集合为 $T_{match} = \{(O_{\lambda}^1, I_{\lambda}^1), (O_{\vee}^1, I_{\vee}^1), (O_{\vee}^2, I_{\vee}^2)\}$, 以 $(O_{\lambda}^1, I_{\lambda}^1)$ 做切片, 可得 $\sigma = L(CFN_1)/(O_{\lambda}^1, I_{\lambda}^1) = (t_{11} O_{\vee}^1 t_{12} \otimes (t_{13} t_{14} t_{15}) I_{\vee}^1) \parallel (t_{21} t_{22} O_{\vee}^2 t_{23} \otimes (t_{24} t_{25} t_{26}) I_{\vee}^2)$, 再求解 $\overline{\mathfrak{R}}(\sigma, C)$ 可得 $\overline{\mathfrak{R}}(\sigma, C) = \{t_{11} O_{\vee}^1 t_{12} \otimes (t_{13} t_{14} t_{15}) I_{\vee}^1, t_{21} t_{22} O_{\vee}^2 t_{23} \otimes (t_{24} t_{25} t_{26}) I_{\vee}^2\}$.

定理 6. CFN 为一个控制流网, T_{match} 为其匹配逻辑变迁对集合, 存在 $(x, y) \in T_{match}$ 使得 $\sigma_1, \sigma_2 \in \overline{\mathfrak{R}}(L(CFN)/(x, y), C)$ 且 $\sigma_1 \simeq \sigma_2$, 则 CFN 存在等价流程.

证明. 存在 $(x, y) \in T_{match}$ 使得 $\sigma_1, \sigma_2 \in \overline{\mathfrak{R}}(L(CFN)/(x, y), C)$, σ_1 和 σ_2 为由逻辑变迁 x 和 y 连接的两段子流程结构的 CFN-L, 将这两段子控制流网命名为 CFN_1 和 CFN_2 , 由 $\sigma_1 \simeq \sigma_2$ 及定理 4 可知 $CFN_1 \sim CFN_2$. 又 $CFN_1 \subseteq CFN, CFN_2 \subseteq CFN$,

所以 CFN 存在等价流程.

证毕.

4.2.3 结构范式判定算法

在进行服务流程结构范式判定时, 首先需要将其转化为服务网, 然后求解该服务网对应的 CFN-L, 通过 CFN-L 中字符串序列的性质研究服务网的良构、分支连通性、是否存在重复以及等价子控制流网, 进而判定服务流程的结构范式, 具体参见算法 1.

算法 1. Web 服务流程结构范式判定算法.

输入: Web 服务流程

输出: 结构范式级别 SF

Stru_NF(SP)

1. {construct the service net SN and the control flow net CFN of SP ;
2. unfold $L(CFN)$ and build the set C_{path} with all the sequence of $L(CFN)$;
3. $n = Num(C_{path})$;
4. FOR $i = 2$ TO n
5. {Loop_check(i);}
6. IF there exist a loop THEN eliminate the checking paths from C_{path} ;
7. IF ($Num(C_{path}) = 0$) THEN $SF = 0$;
8. IF ($0 < Num(C_{path}) < n$) THEN $SF = 1$;
9. IF ($\Delta(\Pi_{\Gamma \rightarrow \Sigma}(L(CFN))) = \epsilon \wedge SF = 1$) THEN $SF = 2$;
10. FOR $\forall (x, y) \in T_{match}$
11. IF ($\sigma_1, \sigma_2 \in \overline{\mathfrak{R}}(P(CFN)/(x, y), C) \wedge \sigma_1 \simeq \sigma_2$) THEN $Equ_process = \text{True}$;
12. IF ($!Equ_process \wedge SF = 2$) THEN $SF = 3$;
13. IF ($SF = 3 \wedge !Repeatsubstring_check(P(CFN), u)$) THEN $SF = 4$;
14. Return(SF); }

算法首先构建服务流程 SP 的服务网 SN 以及控制流网 CFN , 求解 $L(CFN)$ 并将其展开, 获取 CFN 所有可能存在的从起始库所 i 到终止库所 o 的连通路径集 C_{path} ; 若 C_{path} 中存在 n 条路径, 即 $Num(C_{path}) = n$, 依次分别取 $2, 3, \dots, n$ 条路径, 根据定理 3 判定所取路径中是否存在环, 若某些路径中存在环, 则将所取路径从 C_{path} 中删除.

(1) 若 $Num(C_{path}) = 0$, 则 SF 为 0;

(2) 若 $0 < Num(C_{path}) < n$, 则 SF 为 1;

(3) 若 $Num(C_{path}) = n$, 判定 CFN 是否良构, 若为良构则 SF 为 2;

(4) 在 SF 为 2 的基础上, 判定 CFN 中是否存在等价服务流程, 若不存在则 SF 为 3;

(5) 在 SF 为 3 的基础上, 判定 CFN 是否存在重复子服务流程, 若不存在则 SF 为 4; 利用函数 $Repeatsubstring_check(L(CFN), u)$ 表示对是否

存在重复字符串进行判定,函数检测给定字符串 $L(CFN)$,若其中存在长度大于 u 的重复子串时,返回 True,否则返回 False,该函数可以利用编程语言中提供的字符串处理函数实现。

算法 1 在进行结构范式判定时,对不存在可达路径的服务流程,将其结构范式级别定义为 0,特此说明。

5 应用实例

5.1 满足 4NF 的服务网算例

本节在图 2 给出的电子商务交易系统服务网模型的基础上,对卖方服务流程的功能进行扩充.在买方进行商品查询时,如果查询不成功,除了发送至买方一个查询失败处理外,增加一个可用于卖方采购的子服务流程,该服务流程由服务 s_order 、 s_defray 以及 $s_receive$ 以顺序结构组成,在服务网中分别表示为 t_{31} 、 t_{32} 以及 t_{33} .该服务流程与查询失败处理服务构造并行流程,如图 6 所示。

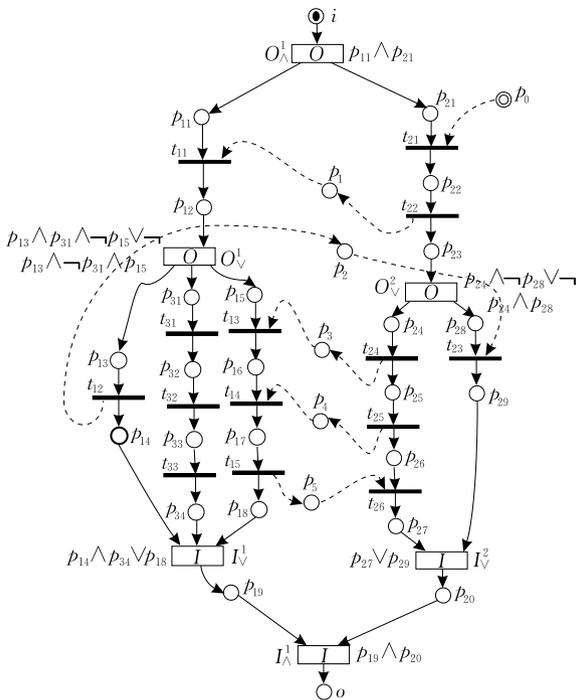


图 6 满足 4NF 的服务网

用 SN_1 和 CFN_1 表示该服务网及其对应的控制流网,则 $L(CFN_1) = O_{\wedge}^1 (t_{11} O_{\vee}^1 (t_{12} \parallel t_{31} t_{32} t_{33}) \otimes (t_{13} t_{14} t_{15}) I_{\vee}^1) \parallel (t_{21} t_{22} O_{\vee}^2 t_{23} \otimes (t_{24} t_{25} t_{26}) I_{\vee}^2) I_{\wedge}^1$, 根据良构流程判定规则,该服务网是良构的。

展开 $L(CFN_1)$ 可以得到路径集 C_{path} , 且 $Num(C_{path}) = 5$, 所有的路径分别为 $path1 = O_{\wedge}^1 t_{11} O_{\vee}^1 t_{12} I_{\vee}^1 I_{\wedge}^1$, $path2 = O_{\wedge}^1 t_{11} O_{\vee}^1 t_{31} t_{32} t_{33} I_{\vee}^1 I_{\wedge}^1$, $path3 =$

$O_{\wedge}^1 t_{11} O_{\vee}^1 t_{13} t_{14} t_{15} I_{\vee}^1 I_{\wedge}^1$, $path4 = O_{\wedge}^1 t_{21} t_{22} O_{\vee}^2 t_{24} t_{25} t_{26} I_{\vee}^2 I_{\wedge}^1$ 以及 $path5 = O_{\wedge}^1 t_{21} t_{22} O_{\vee}^2 t_{23} I_{\vee}^2 I_{\wedge}^1$.

根据结构范式判定算法,该服务流程的控制流网的路径之间不存在环路,可达路径条数为 5,在服务网内部不存在等价控制流网,同时也不存在重复控制流网,因此该服务网的结构范式为 4NF。

5.2 满足 3NF 的服务网算例

现假设在为卖方服务增加采购子服务流程时,并未增加卖方专用采购流程服务 t_{31} 、 t_{32} 以及 t_{33} ,而是使用了买方中的购买服务 t_{24} 、 t_{25} 以及 t_{26} ;上述服务流程对应的服务网变为图 7 所示。

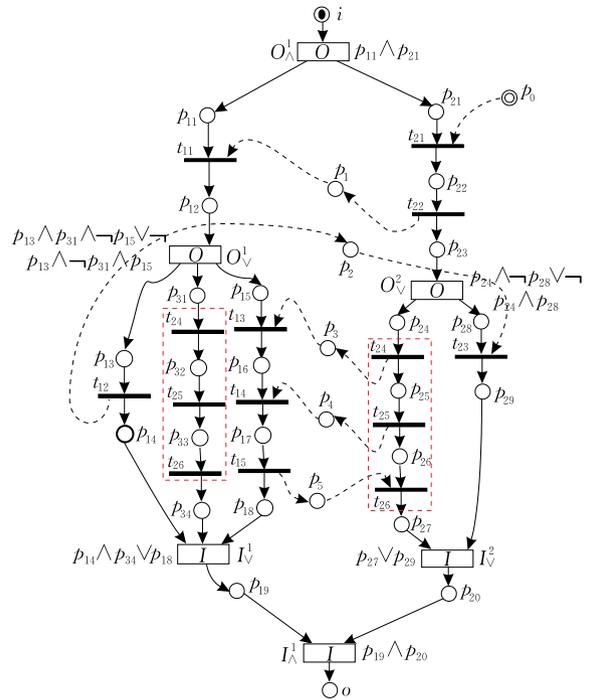


图 7 满足 3NF 的服务网

用 SN_2 和 CFN_2 表示该服务网及其对应的控制流网,则 $L(CFN_2) = O_{\wedge}^1 (t_{11} O_{\vee}^1 (t_{12} \parallel t_{24} t_{25} t_{26}) \otimes (t_{13} t_{14} t_{15}) I_{\vee}^1) \parallel (t_{21} t_{22} O_{\vee}^2 t_{23} \otimes (t_{24} t_{25} t_{26}) I_{\vee}^2) I_{\wedge}^1$, 与上一节中 $L(CFN_1)$ 相比, $L(CFN_2)$ 在路径可达数目以及是否具有等价控制流网上完全相同. 但 $L(CFN_2)$ 中存在重复字符序列 $t_{24} t_{25} t_{26}$, 且其长度为 3, 如果给定用于评价流程结构范式的第 4 层次中允许重复的子流程长度为 2, 则该服务网的结构范式级别为 3; 如果第 4 层次中允许重复的子流程长度为 3, 则该服务网的结构范式级别仍为 4。

5.3 满足 2NF 的服务网算例

为了构建满足 2NF 的服务网,将上述服务流程中的卖方服务单独抽取出一部分,只保留流程“接受预定-收款-发货”和“查询失效处理”两个流程分支,

现拟在流程“接受预定-收款-发货”位置处添加一个“接受预定-发货-收款”的子服务流程以便为买方提供更好的服务。

如图 8, 若在构建流程 f_1 时未将收款和发货顺序调整, 则与 f_0 相同, 命名为 f_2 . 若 f_2 与 f_0 组成并行流程结构, 其对应服务网以及控制流网分别命名为 SN_3 以及 CFN_3 , 则 $L(CFN_3) = t_{11} O_V^1 (t_{12} \otimes O_\Lambda^2 ((t_{13} t_{14} t_{15}) \otimes (t_{13} t_{14} t_{15})) I_V^2) I_V^1$.

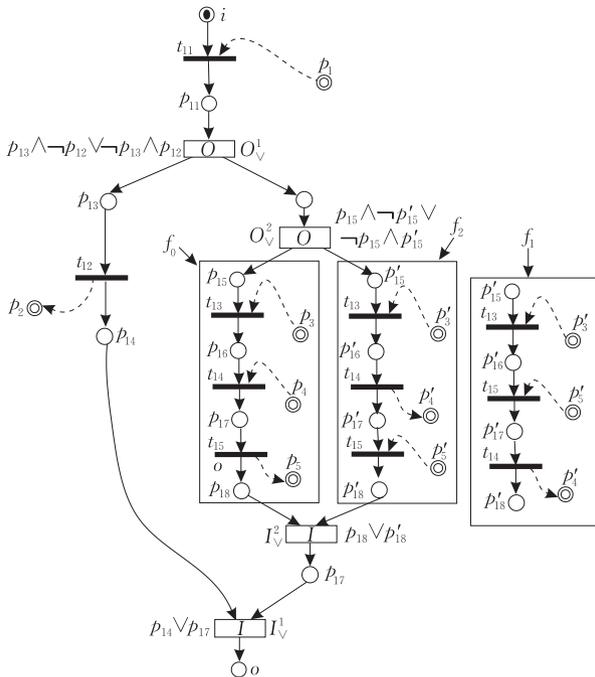


图 8 满足 2NF 的服务网

由于 $L(CFN_3)$ 的割集中存在两段相同的路径表达式序列 $t_{13} t_{14} t_{15}$, 因此 CFN_3 存在等价服务流程, CFN_3 的结构范式不满足第 3 层次, 仅仅满足第 2 层次。

5.4 满足 1NF 的服务网算例

若将图 6 中的服务流程做如下调整: 买方要求先收取商品再付款, 即将变迁 t_{25} 和 t_{26} 的顺序进行颠倒, 调整后的服务网如图 9 所示, 其对应服务网以及控制流网分别命名为 SN_4 以及 CFN_4 , 则 $L(CFN_4) = O_\Lambda^1 (t_{11} O_V^1 (t_{12} \parallel t_{31} t_{32} t_{33}) \otimes (t_{13} t_{14} t_{15}) I_V^1) \parallel (t_{21} t_{22} O_V^2 t_{23} \otimes (t_{24} t_{26} t_{25}) I_V^2) I_\Lambda^1$. 展开 $L(CFN_4)$ 可以得到路径集 C_{path} , 且 $Num(C_{path}) = 5$, 所有的路径分别为 $path1 = O_\Lambda^1 t_{11} O_V^1 t_{12} I_V^1 I_\Lambda^1$, $path2 = O_\Lambda^1 t_{11} O_V^1 t_{31} t_{32} t_{33} I_V^1 I_\Lambda^1$, $path3 = O_\Lambda^1 t_{11} O_V^1 t_{13} t_{14} t_{15} I_V^1 I_\Lambda^1$, $path4 = O_\Lambda^1 t_{21} t_{22} O_V^2 t_{24} t_{26} t_{25} I_V^2 I_\Lambda^1$ 以及 $path5 = O_\Lambda^1 t_{21} t_{22} O_V^2 t_{23} I_V^2 I_\Lambda^1$.

通过判断可知, CFN_4 是良构的, 但路径 $path3$ 和 $path4$ 中内部库所 p_4 和 p_5 存在 $r(p_4) <_{C_3} s(p_5)$

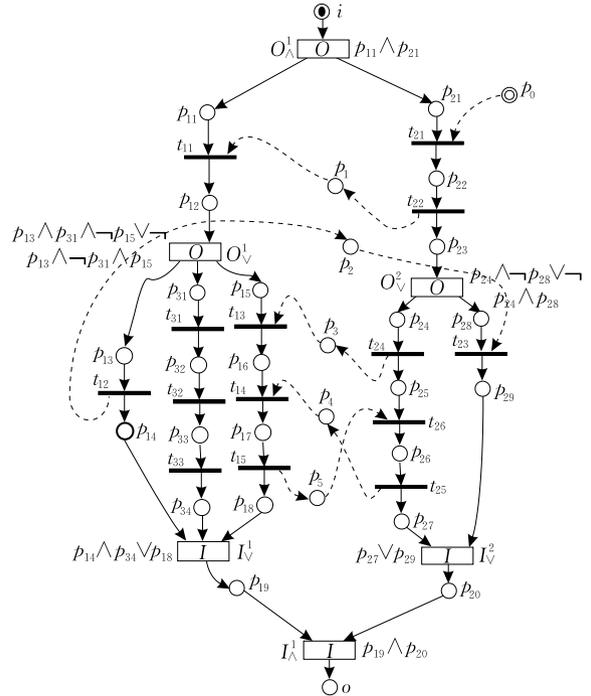


图 9 满足 1NF 的服务网

且 $r(p_5) <_{C_4} s(p_4)$, 因此存在环路, $path3$ 和 $path4$ 为不可达路径, 需要删除, 最终 $Num(C_{path}) = 3$, 因此 CFN_4 的结构范式为 1.

6 结束语

已有研究中仅以路径的可达性作为对服务流程健壮性进行判定的方法不能科学和全面地评价组合服务流程的结构合理性. 为此, 本文提出一种从路径的可达性和结构冗余性等角度对服务流程的结构健壮性进行了层次划分的准则, 并将其定义为服务流程的结构范式, 用于对流程的健壮性以及服务组合的合理性进行定量评价。

文中采用逻辑 Petri 网作为形式化工具, 将服务流程建模为服务网. 为了降低模型分析难度, 将服务网中的数据流与控制流分开, 构建控制流网. 基于逻辑 Petri 网在建模过程中表现出的良好结构性, 提出一种利用约简的方式求解控制流网语言的方法, 进而通过控制流网语言的结构特征判定服务网的结构范式所属层次. 针对提出的四层结构范式, 文中给出了相关实例. 下一步的研究工作中将进一步细化结构范式的粒度, 开发服务流程结构范式自动判定工具。

致 谢 感谢审稿专家及责任编辑对本文的改进提出宝贵意见!

参 考 文 献

- [1] 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
- [2] Liu G J, Jiang C J, Zhou M C, et al. Interactive Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2013, 43(2): 291-302
- [3] Tan W, Fan Y, Zhou M C, et al. Data-driven service composition in enterprise SOA solutions: A Petri net approach. *IEEE Transactions on Automation Science and Engineering*, 2010, 7(3): 686-694
- [4] Vidal J C, Lama M, Bugarin A. Toward the use of Petri nets for the formalization of OWL-S choreographies. *Knowledge and Information Systems*, 2012, 32(3): 629-665
- [5] Lin Chuang, Wan Jian-Xiong, Xiang Xu-Dong, et al. Dynamic optimization in computer systems and computer networks: Models, solutions, and applications. *Chinese Journal of Computers*, 2012, 35(7): 1339-1357(in Chinese)
(林闯, 万剑雄, 向旭东等. 计算机系统与计算机网络中的动态优化: 模型、求解与应用. *计算机学报*, 2012, 35(7): 1339-1357)
- [6] Men Peng, Duan Zhen-Hua. Modeling and evaluation of composite Web services based on generalized stochastic Petri net. *Journal of Xi'an Jiaotong University*, 2008, 42(8): 967-971(in Chinese)
(门鹏, 段振华. 广义随机 Petri 网下的组合 Web 服务建模与评价. *西安交通大学学报*, 2008, 42(8): 967-971)
- [7] Cambroner M E, Diaz G, Valero V, et al. Validation and verification of Web services choreographies by using timed automata. *The Journal of Logic and Algebraic Programming*, 2011, 80(1): 25-49
- [8] Lomuscio A, Qu H, Solanki M. Towards verifying contract regulated service composition. *Autonomous Agents and Multi-Agent Systems*, 2012, 24(3): 345-373
- [9] Wang Y. Research on Web service composition algorithm using description logic. *TELKOMNIKA Indonesian Journal of Electrical Engineering*, 2014, 12(1): 852-858
- [10] Salaun G, Bordeaux L, Schaerf M. Describing and reasoning on Web services using process algebra. *International Journal of Business Process Integration and Management*, 2006, 1(2): 116-128
- [11] Pathak J, Basu S, Honavar V. Composing Web services through automatic reformulation of service specifications// *Proceedings of the 2008 IEEE International Conference on Services Computing*. Hawaii, USA, 2008: 361-369
- [12] Du Y Y, Guo B Q. Logic Petri nets and equivalency. *Information Technology Journal*, 2009, 8(1): 95-100
- [13] Hu Qiang, Du Yuyue, Yu Shuxia. Service net algebra based on logic Petri nets. *Information Sciences*, 2014, 268(1): 271-289
- [14] Benatallah B, Casati F, Toumani F. Representing, analyzing and managing Web service protocols. *Data & Knowledge Engineering*, 2006, 58(3): 327-357
- [15] Xiong P C, Pu C, Zhou M C. Protocol-level service composition mismatches: A Petri net siphon based solution. *International Journal of Web Services Research*, 2010, 7(4): 1-20
- [16] Tan Zhangxi, Lin Chuang, Yin Hao, et al. Approximate performance analysis of Web services flow using stochastic Petri net// *Proceedings of the 3rd International Conference on Grid and Cooperative Computing*. Wuhan, China, 2004: 193-200
- [17] Mansour H E, Dillon T. Dependability and rollback recovery for composite Web services. *IEEE Transactions on Services Computing*, 2011, 4(4): 328-339
- [18] Mateescu R, Poizat P, Salaun G. Adaptation of service protocols using process algebra and on-the-fly reduction techniques. *IEEE Transactions on Software Engineering*, 2012, 38(4): 755-777
- [19] Deng Shui-Guang, Li Ying, Wu Jian, et al. Determination and computation of behavioral compatibility for Web services. *Journal of Software*, 2007, 18(12): 3001-3014(in Chinese)
(邓水光, 李莹, 吴健等. Web 服务行为的兼容性判定与计算问题. *软件学报*, 2007, 18(12): 3001-3014)
- [20] Li Xi-Tong, Fan Yu-Shun. Analyzing compatibility and similarity of Web service processes. *Chinese Journal of Computers*, 2009, 32(12): 2429-2437(in Chinese)
(李喜彤, 范玉顺. Web 服务流程相容性和相似性分析. *计算机学报*, 2009, 32(12): 2429-2437)
- [21] Song Wei, Tang Jin-Hui, Zhang Gong-Xuan, Ma Xiao-Xing. Substitutability analysis of WS-BPEL services. *Science China: Information Science*, 2012, 42(3): 264-279(in Chinese)
(宋巍, 唐金辉, 张功萱, 马晓星. WS-BPEL 服务可替换性分析. *中国科学: 信息科学*, 2012, 42(3): 264-279)
- [22] Lei Li-Hui, Duan Zhen-Hua. An extended deterministic fine automata based method for the verification of composite Web service. *Journal of Software*, 2007, 18(12): 2980-2990(in Chinese)
(雷丽晖, 段振华. 一种基于扩展有限自动机验证组合 Web 服务的方法. *软件学报*, 2007, 18(12): 2980-2990)
- [23] Clempner J. Verifying soundness of business processes: A decision process Petri nets approach. *Expert Systems with Applications*, 2014, 41(11): 5030-5040
- [24] Du Y, Li X, Xiong P C. A Petri net approach to mediation-aided composition of Web services. *IEEE Transactions on Automation Science and Engineering*, 2012, 9(2): 429-435
- [25] Martens A. On compatibility of Web services. *Petri Net Newsletter*, 2003, 65(1): 12-20

- [26] Groefsema H, Bucur D. A survey of formal business process verification: from soundness to variability//Proceedings of the International Symposium on Business Modeling and Software Design. Noordwijkerhout, Netherlands, 2013; 328-341
- [27] 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
- [28] Du Y Y, Qi L, Zhou M C. A vector matching method for analysing logic Petri nets. *Enterprise Information Systems*, 2011, 5(4): 449-468
- [29] Du Y Y, Jiang C J, Zhou M C. A Petri net-based model for verification of obligations and accountability in cooperative

systems. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 2009, 39(2): 299-308

- [30] Du Y Y, Jiang C J, Zhou M C. Modeling and analysis of real-time cooperative systems using Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 2007, 37(5): 643-654
- [31] Du Y Y, Jiang C J, Zhou M C, et al. Modeling and monitoring of E-commerce workflows. *Information Sciences*, 2009, 179(7): 995-1006
- [32] Du Y Y, Jiang C J, Zhou M C. A Petri-net-based correctness analysis of Internet stock trading systems. *IEEE Transactions on Systems, Man, and Cybernetics: Part C*, 2008, 38(1): 93-99



HU Qiang, born in 1980, Ph. D., lecturer. His main research interests include formal modeling by Petri net and service computing.

DU Jun-Wei, born in 1974, Ph. D., associate professor, Ph.D. supervisor. His main research interest is the formal modeling and verification technology of software.

DU Yu-Yue, born in 1960, Ph. D., professor, Ph. D. supervisor. His main research interests include computer supported cooperative work, formal engineering, Petri nets, services computing, real-time systems, Web services, and workflows.

Background

Service composition is an efficient way to obtain the value-added Web services and it is a hot research topic in the domain of service computing. A service process is generated while performing service composition. Recent studies focused on composition methods and the partial structure properties of the service process, such as the interactive compatibility. The studies on these structure properties were based on the premise that the structures of service processes were sound and logically correct. However, not all the structures of service process are reasonable and correct. Some service processes may be not well-structured due to its illogical flow control dependence while others may be deadlock for mismatched data dependence. To study some partial structure properties or to test whether a service process achieved its expected goal, the level of structure soundness of the service process should be evaluated from the overall structure. As far as we know that there are no literatures about the level of structure soundness of the service processes.

Inspired by the concept of normal form in relational algebra, the authors propose the structure normal form of the service processes in this paper. The level of structure soundness of a service process is divided into four levels. To efficiently and succinctly evaluate the structure normal form of a service process, the authors model the service process as

a service net based on logic Petri nets. The evaluation of structure normal form is converted to analyzing the structure properties of the service net. Benefiting from the excellent modeling structure of logic Petri nets, the authors also provides a Petri net language method to study the structure properties of the service net. Thus, the structure normal form is easily evaluated. The authors give the definitions and testing algorithm of the structure normal form in this paper.

This work is supported by the National Basic Research Program of China under Grant No.2010CB328101; the National Natural Science Foundation of China under Grant Nos.61170078, 61273180; the Doctoral Program of Higher Education of the Specialized Research Fund of China under Grant No.20113718110004; the Natural Science Foundation of Shandong Province under Grant No.ZR2012FL17; the Scientific and Technological Program of Qingdao under Grant No.13-1-4-116-jch, and the Supported Program of Research & Innovation Teams in Shandong University of Science and Technology under Grant No.2011KYTD102. The team has published a lot of papers on the study of logic Petri net, service composition and verification of the properties in service processes.