ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.2, February 2010, pp.231−247
doi: 10.3724/SP.J.1001.2010.03790

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于Petri网的服务组合故障诊断与处理*

范贵生[1,2+], 虞慧群[1,2], 陈丽琼[3], 刘冬梅[1]

[1](华东理工大学 计算机科学与工程系,上海 200237)

[2](上海市计算机软件评测重点实验室,上海 201112)

[3](上海应用技术学院 计算机科学与信息工程系,上海 200235)

Fault Diagnosis and Handling for Service Composition Based on Petri Nets

FAN Gui-Sheng[1,2+], YU Hui-Qun[1,2], CHEN Li-Qiong[3], LIU Dong-Mei[1]

[1](Department of Computer Science and Engineering, East China University of Science and Technology, Shanghai 200237, China)

[2](Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai 201112, China)

[3](Department of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai 200235, China)

+ Corresponding author: E-mail: yhq@ecust.edu.cn, http://cs.ecust.edu.cn/~yhq

Fan GS, Yu HQ, Chen LQ, Liu DM. Fault diagnosis and handling for service composition based on Petri nets. *Journal of Software*, 2010,21(2):231−247. http://www.jos.org.cn/1000-9825/3790.htm

Abstract: In this paper, a framework is proposed for handling fault of service composition through analyzing fault requirements. Petri nets are used in the framework for fault detecting and its handling, which focuses on targeting the failure of available services, component failure and network failure. The corresponding fault models are given. Based on the model, the correctness criterion of fault handling is given to analyze fault handling model, and its correctness is proven. Finally, CTL (computational tree logic) is used to specify the related properties and enforcement algorithm of fault analysis. The simulation results show that this method can ensure the reliability and consistency of service composition.

Key words: Petri net; service composition; fault hanlding; CTL (computational tree logic); reliability

摘 要: 通过分析服务组合的故障需求,给出服务组合故障处理的框架.该框架采用 Petri 网来解决服务组合的错误发现及其处理问题.重点讨论了可用服务失败、组件失败及网络故障的情况,并相应地给出了服务组合故障模型.在此基础上对故障处理模型进行分析,给出服务组合故障处理正确性准则,并证明了其正确性.最后,采用 CTL (computational tree logic)描述相关性质并提出验证服务组合故障分析的实施算法.仿真结果表明,该方法在处理服务组合故障时具有一定的优越性.

关键词: Petri 网;服务组合;故障处理;CTL (computational tree logic);可靠性

* Supported by the National Natural Science Foundation of China under Grant Nos.60473055, 60773094 (国家自然科学基金); the National Key Technology R&D Program of China under Grant No.009BAH46B03 (国家科技支撑计划); the Shanghai Shuguang Program of China under Grant No.07SG32 (上海市曙光计划); the Fund of Key Laboratory of Shanghai Science and Technology of China under Grant No.09DZ2272600 (上海市科委重点实验室基金); the Open Research Foundation of Shanghai Institute of Technology of China under Grant No.YJ2009-17 (上海应用技术学院引进人才科研启动项目)

Received 2009-06-15; Revised 2009-09-11; Accepted 2009-12-07

中图法分类号: TP311 文献标识码: A

面向服务的体系架构(service oriented architecture,简称SOA)是一种新兴及蓬勃发展的软件架构理念,它具有良好的可重用性、松耦合性、平台无关性和互操作性,并使用开放的协议和标准.而Web服务作为目前广泛应用的分布式计算技术,能够统一地封装信息、行为以及业务流程,而无须考虑应用所在的环境.也正是因为这些优良的特性,Web服务技术已成为实现SOA架构的首选方式.随着Web服务技术的日益成熟,越来越多稳定、易用的Web服务已共享在网络上.但单一的服务所能提供的功能有限,为了满足实际业务的需求,人们提出服务组合的概念.服务组合就是通过基本服务之间的相互通信和协作,把相对独立及简单的服务组合成具有新功能的大粒度服务的过程,以产生满足服务请求者需求的、增值的服务[1].

然而,Web服务所处的网络环境是一个异构的、分布式自治和快速变化发展的动态环境.网络环境的异构性、分布式自治等特性决定了服务组合在执行过程中可能会受到通信模式的变化、网络失效、服务拒绝攻击、基础设施失效等问题的影响[2].组件服务的自治性、平台独立性以及自身的更新与升级都可能导致原有服务操作失效,甚至出现服务不可用,这都将影响服务组合的正常运行.本文基于文献[3,4]对服务组件发生故障的情况进行归纳,给出服务组合运行时刻可能出现的故障情况:(1) 可用服务运行失败.服务组合执行环境中的服务运行失败所引起的故障使得服务不可调用.(2) 组件运行故障.该组件所对应的可用服务都运行失败,且这些可用服务均不可重复.(3) 网络发生故障.两个组件之间的网络连接出现中断,使得组件之间无法正常通信.

针对服务组合执行过程中发生故障的情况,研究机构提出了许多不同的处理技术,这些技术包含如何管理服务、服务组合与服务组件之间的对应关系,使服务组合的可靠性得到保证.相关工作主要通过服务组合故障处理协商模型、故障注入技术、Petri网等方法对该问题进行研究.由于现有的服务组合故障处理方法已经不能满足实际的应用需求,文献[5−8]对当前的服务组合协议SOAP,WS-FTM和BPEL等进行扩展,然后利用这些协议的故障处理机制来解决错误问题,以增加服务组合的故障处理能力.该方法的优点是利用现有的协议可以提高服务组合开发效率,但Web服务标准体系本身不断发展变化,并且标准之间存在兼容性问题,使得方法不易推广.因此,故障注入[9−11]已成为解决服务组合故障的一个有效方法.故障注入是计算机系统可靠性评估的重要手段.该方法主要是在计算机系统中加入软件实现的错误检测机制,以达到较高的错误检测覆盖率.文献[9]提出了循序渐进故障注入的自动测试案例生成方法,用于测试和评估面向服务架构中功能类似的可用服务.文献[11]在网络层使用错误植入技术对基于SOAP的服务可依赖性进行评估,并通过实验仿真评估方法的可行性.上述的错误注入方法在解决服务的故障方面具有一定的作用,但是所产生的错误注入方案一般是静态规划,如果在组合过程中出现未规划的故障方案,则整个服务组合的运行就有可能会失败.同时,在组合的过程中需要判断具体的方案,这些方法将不可避免地降低服务组合的性能.因此,本文利用Petri网对服务组合的故障处理进行分析.该方法可以描述服务组合过程中的一些不确定性因素,以解决服务组合故障发现及处理的问题.文献[12]虽然也利用Petri网对服务的行为进行分析,以验证组合过程是否能够得到预期的结果,但该方法主要考虑服务组合的流程结构特点,对系统各状态下故障处理机制并没有涉及,而这对服务组合的可靠性有重要的影响.我们在文献[13]中对服务事务属性及其失效处理机制方面进行研究,建立服务组合的失效处理模型,并提出一种构造可靠服务组合的协调策略及实施方法.该方法主要是基于服务请求者的可接受状态进行服务组合的构建,没有考虑服务组合可能出现的故障情况.为了处理这些复杂并且难以预测的Web服务运行故障,使Web服务的可靠性得到保证,本文给出服务组合的故障处理框架.该框架由服务组合的执行和故障分析两部分组成,其中,执行部分主要根据服务组合的特点和需求提出层次服务组合网(hierarchical service composition net,简称HSCN)模型,并利用HSCN描述服务组合中的基本元素,如服务、组件、连接器等;故障分析部分主要是对服务组合中出现的故障及时地发现、定位和处理,并采用CTL(computational tree logic)进行验证.最后,通过仿真实验的结果对比来说明该方法的可行性和有效性.

本文第 1 节给出服务组合故障处理框架.第 2 节提出服务组合故障网,并构造服务组合的的故障模型.第 3 节分析服务组合故障框架的正确性,给出实施算法.第 4 节通过仿真实验说明方法的有效性.最后是结论和下一

步的工作.

1 服务组合的故障处理框架

1.1 服务组合的故障处理需求

由于服务组合的功能由多个独立运行的子功能构成,本文将每个子功能下,组件间的通信过程分别称为组件和连接器.在服务组合中,每个组件会有多个可用服务与之对应,组件和组件之间通过网络进行通信.将服务组合中的可用服务、组件、连接器统称为服务组合的元素.假设同一时刻,系统中只有唯一的元素发生故障.

根据 Web 服务自身的特点,文献[14]提出的模型指明了服务事务属性的语义,该模型是基于文献[15]所考虑的 3 种不同的类型的事务属性,由此可以延伸 Web 服务执行组件时主要有下面 3 种特性:可重复的(r)、可补偿的(cp)、不可补偿也不可重复的(p).相应的服务事务属性可以有以下几种情况:$\{cp\},\{p\},\{r\},\{r,cp\}$.

定义 1. 服务组合故障处理的需求模型是一个七元组,即$\Xi=\{C,WS,CT,RL,TW,RT,RC\}$:(1) C,WS,CT分别是有限的组件集、可用服务集和连接器集.(2) $RL:C\times C\to\{>,+,\|,n\}$是组件间的关系函数,其中,$>,+,\|,n$分别表示顺序、选择、并行和循环关系;若组件$C_i$和$C_j$的关系为顺序,则称$C_i$为$C_j$的前向组件,$C_j$为$C_i$的后向组件.记集合$Fork(C_i),Back(C_i)$分别为组件$C_i$的前向、后向组件集.(3) $TW:C\to WS^*$是组件的可用服务函数,$TW(C_i)=WS_i=\{WS_{i,1},WS_{i,2},\ldots,WS_{i,m}\}$表示组件$C_i$的可用服务集,其中,服务$WS_{i,j}$表示组件$C_i$的第$j$个可用服务;(4) $RT:WS\to R\times TP$是服务的QoS函数,其中,R是指(0,1)区间的实数集,$TP=\{r,p,cp,\{r,cp\}\}$是服务的事务属性.$RT(WS_{i,j})=(SP_{i,j},TP_{i,j}),SP_{i,j},TP_{i,j}$分别表示服务$WS_{i,j}$的成功概率和事务属性.(5) $RC:CT\to(0,1)$是连接器的成功概率,根据实际情况可知,连接器的成功概率不为 1.

例 1:本文用一个正在开发的国际贸易出口服务来说明服务组合的需求模型.出口服务流程首先根据产品的相关需求来查询信息和选择目的地(C_1),并通过包装服务(C_2)负责对出口的产品进行加工和包装.产品包装完成后,并行执行出口运输方式预订、商检局报检(C_3)、保险办理(C_4),其中,出口运输方式预订业务包括订购水运(C_5)和航运(C_6)等.该业务将对应服务执行的结果(班次、出发到达时间、价格、付款帐户等)反馈给出口商,商检局报检服务则反馈相应的检验证书给出口商,而保险办理服务则为出口商品办理保险并取得保险单.所有信息出口商确认后,将信息送到监管部门检查(C_7),如海关的出口放行检查、检验检疫、税收检查.产品信息检验合格后,货物装运并得到提货单.最后通过财务服务(C_8)对产品的相关票据进行核销,并将退税信息反馈给出口商.该业务流程可表示为表达式$C_1>C_2>(C_3\|C_4\|(C_5+C_6))>C_7>C_8$,具体可用服务及其属性见表 1.设出口服务中连接器的成功概率均为 95%.

Table 1 Available service and its attributes

表 1 可用服务及其属性

WS	RT		WS	RT		WS	RT		WS	RT	
	SP (%)	TP		SP (%)	TP		SP (%)	TP		SP (%)	TP
$WS_{1,1}$	92.56	r	$WS_{2,1}$	96.44	cp	$WS_{3,1}$	89.63	p	$WS_{4,1}$	92.47	r,cp
$WS_{1,2}$	97.74	cp	$WS_{2,2}$	83.95	p	$WS_{3,2}$	97.51	cp	$WS_{4,2}$	80	p
$WS_{1,3}$	96.44	r	$WS_{2,3}$	77.18	p	$WS_{3,3}$	70	r,cp	$WS_{4,3}$	97.93	cp
$WS_{1,4}$	99.76	r,cp	$WS_{2,4}$	83.36	cp	$WS_{3,4}$	65.79	p	$WS_{4,4}$	78.70	r
$WS_{5,1}$	97.59	p	$WS_{2,5}$	67.18	cp	$WS_{3,5}$	84.24	p	$WS_{4,5}$	94.05	cp
$WS_{5,2}$	88.92	p	$WS_{2,6}$	55.36	p	$WS_{7,1}$	96.73	cp	$WS_{4,6}$	81.36	r
$WS_{5,3}$	75.13	cp	$WS_{6,1}$	87.32	cp	$WS_{7,2}$	69.89	r	$WS_{8,1}$	83.81	p
$WS_{5,4}$	57.90	p	$WS_{6,2}$	60	p	$WS_{7,3}$	93.38	p	$WS_{8,2}$	59.29	p
$WS_{5,5}$	95.56	r	$WS_{6,3}$	94.24	r	$WS_{7,4}$	89.15	r,cp	$WS_{8,3}$	92.68	r
$WS_{5,6}$	92.31	r,cp	$WS_{6,4}$	44.73	r,cp	$WS_{7,5}$	97.72	p	$WS_{8,4}$	88.77	cp
$WS_{5,7}$	66.07	cp	$WS_{6,5}$	89.80	cp	$WS_{7,6}$	88.31	r	$WS_{8,5}$	97.49	r,cp

1.2 服务组合故障处理框架

服务组合故障处理框架的具体执行流程如图 1 所示,该方法采用 Petri 网对组合过程进行描述,并借助 Petri 网的相关技术和分支时态逻辑对服务组合的故障处理过程进行分析和验证.整个处理过程主要分为两个阶段:

服务组合的执行阶段和故障处理分析阶段.

第 1 阶段为服务组合的执行阶段.该阶段采用高级 Petri 网作为服务组合的描述工具,根据服务组合的特点和需求对服务组合的执行过程进行建模,最后获得服务组合的故障模型.

第 2 阶段为服务组合的故障处理分析阶段.本阶段所涉及的内容包括故障诊断和故障处理,其中:

(1) 故障诊断是指当服务组合发生故障时,系统可以及时觉察(故障发现)并判断故障类型及其具体位置(故障定位).具体位置是指哪个元素发生故障.

(2) 故障处理:若可用服务失败且该服务是可重复的,则重新运行该服务,否则对该组件重新绑定;若组件运行失败(该组件不存在可重复的服务,且所有可用服务都运行失败),则向系统抛出故障;若连接器发生故障,则重试发送消息,但每个消息最多只能发送 3 次(可以根据实际需求进行定义).若消息包发送 3 次都失败,则向系统抛出故障.若系统收到故障,则对任意的可用服务$WS_{i,j}$采取如下处理措施:① 服务$WS_{i,j}$没有得到调用或已经运行失败,则不做任何操作;② 服务$WS_{i,j}$处在运行位置或等待重试,则取消$WS_{i,j}$的运行;③ 服务$WS_{i,j}$处在结束位置且是可补偿的,则对$WS_{i,j}$进行补偿;④ 服务$WS_{i,j}$处在结束位置且不可补偿,则不做任何操作,服务$WS_{i,j}$停留在结束位置.

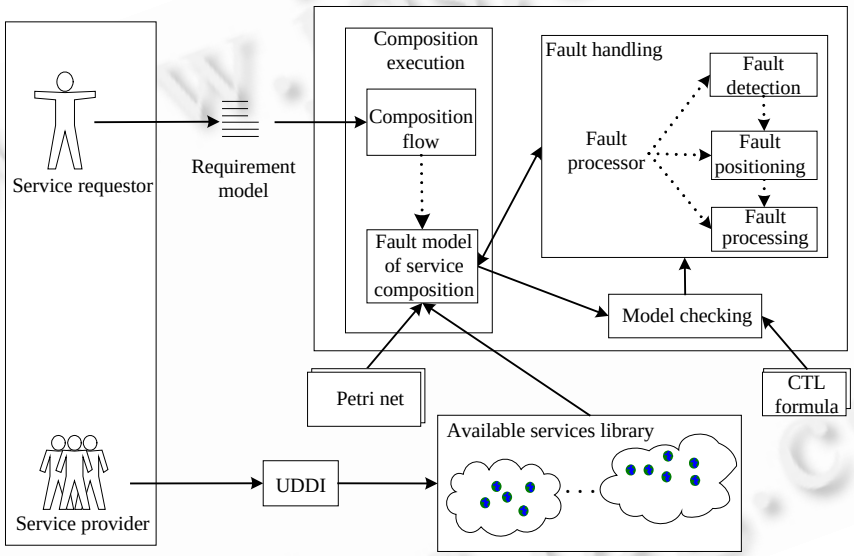


Fig.1 Fault handling framework of service composition

图 1 服务组合故障处理框架

2 服务组合的执行

下面根据服务组合的故障需求提出HSCN模型,并用它对服务组合中的服务、组件、连接器以及整个组合流程进行描述.为了区别变迁和库所所属的服务、组件、连接器,在变迁和库所前标注对应的部分.如服务$WS_{i,j}$中的开始变迁t_{in}表示为$WS_{i,j}.t_{in}$.如果变迁是为了描述数据流程而引入,则不标注.

2.1 层次服务组合网

Petri 网作为一种图形化建模工具和一种具有丰富数学基础的形式化模型,可以广泛应用于描述和研究并发、异步和分布式特征的系统.因此,Petri 非常适合描述 Web 服务这种松耦合的分布式系统.下面介绍一些系统运行的基本概念,其余的概念见文献[16].

定义 2. 七元组$\Sigma=(N,IO,D,A_T,A_F,\lambda,M_0)$称作基本服务网(basic servic net,简称BSN),其中:

(1) $N=(P,T,F)$是一个基本 Petri 网,其中,P,T,F 分别表示库所、变迁和弧的有限集,且两两不相交.

(2) $IO\subset P$ 是一类特殊的库所,称为Σ的接口,用虚线圆圈表示.

(3) D为非空有限的个体集,规定f_D,f_S分别为D上给定的谓词集和符号和集.

(4) $A_T:T\to f_D$,对于$t\in T$,$A_T(t)$中的自由变量必须是以t为一端的有向弧上自由变量.

(5) $A_F:F\to f_S$,若$(p,t)\in F$或$(t,p)\in F$,则$A_F(p,t)$或$A_F(t,p)$为n元符号和,缺省为空.

(6) $\lambda:T\to(0,1)\times N^*$是变迁的属性函数,$\forall t_i\in T,\lambda(t_i)=(\alpha_i,\beta_i)$,其中,$\alpha_i,\beta_i$分别描述$t_i$的触发概率和优先级,默认值为(1,0).本文假设$\beta_i$越小,变迁$t_i$的优先级越大.

(7) $M_0:P\to f_S$是Σ的初始标识,$p\in P$,$M_0(p)$不含任何自由变量.

BSN模型主要是对可用服务执行过程进行建模,其中,库所、变迁和接口分别描述可用服务的所处位置、可能操作和输入输出参数.对任意$x\in(P\cup T)$,集合${}^\bullet x=\{y|y\in(P\cup T)\wedge(y,x)\in F\}$和$x^\bullet=\{y|y\in(P\cup T)\wedge(x,y)\in F\}$分别对应于$x$的输入和输出;对任意$t_i\in T$,将变迁$t_i$的输入/输出弧以及$A_T(t_i)$中出现的自由变量集记作$FV(t_i)$.

定义 3. 六元组$\Omega=\{\Sigma,\Gamma,TI,TA,PI,PA\}$称为层次服务组合网(hierarchical service composition net,简称 HSCN),其中:

(1) Σ是一个 BSN 模型,描述了Ω的基本结构.

(2) $\Gamma=\{\Gamma_i|i\in N^*\}$是页的有限集,每个页面是一个除它本身以外的BSN或HSCN模型,页和页之间互不相交:设Γ_i对应的Petri网结构$N_i=(P_i,T_i,F_i)$,则$\forall\Gamma_i,\Gamma_j\in\Gamma$,有$(P_i\cup T_i\cup F_i)\cap(P_j\cup T_j\cup F_j)=\varnothing$.

(3) $TI\subset T$ 是替代节点的集合,其中每个页面对应一个替代节点.

(4) $TA:TI\to\Gamma$是页的分配函数,即为每个替代节点分配具体页面.

(5) $PI\subset P$ 是端口节点的集合,端口节点描述替代节点的输入和输出库所.

(6) PA 是端口映射函数,其功能是把替代节点的端口节点映射到对应页面的输入和输出接口,且每个端口节点只能映射到一个接口.

BSN模型是HSCN的一个特例,即Γ为空的HSCN模型.HSCN模型主要是对组件或整个服务组合流程进行建模.个体集D主要用来描述服务组合中的资源,如可用服务集和数据包.本文将可用服务$WS_{i,j}$抽象成个体$d_{i,j}=(j,SP_{i,j},TP_{i,j})$,其中,$j$表示服务$WS_{i,j}$在可用服务集$WS_i$中所处位置,$SP_{i,j},TP_{i,j}$分别表示服务的成功概率和事务属性.而把服务组合中数据包统一抽象成个体φ.如无特殊说明,HSCN模型中出现的个体均为φ.为了对模型中的符号进行区分,本文用大写字母标注端口节点和替代变迁,其余库所和变迁均用小写字母标注;另外,输入接口用上标I标示,输出接口用上标O标示.

例 1 中组件C_1的HSCN模型Ω_1如图 2 所示,个体集$D=\{(1,92.56,r),(2,97.74,cp),(3,96.44,r),(4,99.76,r,cp),\varphi\}$描述了组件$C_1$的资源,如个体$d_{1,2}=(2,97.74,cp)$描述了可用服务$WS_{1,2}$.$\Omega_1$中的替代节点$WS$分配给服务页面(见图 3),相应的端口节点$P_{in},P_{ou},P_{tr},P_{fa}$分别映射为服务页面的输入和输出接口 $p_{in}^I,p_{ou}^I,p_{tr}^I,p_{fa}^I$.$A_T(t_f)$为$M(p_w)=\varnothing$,即C_1的所有可用服务都运行失败且均不可重复.初始标识为$M_0(p_w)=\{(1,92.56,r),(2,97.74,cp),(3,96.44,r),(4,99.76,\{r,cp\})\}$.另外,设置变迁$t_{ab},t_{in},t_{eh},t_f$的优先级为 0,其余变迁优先级为 1.

在某时刻各库所中个体的分布状况称为HSCN模型的标识,记作$M.M:P\to f_S$主要描述了系统的资源分布.如M_1是Ω_1的一个标识,其中,$M_1(p_i^I)=M_1(p_t^I)=\varphi,M_1(p_w)=M_0(p_w)$.下面根据HSCN模型的标识给出可行替换、变迁发生权、有效触发、触发规则的形式化定义.

$\forall t\in T$,若$FV(t_i)=\{x_1,x_2,\ldots,x_n\}$,个体集$\{d_1,d_2,\ldots,d_n\}$满足$d_i\in\{M(p)|p\in{}^\bullet t\cup t^\bullet\}$且$d_i$对应于变量$x_i$,则将个体$d_1,d_2,\ldots,d_n$分别替换$x_1,x_2,\ldots,x_n$所得到的变迁$t$的实例称为变迁$t$的一个替换,记作$t\langle x_1\leftarrow d_1,x_2\leftarrow d_2,\ldots,x_n\leftarrow d_n\rangle$,简写为$t\langle d_1,d_2,\ldots,d_n\rangle$.替换主要是对$t$的输入/输出弧以及$A_T(t)$中出现的所有自由变量绑定相应的个体.记$A_T(t)\langle d_1,d_2,\ldots,d_n\rangle$和$A_F(p,t)\langle d_1,d_2,\ldots,d_n\rangle$分别描述将个体$d_1,d_2,\ldots,d_n$替换到公式$A_T(t)$和输入弧上谓词$A_F(p,t)$所得到的值.若替换$t\langle d_1,d_2,\ldots,d_n\rangle$使得$A_T(t)\langle d_1,d_2,\ldots,d_n\rangle$=true,则称$t\langle d_1,d_2,\ldots,d_n\rangle$是标识$M$下变迁$t$的可行替换.记变迁$t$在标识$M$下的所有可行替换集合为$VP(M,t)$.设变迁$t$的自由变量集$FV(t_i)$为空,若$\forall p\in{}^\bullet t,M(p)\neq\varnothing$,则变迁$t$在标识$M$下肯定存在可行替换.对应可行替换的执行过程为从$t$的每个输入库所$p_i$中任选一个个体$d_i$即可.如$\Omega_1$模型中,替换$t_{in}\langle(1,92.56,r)\rangle$和$t_{in}\langle(2,97.74,cp)\rangle$均为变迁$t_{in}$在标识$M_1$下的可行替换.

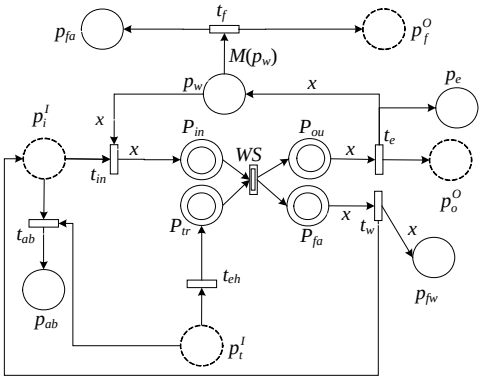


Fig.2 HSCN model of component C_1

图 2 组件C_1的HSCN模型

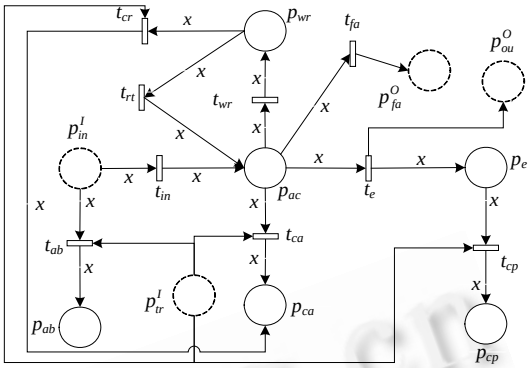


Fig.3 Fault model of service $WS_{i,j}$

图 3 服务$WS_{i,j}$的故障模型

定义 4. 设Ω为 HSCN 模型,M 为Ω的一个标识,若 $VP(M,t)\neq\varnothing$,则称变迁 t 在标识 M 下有发生权,记作 $M[t\rangle$.

在HSCN模型中,变迁t在标识M下是有发生权的当且仅当t在标识M下存在可行替换.将标识M下所有有发生权的变迁集合记为$ET(M)$.如变迁t_{in}在标识M_1下存在可行替换$t_{in}\langle(1,92.56,r)\rangle$,则变迁$t_{in}$在标识$M_1$下是有发生权的.

定义 5. 设Ω为HSCN模型,M为Ω的一个标识.对于变迁$t_i\in ET(M)$,如果t_i满足下列条件:$\beta_i\leq\min(\beta_j)$,其中,$t_j\in ET(M)$,则称变迁t_i在标识M下的触发是有效的.将标识M下所有可以有效触发的变迁集合记为$FT(M)$.

标识M下t_i的触发是有效的当且仅当t_i在标识M下有发生权,且变迁集$ET(M)$中不存在比t_i优先级高的变迁.在标识M下,通过有效触发变迁t_i的一个可行替换$t_i\langle d_1,d_2,\ldots,d_n\rangle$到达新标识$M'$的过程记为$M[t_i\langle d_1,d_2,\ldots, d_n\rangle\rangle M'$.如$\Omega_1$中,变迁$t_{ab},t_{in}$在标识$M_1$下均有发生权,但是由于$t_{ab}$的优先级高于$t_{in}$,所以$t_{ab}$的触发是有效的.HSCN模型同样存在Petri网状态空间爆炸问题,为此,本文引入最大触发集的概念.

定义 6. 设Ω为HSCN模型,M为Ω的一个标识.对于$\forall t_i,t_j\in FT(M)$,若${}^\bullet t_i\cap{}^\bullet t_j=\varnothing$,则称$t_i,t_j$在标识$M$下并发,否则称变迁$t_i,t_j$在$M$下冲突.

若两个变迁间存在并发关系,则无论触发哪个变迁都不会影响另一个变迁的触发.将M下并发的变迁集合称为M的最大并发集,记为$MT(M)$.集合$H(M)=\{t\langle d_1,d_2,\ldots,d_n\rangle|t\in MT(M),t\langle d_1,d_2,\ldots,d_n\rangle\in VP(M,t)\}$称为$M$的一个最大触发集.

定义 7. 设Ω为 HSCN 模型,M 为Ω的一个标识,系统通过有效触发 $H(M)$中所有变迁到达新的标识 M',记作 $M[H(M)\rangle M'$,称 M'是 M 的可达标识.M'分别按如下规则计算:

$$\forall t_i\langle d_1,d_2,\ldots,d_n\rangle\in H(M),\forall p_j\in{}^\bullet t_i\cup t_i^\bullet: M'(p_j)=M(p_j)-A_F(p_j,t_i)\langle d_1,d_2,\ldots,d_n\rangle+A_F(t_i,p_j)\langle d_1,d_2,\ldots,d_n\rangle.$$

在HSCN模型中,可达标识按照触发变迁的可行替换进行相应的调整.如果存在触发序列$H_1,H_2,\ldots,H_k$和标识序列$M_1,M_2,\ldots,M_k$使得$M[H_1\rangle M_1[H_2\rangle M_2\ldots M_{k-1}[H_k\rangle M_k$,则称$M_k$是从$M$可达的.从$M$可达的所有标识集合记为$R(M)$,规定$M\in R(M)$.基于上述触发规则,从初始标识$M_0$出发构造故障模型的可达图,其中,节点为可达标识,边上标记触发的变迁及其触发概率.

2.2 服务组合的故障模型

层次服务组合网 HSCN 可以抽象为具有层次结构的网模型,其中,最高层的模型主要是依据服务请求者需求生成整个应用所对应的流程,第 2 层定义了各个组件的故障模型,最底层是可用服务执行过程所对应的具体页面.下面依次对服务组合的基本元素进行建模,然后利用 HSCN 的层次性构造整个服务组合的故障模型.

可用服务的故障模型如图 3 所示,设组件C_i调用了服务$WS_{i,j}$(即x等于$WS_{i,j}$对应的个体$d_{i,j}$),则$WS_{i,j}$的具体运行流程为:

(1) 若服务$WS_{i,j}$初始化t_{in}之前需要对其进行故障处理(库所 p_{tr}^{I} 中有令牌),则调用中断操作t_{ab}使服务$WS_{i,j}$进入中断位置p_{ab}.

(2) 否则,初始化t_{in}使得服务$WS_{i,j}$进入运行位置p_{ac}.

(3) 若运行过程中(即$M(p_{ac})\neq\varnothing$)需要故障处理,则调用取消操作t_{ca},使服务$WS_{i,j}$进入取消位置p_{ca}.

(4) 若服务运行失败且服务是可重复的,则调用重试操作t_{wr}使服务进入等待重试位置p_{wr};若服务运行失败且服务是不可重复的,则调用失败操作t_{fa}输出失败信息到库所 p_{fa}^{O}.

(5) 若服务处于重试位置p_{wr}时需要故障处理,则调用取消重试操作t_{cr},使服务$WS_{i,j}$进入取消位置p_{ca}.

(6) 若服务$WS_{i,j}$运行成功t_e,则进入结束位置p_e并输出执行结果 p_{ou}^{O}.

(7) 若服务处于结束位置p_e时需要故障处理,且服务$WS_{i,j}$是可补偿的,即$cp\in TP_{i,j}$,则调用补偿操作t_{cp},使服务$WS_{i,j}$进入补偿位置p_{cp};否则,不执行任何操作,服务保持在结束位置p_e.

为了体现服务的QoS函数,在故障模型中设置$\alpha(t_{fa})=\alpha(t_{wr})=1-SP(x),\alpha(t_e)=SP(x)$,变迁$t_{fa},t_{wr},t_{cp}$的$A_T$函数分别为$r\notin TP(x),r\in TP(x),cp\in TP(x)$.设置变迁$t_{ab},t_{ca},t_{cr},t_{cp}$的优先级为 0,其余变迁的优先级为 1.通过对图 3 的分析可知,事务属性不同的服务,运行过程可能到达的位置和执行的操作也不同.服务$WS_{i,j}$故障模型中库所和变迁的实际映射见表 2.

Table 2 Transitions and places in fault model of service

表 2 服务的故障模型中的变迁和库所

P	Position	T	Operation
p_{in}^{I}	Initial position	t_{in}	Running operation
p_{ou}^{O}	Output results	t_{ca}	Cancellation operation
p_{ca}	Cancellation position	t_{wr}	Waiting for retrying
p_{ac}	Running position	t_{cr}	Cancelling retrying
p_{wr}	Wating for retrying	t_{cp}	Compensation operation
p_{ab}	Abortion position	t_{ab}	Abortion operation
p_{fa}^{O}	Failure output	t_{fa}	Failure operation
p_{cp}	Compensation position	t_{rt}	Retrying operation
p_{tr}^{I}	Transaction input	t_e	Termination operation
p_e	Termination position		

本文中,组件的故障模型结构都是一样的,只是在初始标识下库所p_w所对应的个体集不同.组件C_i的故障模型如图 2 所示,设组件C_i的可用服务集为WS_i,具体执行流程为:

(1) 组件得到所需的信息 p_i^I ,若此时出现故障(库所 p_t^I 中有令牌),则调用中断操作t_{ab},使组件进入中断位置p_{ab}.

(2) 否则,触发变迁t_{in}来选择库所p_w(p_w映射为组件C_i的可用服务库,在初始标识下,该库所包含WS_i中所有可用服务对应的个体)中的任意个体x作为执行服务并开始整个组件的运行,同时,系统调用对应的服务页面(如图 3 所示)执行.

(3) 若变迁t_{in}触发后系统出现故障,则调用故障处理操作t_{eh}对C_i的执行服务进行故障处理,即传输令牌到库所P_{tr}.

(4) 若该组件不存在可重复的服务且所有可用服务都运行失败,则组件进入运行失败位置p_{fa}同时输出故障 p_f^O .

(5) 若失败的服务是不可重复的且不是最后一个服务,则重新调用剩余可用服务(即调用p_w中剩余的某个个体),并将个体x放入失败服务库p_{fw}(该库所用来存放所有运行失败的可用服务).

(6) 否则,调用结束变迁t_e,输出运行结果φ到库所p_e和 p_o^O .

将组件C_i和C_j的通信过程抽象成一个连接器$C_{i,j}$,对应的故障模型如图 4 所示,具体执行流程为:

(1) 引入变迁t_{st}将需要传输的数据 p_i^I 传送到网络上p_{ac},同时将数据写入缓存p_p.

(2) 若数据传送失败t_{fa},则进入等待重试位置p_{wr},此时,如果还有重试机会(每个消息只能重试 2 次),则将缓存的数据重新进行发送,否则,调用失败处理操作t_f,释放缓存并输出网络故障 p_f^O,同时,连接器进入失败位置p_{fa}.

(3) 若数据发送成功t_e,则释放缓存,输出数据包.

(4) 若连接器运行期间系统出现故障,则调用取消发送操作t_{ca}.其中,$\alpha(t_{fa})=1-RC(C_{i,j})$,$\alpha(t_e)=RC(C_{i,j})$,变迁$t_{ca}$,$t_f$的优先级为 0,其余变迁的优先级均为 1.

服务运行失败的处理过程在相应组件的故障模型中已经得到建模,而组件和连接器运行失败的处理过程所对应的HSCN模型如图 5 所示:$\forall x\in C\cup CT$,引入变迁$t_{eh,x}$将组件的故障输出到库所p_{cn},此时,整个组合流程处于等待故障处理的位置p_{cn};变迁t_{cn}表示组合流程故障处理开始操作,故障处理需要对所有组件和连接器进行操作,设置$^\bullet t_{cn}=p_{cn}$, $t_{cn}^\bullet=\{x.P_t|x\in C\cup CT\}\cup\{p_{eh}\}$,其中,$p_{eh}$表示服务组合的故障位置,设置变迁$t_{eh,i}$,$t_{cn}$的优先级均为 0.

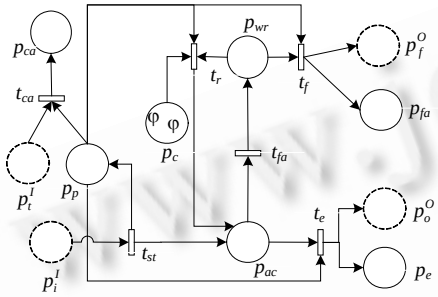


Fig.4 Fault model of connector $C_{i,j}$

图 4 连接器$C_{i,j}$的故障模型

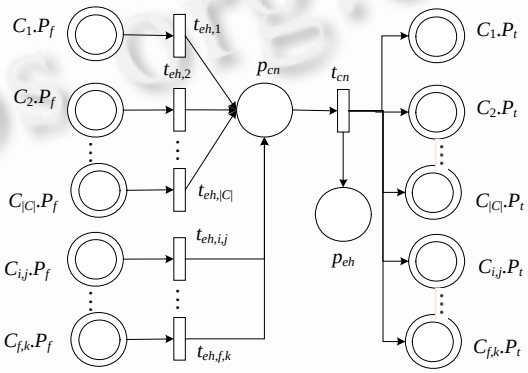


Fig.5 Fault handling model

图 5 故障处理模型

设系统根据每个组件的特性构造了相应的故障模型,下面根据组件间的关系组合组件的故障模型.这里主要考虑组件组合过程的数据流程.为了构造组件间的并行、选择和循环等关系,需要分别对组件间的两对基本组合即AND-split和OR-split,AND-join和OR-join进行建模,具体的故障模型(设组件C_f,C_k分别是C_i,C_j的共同前向和后向组件)如下:

(1) 顺序关系$C_i>C_j$的HSCN模型如图 6(a)所示,引入$t_{s,i,j}$将C_i的输出数据传送给$C_{i,j}$的输入端口,$t_{e,i,j}$将传输好的消息发送到C_j的输入端口.

(2) AND-split的HSCN模型如图 6(b)所示,引入$t_{f,ij}$将组件C_f的结果传送到$C_{f,i}$,$C_{f,j}$的输入接口:

$$^\bullet t_{f,ij}=C_f.P_o,\ t_{f,ij}^\bullet=\{C_{f,i}.P_i,C_{f,j}.P_i\}.$$

(3) AND-joint的HSCN模型如图 6(c)所示,引入$t_{ij,k}$将$C_{f,i}$,$C_{f,j}$的结果传送到组件C_k的输入接口:

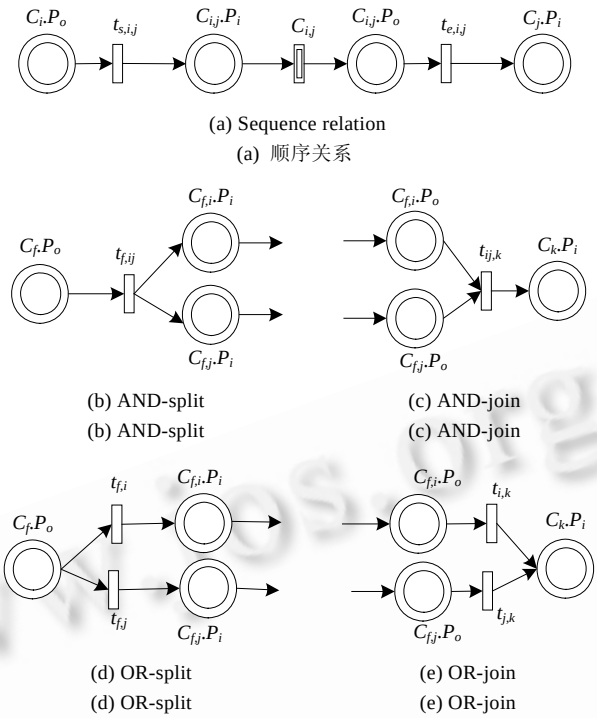
$$^\bullet t_{ij,k}=\{C_{f,i}.P_o,C_{f,i}.P_o\},\ t_{ij,k}^\bullet=C_k.P_i.$$

(4) OR-split的HSCN模型如图 6(d)示,引入$t_{f,i}$,$t_{f,j}$将组件C_f的结果传送到$C_{f,i}$,$C_{f,j}$的输入接口:

$$^\bullet t_{f,i}=^\bullet t_{f,j}=C_f.P_o,\ t_{f,i}^\bullet=C_{f,i}.P_i,\ t_{f,j}^\bullet=C_{f,j}.P_i.$$

(5) OR-joint的HSCN模型如图 6(e)所示,引入$t_{i,k}$,$t_{j,k}$将$C_{f,i}$,$C_{f,j}$的结果传送到C_k的输入接口:

$$t_{i,k}^\bullet=t_{j,k}^\bullet=C_k.P_i,\ ^\bullet t_{i,k}=C_{f,i}.P_o,\ ^\bullet t_{j,k}=C_{f,j}.P_o.$$



(a) Sequence relation

(a) 顺序关系

(b) AND-split (c) AND-join

(b) AND-split (c) AND-join

(d) OR-split (e) OR-join

(d) OR-split (e) OR-join

Fig.6 Fault model of basic composition

图 6 基本组合的故障模型

根据上面各个基本元素的模型,需求模型Ξ的故障模型Ω的构造步骤如下:

(1) 根据服务故障模型的构造方法,结合可用服务的属性函数 RT,构造每个组件所对应的服务故障模型.

(2) 根据组件对应的可用服务形成组件的故障模型,进而构建组件间连接器的故障模型.

(3) 引入变迁t_{st}和库所p_{st}来描述整个系统的开始操作和开始位置,依据组件的特征对整个系统进行初始化,使得${}^\bullet p_{st}=\varnothing$, $p_{st}^\bullet=t_{st}$,${}^\bullet t_{st}=p_{st}$, $t_{st}^\bullet=\{C_i.P_i|Forw(C_i)=\varnothing\}$,设置初始标识$M_0(p_{st})=\varphi$.

(4) 引入变迁t_{en}和库所p_{en}来描述整个系统的结束操作和结束位置,使得

$$t_{en}^\bullet=p_{en},\ p_{en}^\bullet=t_{en},\ {}^\bullet t_{en}=\{C_i.P_o|Back(C_i)=\varnothing\},\ p_{en}^\bullet=\varnothing.$$

3 服务组合的故障分析

3.1 框架的正确性分析

服务组合故障处理框架的正确性主要包含[17]:(1) 元素的正确性.服务故障模型的运行符合服务的需求,对于连接器和组件而言,正确性是指故障模型中的每个元素都有可能执行成功.(2) 服务组合的正确性是指整个服务组合在运行过程中能够得到正确的执行.(3) 故障处理的正确性是指若系统发生故障,则整个服务组合可以得到正确的处理.

服务故障模型的正确性是指其运行过程中所到达的位置符合服务的 QoS 需求.根据服务的事务属性,服务运行过程中可到达的位置可能不同,但同一时刻该服务只能处于初始、中断、运行、取消、失败、等待重试、结束和补偿中的一个位置.

性质 1. 在HSCN模型中,$\forall WS_{i,j}\in WS,\forall M\in R(M_0),\exists!p_k\in P$,有$d_{i,j}\in M(p_k)$.

证明:$\forall WS_{i,j}\in WS$,在初始标识M_0中有$d_{i,j}\in M_0(C_i.p_w)$.

$\forall M\in R(M_0)$,则服务$WS_{i,j}$在标识M下有两种情形:

① 若服务$WS_{i,j}$没有被调用,则对应的个体仍然处于初始库所$C_i.p_w$中,设$p_k=C_i.p_w$,命题成立.

② 若服务$WS_{i,j}$被调用,则由服务运行过程的故障模型(如图 3 所示)可知,其满足:

$M(WS_{i,j}.p_{in}^I)\cup M(WS_{i,j}.p_{ac})\cup M(WS_{i,j}.p_{wr})\cup M(WS_{i,j}.p_{ca})\cup M(WS_{i,j}.p_e)\cup M(WS_{i,j}.p_{ab})\cup M(WS_{i,j}.p_{cp})\cup M(WS_{i,j}.p_{fa}^O)=d_{i,j}$.

一旦服务$WS_{i,j}$被调用,就会对其进行初始化(变迁$C_i.t_{in}$),即输入个体$d_{i,j}$到库所$WS_{i,j}.p_{in}$.根据HSCN的运行机制可得$\exists!p_k\in\{WS_{i,j}.p_{in},WS_{i,j}.p_{ac},WS_{i,j}.p_{wr},WS_{i,j}.p_{ca},WS_{i,j}.p_{fa}^O,WS_{i,j}.p_e,WS_{i,j}.p_{ab},WS_{i,j}.p_{cp}\}$,使得$d_{i,j}\in M(p_k)$.若服务运行失败且是不可重复的,则组件将该服务放于运行失败的服务库$C_i.p_{fw}$,即$p_k=C_i.p_{fw}$.

综上所述,在HSCN模型中,$\forall WS_{i,j}\in WS,\forall M\in R(M_0),\exists!p_k\in P$,有$d_{i,j}\in M(p_k)$. □

性质 1 说明,在 HSCN 模型运行过程中,可用服务对应的个体都处于唯一的位置.下面分析连接器故障模型的正确性,即只要系统调用某个连接器来发送消息,该消息就可能发送成功.

性质 2. 在HSCN模型中,$\forall C_{i,j}\in CT$,若$\exists M\in R(M_0),M(C_i.p_o^O)=\varphi$,则$\exists M'\in R(M)$使得$M'(C_j.p_i^I)=\varphi$.

证明:$\forall C_{i,j}\in CT$,若$\exists M\in R(M_0)$,有$M(C_i.p_o^O)=\varphi$,下面根据组件C_i和C_j的关系,分顺序、AND-split和OR-split这 3 种情形来分析:

① 若$C_i>C_j$,则:

因为$^\bullet t_{s,i,j}=C_i.P_o\wedge(\forall t_k\in T,{}^\bullet t_k\cap{}^\bullet t_{s,i,j}=\varnothing)$,

所以$\exists M_1\in R(M)$,使得$t_{s,i,j}\in FT(M_1)$.

设H_1是M_1的最大触发集,令$M_2=M_1[H\rangle$,则有$M_2(C_{i,j}.p_i)=\varphi$;

根据通信组件故障模型的运行机制可知,$\exists M_3\in R(M_2)$,使得$M_3(C_{i,j}.P_o)=\varphi$.

因为$^\bullet t_{e,i,j}=C_{i,j}.P_o\wedge(\forall t_k\in T,{}^\bullet t_k\cap{}^\bullet t_{e,i,j}=\varnothing)$,

所以设H_3是M_3的最大触发集,令$M_4=M_3[H_3\rangle$,则有$M_4(C_j.p_i)=\varphi$.

所以令$M'=M_4$,命题成立.

② 同理可以证明,在情形 AND-split 和 OR-split 下命题成立.

综上所述,$\forall C_{i,j}\in CT$,若$\exists M\in R(M_0),M(C_i.p_o^O)=\varphi$,则$\exists M'\in R(M)$,使得$M'(C_j.p_i^I)=\varphi$. □

性质 2 说明,在HSCN模型中,对于任意一个连接器$C_{i,j}$,若发送消息的组件C_i运行成功,则通过该连接器可能将消息包传送到接收消息的组件C_j.下面分析组件故障模型的正确性.

性质 3. 在HSCN模型中,$\forall C_i\in C,\exists M\in R(M_0)$,有$M(C_i.p_e)=\varphi$.

证明:$\forall C_i\in C$,

(1) 若组件C_i无前向组件,则由服务组合的容错模型可知,$C_i.p_i\in t_{st}^\bullet$,

所以$M_1=M_0[t_{st}\rangle$且$M_1(C_i.p_i)=\varphi$.

因为$|WS_i|\neq\varnothing$,即$M_1(C_i.p_w)=M_0(C_i.p_w)\neq\varnothing$,

所以$C_i.t_{in}\in FT(M_1)$,即组件C_i调用了某个可用服务$WS_{i,j}$.

根据性质 1 可得:$\exists M_2\in R(M_1)$,有$M_2(WS_{i,j}.p_{ou}^O)=\varphi$.

因为$^\bullet C_i.t_e=WS_{i,j}.P_{ou}\wedge(\forall t_k\in T,{}^\bullet t_k\cap{}^\bullet t_i=\varnothing)$,

所以$C_i.t_e\in FT(M_2)$,即$\exists M_3\in R(M_2)$,有$M_3(C_i.p_e)=\varphi$.

令$M=M_3$,则命题成立,

所以当组件C_i无前向组件时命题成立.

(2) 设对于组件C_i的所有前向组件,命题均成立,下面证明对于组件C_i,命题也成立.

① 设存在组件C_k,满足$C_k>C_i$,

因为对于组件C_k,命题也成立,

所以$\exists M_1\in R(M_0)$,有$M_1(C_k.p_e)=\varphi$,即$M_1(C_k.p_o^O)=\varphi$.

因为根据性质 3 可得:$\exists M_2\in R(M_1)$,使得$M_2(C_i.p_i^I)=\varphi$,

因为$|WS_i|\neq\varnothing$,即$M_1(C_i.p_w)=M_0(C_i.p_w)\neq\varnothing$,

所以$C_i.t_{in}\in FT(M_1)$,即组件C_i调用了某个可用服务$WS_{i,j}$.

根据性质 1 可得:$\exists M_2\in R(M_1)$,有$M_2(WS_{i,j}.p_{ou}^O)=\varphi$.

因为$^\bullet C_i.t_e=WS_{i,j}.P_{ou}\wedge(\forall t_k\in T,{}^\bullet t_k\cap{}^\bullet t_i=\varnothing)$,

所以$C_i.t_e\in FT(M_2)$,即$\exists M_3\in R(M_2)$,有$M_3(C_i.p_e)=\varphi$.

令$M=M_3$,则命题成立.

② 同理可以证明,对于其他两对基本组件,即 AND-split 和 OR-split,AND-join 和 OR-join,命题均成立.

综上所述,在HSCN模型中,$\forall C_i\in C,\exists M\in R(M_0)$,有$M(C_i.p_e)=\varphi$. □

性质 3 说明,在HSCN模型中,对于任意一个组件C_i,系统都可能到达一个组件C_i运行成功的状态.

性质 4. 在HSCN模型中,$\forall x\in C\cup WS\cup CT,\exists M\in R(M_0)$,有$x.t_{in}\in FT(M)$.

证明:$\forall x\in C\cup WS\cup CT$,若$x\in C$,根据性质 3 可知,组件$x$可能被执行,即$\exists M\in R(M_0)$,有$x.t_{in}\in FT(M)$.若$x\in WS$,则不妨设与$x$对应的组件为$C_i$,因为$C_i.t_{in}$的谓词永真,即随机选择可用服务库$(p_w)$中任意可用服务作为执行服务,所以服务$x$可能被调用,即$\exists M\in R(M_0)$,有$x.t_{in}\in FT(M)$.若$x\in CT$,则根据性质 2 可知,$\exists M_1\in R(M_0)$,使得其发送组件$C_i$满足$M_1(C_i.p_e)=\varphi$.从连接器的模型可知,$\exists M\in R(M_1)$,有$x.t_{in}\in FT(M)$,命题成立.

综上所述,在HSCN模型中,$\forall x\in C\cup WS\cup CT,\exists M\in R(M_0)$,有$x.t_{in}\in FT(M)$.证毕. □

性质 4 说明,服务组合中的任意一个服务、组件或连接器都会得到调用.性质 1~性质 4 主要分析服务组合故障处理框架中基本组成部分的正确性,下面分析整个系统的正确性.

根据 HSCN 模型的运行机制,其状态空间会有一些特殊的标识.同一个服务组合故障模型Ω,若组件调用的服务不同,系统的终止标识也会不同.设 M 为Ω的一个可达标识,记Ω的所有可能终止标识为 $TS(\Omega)$.

(1) 若$FT(M)=\varnothing$,则称M为Ω的终止标识,记$TS(\Omega)=\{M|M\in R(M_0)\wedge FT(M)=\varnothing\}$.

(2) 若$M\in TS(\Omega)\wedge M(p_{en})=\varphi$,则称$M$为$\Omega$的正常终止标识,记$TS^E(\Omega)=\{M|M\in TS(\Omega)\wedge(M(p_{en})=\varphi)\}$.

(3) 若$M\in TS(\Omega)\wedge(\exists x\in C\cup CT\rightarrow M(p_{eh})=x)$,则称$M$为$\Sigma$的失败终止标识,记

$$TS^F(\Omega)=\{M|M\in TS(\Omega)\wedge(\exists x\in C\cup CT\rightarrow M(p_{eh})=x)\},$$

其中,$TS(\Omega)=TS^E(\Omega)\cup TS^F(\Omega)$.终止标识是指在该标识下模型没有变迁可以触发;正常终止标识是指服务组合运行结束的终止标识,即完成所需要的功能需求;而失败终止标识表示系统中某个组件x出现故障,经过系统协调到达的终止标识.服务组合故障模型的正确性指系统到达的任何终止标识都是可预期的.如当系统处于正常终止标识M时,整个服务组合中不存在某个组件或连接器发生故障;而在失败终止标识时,整个服务组合中存在唯一组件或连接器发生故障.

定理 1. 在HSCN模型中,$\forall M\in TS^E(\Omega),\forall x\in C\cup CT,M(x.p_{fa})=\varnothing$.

证明:用反证法.

设$\forall M\in TS^E(\Sigma),\exists x\in C\cup CT,M(x.p_{fa})\neq\varnothing$,

因为$M(x.p_{fa})\neq\varnothing$,

所以$M(x.p_{fa})=\varphi$.

因为$^\bullet(x.p_{fa})=x.t_f\wedge(x.t_f)^\bullet=\{x.p_{fa},x.p_f^O\}$,

所以$\exists M_1$,使得$M_1(x.p_{fa})=M_1(x.p_f^O)=\varphi\wedge M\in R(M_1)$.

所以故障模型中的端口节点$M_1(x.p_f)=\varphi$.

因为$\forall x\in C\cup CT,\exists!t\in T$,使得$^\bullet t=x.p_f\wedge t^\bullet=p_{cn}$,

又因为$^\bullet t_{cn}=p_{cn}\wedge p_{eh}\in t_{en}^\bullet$,

所以$\forall M_2\in R(M_1)$,有$M_2(p_{eh})=x$.

因为$M\in TS^E(\Omega)$,

所以 $ET(M)=\varnothing$,即 M 是一个终止标识.

因为$M\in R(M_1)$,

所以$M(p_{eh})=x$.

根据失败终止标识的定义可知,$M\in TS^F(\Omega)$,与假设$M\in TS^E(\Omega)$矛盾.因此,假设不成立.

所以$\forall M\in TS^E(\Omega),\forall x\in C\cup CT,M(x.p_{fa})=\varnothing$. □

定理 2. 在HSCN模型中,$\forall M\in TS^F(\Omega),\exists!x\in C\cup CT$,使得$M(x.p_{fa})=\varphi$.

证明:用反证法.

设$\forall M\in TS^F(\Omega)$,命题不成立,即在$M$下存在多个组件或连接器处于故障位置.

不妨设存在$x_i,x_j\in C\cup CT$,使得$M(x_i.p_{fa})=M(x_j.p_{fa})=\varphi$.

因为模型假设同一时刻只有唯一的组件或连接器发生故障,

所以x_i,x_j不是同时发生故障.不妨设x_i先发生故障,而x_j是在系统处理x_i故障过程中发生故障.

所以$\exists M_1\in R(S_0)$,使得$M_1(x_i.p_{fa})=\varphi$,且$\exists M_2\in R(M_1)$,使得$x_j.t_f\in FT(M_2)\wedge M\in R(M_2)$.

因为$^\bullet(x.p_{fa})=x.t_f\wedge(x.t_f)^\bullet=\{x.p_{fa}, x.p_f^O\}$,

所以$\exists S_2$,使得$M_2(x_i.p_{fa})=M_2(x_i.p_f)=\varphi\wedge M\in R(M_2)$.

因为$\forall x\in C\cup CT,\exists!t\in T$,使得$^\bullet t=x.p_f\wedge t^\bullet=p_{cn}$,

又因为$^\bullet t_{cn}=p_{cn}\wedge p_{eh}\in t_{cn}^\bullet\wedge(\forall x\in C\cup CT\rightarrow x.p_t\in t_{cn}^\bullet)$,

所以$\forall x\in C\cup CT,\forall M_3\in R(M_2)$,有$M_3(x.p_t)=\varphi$,即所有元素都可以得到故障处理.

因为$x_j.t_f$的优先级等于 1,但是与事务相关的变迁优先级均等于 0,

所以根据变迁的触发规则可知,$\forall M_k\in R(M_2)$,使得$x_j.t_f\notin FT(M_k)$,即$\forall M_k\in R(M_1),M(x_j.p_{fa})=\varnothing$.

因为$M\in R(M_1)$,

所以$M(x_i.p_{fa})=\varphi\wedge M(x_j.p_{fa})=\varnothing$,与假设矛盾.所以,假设不成立.

所以$\forall M=(M,TR)\in TS^F(\Omega),\exists!x\in C\cup CT$,使得$M(x.p_{fa})=\varphi$. □

定理 1 和定理 2 说明,服务组合故障模型到达的所有终止标识都是正确的.服务组合的故障处理映射到HSCN模型为:在标识M下,若元素x运行故障,则所有服务根据其运行位置和事务属性都能得到正确的处理.若发生故障时$WS_{i,j}$处于运行位置,则取消该服务的运行;若$WS_{i,j}$处于结束位置且其是可补偿的,则对该服务进行补偿.

定理 3. 在HSCN模型中,$\forall M\in R(M_0)$,若$\exists x\in C\cup CT$,使得$M(x.p_{fa})=\varphi$,且$M_f\in R(M)\wedge FT(M_f)=\varnothing$,则$\forall WS_{i,j}\in WS$,有:

(1) 若$d_{i,j}\in M(C_i.p_w)\cup M(C_i.p_{fw})\cup M(WS_{i,j}.p_{fa}^O)\vee(M(WS_{i,j}.p_e)=d_{i,j}\wedge cp\notin TP(WS_{i,j}))$,则个体$d_{i,j}$所处的库所不变.

(2) 若$M(WS_{i,j}.p_{ac})\cup M(WS_{i,j}.p_{wr})=d_{i,j}$,则$M_f(WS_{i,j}.p_{ca})=d_{i,j}$.

(3) 若$M(WS_{i,j}.p_e)=d_{i,j}\wedge cp\in TP(WS_{i,j})$,则$M_f(WS_{i,j}.p_{cp})=d_{i,j}$.

证明:$\forall WS_{i,j}\in WS$,下面分别证明 3 个子命题.

(1) 若$d_{i,j}\in M(C_i.p_w)\cup M(C_i.p_{fw})\cup M(WS_{i,j}.p_{fa}^O)\vee(M(WS_{i,j}.p_e)=d_{i,j}\wedge cp\notin TP(WS_{i,j}))$,即服务$WS_{i,j}$没有得到调用,运行失败且组件重新调用其他可用服务,运行失败但组件还未进行新的调用,运行成功且该服务是不可补偿的.

因为根据服务组合故障模型的运行机制可知,在上述几种情况下,$\forall M_k\in R(M_0),\forall t\in T$,变迁$t$的可行替换中不包含个体$d_{i,j}$,

所以子命题(1)成立.

(2) 若$M(WS_{i,j}.p_{ac})+M(WS_{i,j}.p_{wr})=d_{i,j}$,即服务$WS_{i,j}$得到调用,且处于运行和等待重试位置之一.

因为$^\bullet(WS_{i,j}.t_{ca})=WS_{i,j}.p_{ac}$,$^\bullet(WS_{i,j}.t_{cr})=WS_{i,j}.p_{wr}$,

又因为$WS_{i,j}.t_{ca}$和$WS_{i,j}.t_{ca}$的优先级均为 0,而$WS_{i,j}.t_{fa},WS_{i,j}.t_{wr},WS_{i,j}.t_{rt}$和$WS_{i,j}.t_e$的优先级均为 1,

所以$WS_{i,j}.t_{ca}\in FT(M)\vee WS_{i,j}.t_{ca}\in FT(M)$.

因为$(WS_{i,j}.t_{ca})^\bullet=(WS_{i,j}.t_{cr})^\bullet=WS_{i,j}.p_{ca}$,

又因为$M_f\in R(M)\wedge FT(M_f)=\varnothing$,

所以$M_f(WS_{i,j}.p_{ca})=d_{i,j}$,即子命题(2)成立.

同理可证子命题(3)成立.

综上所述,$\forall M\in R(M_0)$,若$\exists x\in C\cup CT$使得$M(x.p_{fa})=\varphi$,设$M_f\in R(M)\wedge FT(M_f)=\varnothing$,则$\forall WS_{i,j}\in WS$,命题成立. □

3.2 服务组合的故障分析算法

由服务组合的故障处理框架,故障分析的具体步骤如下:

(1) 首先采用 CTL 描述需要验证的性质.设原子命题 $ge(p,d)$表示库所 p 中有个体 d,$en(t)$表示变迁 t 有发生权,$fn(t)$表示变迁 t 是有效触发的,$M\vDash f$ 表示标识 M 满足公式 f.则性质 1~性质 4 采用 CTL 分别可以描述为

$$\forall WS_{i,j}\in WS,\forall M\in R(M_0),M\vDash\Diamond p\rightarrow\Box ge(p,WS_{i,j}),$$

$$\forall x\in C\cup WS\cup CT,\exists M\in R(M_0),M\vDash \mathrm{o}fn(x.t_{in}),$$

$$\forall C_{i,j}\in CT,\exists M\in R(M_0),M\vDash\Diamond ge(C_i.p_o,\varphi)\rightarrow\Diamond ge(C_j.p_i,\varphi);\forall C_i\in C,\exists M\in R(M_0),M\vDash\Diamond ge(C_i.p_o,\varphi).$$

同理可以描述定理 1~定理 3,除了分析本文所提到的这些性质,还可以分析模型的合理性、活性等.

(2) 根据服务组合的需求构建相应的故障模型,并计算其可达图;

(3) 利用 CTL 相关工具验证服务组合的故障模型是否满足性质;

(4) 在服务组合发生故障时采用算法及时进行故障定位;

(5) 基于故障模型的可达图分析服务组合的可靠性,如基于某个组件的失败概率等.设系统到达标识 M 的概率为 TR,若此时系统通过有效触发 $H(M)$中所有变迁到达新的标识 M',则系统到达标识 M'的概率为

$$TR'=TR\times\prod_{t_i\in H(M)}\beta_i\times\prod_{t_i(d_1,d_2,\ldots,d_n)\in H(M)}\beta_i(d_1,d_2,\ldots,d_n).$$

算法 1. 服务组合故障分析算法.

输入:需求模型Ξ和故障模型的可达图 $SP=(E,V)$.

输出:服务组合的正确性和可靠性、故障定位.

(1) *Analysis*(Ξ,SP) //分析服务组合的正确性

1: {For k=1, $k\le|E|$, k++ do

2: For i=1, $i\le|C|$, i++ do //验证组件C_i满足性质 2 或性质 4

3: If $M_k|=\mathrm{o}fn(C_i.t_{in})$ or $M_k\vDash\Diamond ge(C_i.p_o,\varphi)$ break

4: For $WS_{i,j}\in WS$ do

5: {If $M_k\vDash\Diamond p\rightarrow\Box ge(p,WS_{i,j})$ continue; //验证服务$WS_{i,j}$满足性质 1

6: Else if $M_k\vDash\mathrm{o}fn(WS_{i,j}.t_{in})$ //验证服务$WS_{i,j}$满足性质 2

7: Else return false;}

8: If $\exists C_{i,m}\in CT$ do

9: {If $M_k\vDash\Diamond ge(C_i.p_o,\varphi)\rightarrow\Diamond ge(C_m.p_i,\varphi)$ continue;

10: Else if $M_k\vDash\mathrm{o}fn(C_{i,m}.t_{in})$ //验证连接器$C_{i,m}$满足性质 2

11: Else return false;}}

(2) *Compute_AV*((*SP*) //计算服务组合的可靠性 TR

12: {$M_{now}=M_0$; $TR(M_0)=1$; TR=0

13: While ($FT(M_{now})\neq\varnothing$) do

14: {If $M_{now}[H\rangle M'$ do

15: {$TR(M_{now})=TR(M_{now})\times TR(H)$;

16: $M_{now}=M'$;}}

17: For i=1, $i\le|TS^E(\Omega)|$, i++ do

18: {TR+=$TR(M_i)$;

19: retrun TR;}}

(3) *Find_Fault*((*SP*) //故障定位

20: {For i=1, $i\le|TS^F(\Omega)|$, i++ do

21: For j=1, $j\le|C|$, j++ do

22: If $M_i(C_j.p_{fa})=\varphi$ return C_j;

23: Else if $\exists C_{j,m}\in CT$, $M_i(C_{j,m}.p_{fa})=\varphi$ return C_j;}

该算法基于服务组合的需求模型和故障模型可达图分析故障模型的正确性,验证系统的需求和特性;同时,当系统发生故障时,对故障进行定位.函数 *Analysis*(*Ξ*,*SP*)主要用于分析故障模型的性质,函数 *Compute_AV*(*SP*)是计算服务组合的可靠性,而函数 *Find_Fault*((*SP*)是对服务组合发生的故障进行定位.

4 实例仿真

基于出口服务的需求,按照服务组合故障模型的构造步骤,可以得到例 1 中出口服务的最上层故障模型*Ω*.具体模型如图 7 所示,该模型主要描述了出口服务的组合流程.

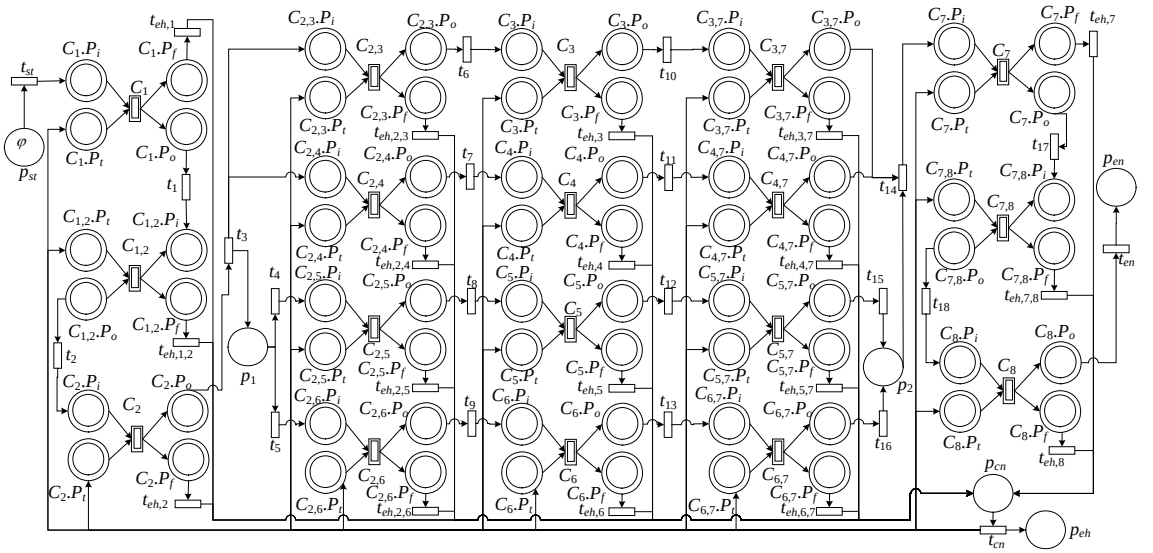


Fig.7 Fault model of an international trade export service

图 7 国际贸易出口服务的故障模型

基于算法 1,我们利用实验平台Windows XP,以C#语言实现了一个系统原型,用于计算模型*Ω*的状态空间.可得该出口服务是可能发生故障,而引起故障的元素为组件C_2或各个连接器.如可能由于C_2故障导致服务组合失败的概率为 0.0032%,而可能由于连接器$C_{1,2}$发送消息失败而使出口服务失败的概率为 0.0125%,而整个出口服务组合失败的概率为 0.1031%.且一旦某个元素发生故障,系统就可以马上抛出故障并进行即时处理.

为了更深入地阐述所提出服务组合故障处理框架的意义和作用,下面通过具体的实验来分析模型的性能.首先,随机生成 45 200 个服务构成出口服务的服务资源,每个服务至少包含服务名称、可靠性、事务属性等基本信息.基于生成的服务资源,对出口服务做两个仿真实验.

对于可重复服务而言,重试相当于调用一个新的服务,该服务与旧的服务具有相同的 QoS 属性.但是对于组件而言,若调用一个可重复服务,则该组件的成功概率为 1,且每次重试过程,系统只要执行两个操作,而重新调用一个服务,系统则需要执行 4 个操作.但由于服务资源中未必存在可重复的服务,即使存在,该服务的可靠性也不一定就比较高,所以有必要分析事务属性和可靠性对整个服务组合的影响,进而根据需要生成组件所调用的服务资源.

实验 1 的目的是分析服务事务属性对服务组合可靠性和故障模型状态空间的影响,具体实验步骤如下:

(1) 取 12 000 个服务分为 5 个实验数据,每个实验分别有 10 个出口服务,且出口服务中每个组件分别有 10,20,30,40,50 个可用服务.取服务资源中成功概率最高的前 10 个服务作为可用服务,依次对每个出口服务进行步骤(2)和步骤(3).

(2) 设每个服务都是可重复的,且组件是随机调用某个服务执行,计算每个服务重复执行 10 次以后服务组合的成功概率TR_1.

(3) 设每个服务都是不可重复的,且组件依次调用这些服务执行,计算服务组合的成功概率TR_2,并计算两组实验结果的差值TR_2-TR_1.

实验 1 的结果如图 8 所示.分析实验结果可知:当服务资源比较贫乏时,差值基本上处于[0.2%,3%];而当服务资源比较丰富时,差异不明显.如当每个组件有 40 或 50 个服务资源时,90%出口服务的差值小于 0.000001%.分析其原因为:当服务资源比较贫乏时,可用服务的平均成功概率较低(基本上都在 50%左右),进而导致差异较大;相反,当服务资源比较丰富时,可用服务的平均成功概率较高(基本上在 85%以上).同时,进一步仿真发现,如果可重复服务的成功概率在 85%以上,则得到的差值能小于 0.000001%.因此,可以根据组件C_i的服务资源特征制定调用规则,在确保服务组合可靠性的基础上,尽可能减少故障模型的状态空间:首先,选择组件C_i服务资源中成功概率最高的 10 个服务构成可用服务集WS_i(可以根据实际情况适当调整可用服务集的大小);若服务$WS_{i,j}$满足$r\in TP_{i,j}\wedge SP_{i,j}=\max\{SP_{i,k},r\in TP_{i,k}\}\in TP(WS_{i,j})\wedge SP_{i,j}\geq 85\%$,则直接调用该服务;否则,按照可用服务的成功概率,从高到低依次调用.可以将这个规则作为故障模型中变迁$C_i.t_{in}$的谓词.

实验 2 的目的是分析可用服务资源的大小对服务组合可靠性的影响,设服务组合因为任意组件C_i发生故障而运行失败的概率要求小于 0.0001%,具体实验步骤如下:

(1) 从服务资源中取 44 000 个服务分为 10 个实验数据,每个实验分别有 10 个出口服务,每个出口服务中组件分别有 10 个~100 个可用服务(以 10 为等级),依次对每个出口服务进行步骤(2).

(2) 为每个组件调用服务资源中成功概率最高的可重复服务.

(3) 假定任意组件C_i发生故障而运行失败的概率都小于 0.0001%,计算各个组件所对应服务至少需要执行的次数,并进一步计算每个出口服务对应执行次数的平均值.

实验 2 的结果如图 9 所示.分析实验结果可知:随着组件的可用服务资源的增加(≥50)时,服务的执行次数能够得到有效的控制(基本上在 2~4 之间),从而减少故障模型的状态空间.但是当服务资源比较丰富时,则执行次数的差异不大.如可用服务为 90 和 100 时,执行次数均处于 2.5~3.5 之间;而在可用服务集为 10 的情况下,40%的出口服务平均执行次数大于 10.分析其原因可知,由于服务资源是随机产生的,所以当服务资源比较少时,容易出现某个组件存在可重复服务的成功概率较低.因此,本文所提出的方法适合服务资源较丰富或成功概率较高的服务.

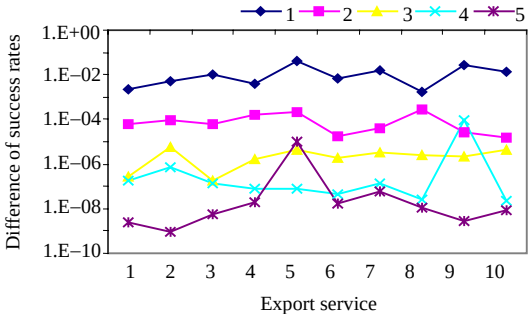


Fig.8 Simulation result of Experiment 1

图 8 实验 1 的仿真结果

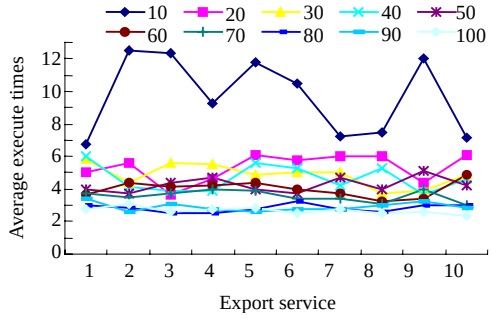


Fig.9 Simulation result of Experiment 2

图 9 实验 2 的仿真结果

限于篇幅,本文只对简化的国际贸易出口服务进行了模拟,但是它足以说明服务组合故障诊断和处理方法

的正确性.由实例模拟过程可知,应用本文所提出方法可以达到如下效果:(1) 正确刻画服务组合中故障发现、定位和处理等问题,所构造的模型可以确保服务组合的故障处理正确运行.如当组件C_2发生故障时,组件C_1所调用的服务都可以得到处理,而其余组件都不会执行.(2) 组合过程的灵活性,在组合过程中可以根据实际需求对组件进行增减.如在出口运输方式预订业务中增加铁路运输C_9时,系统中的所有组件及其子页面都不用修改,只要在上层页面中增加一个C_9的替代节点及其对应页面即可.

5 结束语

服务组合是面向服务计算中的一个重要研究内容,而故障处理是保证服务组合可靠性的关键问题.本文针对现有服务组合技术的研究成果较少考虑出现故障的问题,提出了故障发现、定位及其处理方法.该方法根据服务组合的不同故障情况,包括可用服务运行失败、组件运行故障和网络发生故障,给出相应的故障处理方法.利用 Petri 网对不同的故障情况进行建模,并依据组件的关系生成服务组合的故障处理模型.在此基础上,应用 Petri 网相关的理论和 CTL 分析了故障处理的正确性.最后,结合具体的实例对所提出的方法进行模拟,实验结果表明了该方法的可行性和有效性.与现有服务组合方法相比,本文所提出方法的优势在于:

(1) 从多方面考虑服务组合出现故障的情况,并将故障处理通过 Petri 网来直观地表示,从而将故障的处理转化为对所得到模型的分析,有效地解决了难以对故障进行处理的问题.

(2) 使用 CTL 对系统实现算法进行描述,CTL 的相关理论可以准确地刻画服务故障处理的过程,且可以清晰地表达服务组合的运行机制.

(3) 提高服务体系结构的错误处理能力,使服务组合能够自适应地修正运行错误.

(4) 与其他组合方法相比,服务组合成功率以及服务组合的可靠性都有明显的提高.

今后,我们将继续研究服务组合的故障处理问题,研究如何提升和确保服务组合的可靠性,并基于 Web 服务的网络应用,考虑服务组合流程中的 QoS 问题.另外,本文对模型的实现工具也没有涉及.我们将在这些方面作进一步的探索.

致谢 感谢匿名评阅人对本文的工作提出的宝贵意见和建议.

References:

[1] Li Q, Liu A, Liu H, Lin B, Gu N. Web services provision: Solutions, challenges and opportunities. In: Kim W, Choi HJ, Won D, eds. Proc. of the 3rd Int'l Conf. on Ubiquitous Information Management and Communication. New York: ACM Press, 2009. 70−87.

[2] Papazoglou MP, Heuvel WJ. Service oriented architectures: Approaches, technologies and research issues. Int'l Journal on Very Large Data Bases, 2007,16(3):389−415.

[3] Pleisch S, Schiper A. Approaches to fault-tolerant and transactional mobile agent execution—An algorithmic view. ACM Computing Surveys, 2004,36(3):219−262.

[4] Chan KSM, Bishop J, Steyn J, Baresi L, Guinea S. A fault taxonomy for Web service composition. In: Nitto E, Ripeanu M, eds. Proc. of the Int'l Workshops on Service-Oriented Computing 2007. LNCS 4907, Heidelberg: Springer-Verlag, 2009. 363−375.

[5] Liang D, Chen LF, Chen C, Lin F. Fault tolerant Web service. In: Chiang M, ed. Proc. of the 10th Asia-Pacific Software Engineering Conf. Los Alamitos: IEEE Computer Society Press, 2003. 310−319.

[6] Dobson G. Using WS-BPEL to implement software fault tolerance for Web services. In: Crnkovic I, Grünbacher P, Biffl S, Merz P, Müller P, eds. Proc. of the 32nd EUROMICRO Conf. on Software Engineering and Advanced Applications. Los Alamitos: IEEE Computer Society Press, 2006. 126−133.

[7] Ardissono L, Furnari R, Goy A, Petrone G, Segnan M. Fault tolerant Web service orchestration by means of diagnosis. In: Volker G, Flávio O, eds. Proc. of the Software Architecture. LNCS 4344, Heidelberg: Springer-Verlag, 2006. 2−16.

[8] Vijay D, Simon M, Luc M, Roure DD, Luck M. Transparent fault tolerance for Web services based architectures. In: Monien B, FeldMann R, eds. Proc. of the 8th Int'l Euro-Par Conf. on Parallel Processing. LNCS 2400, Heidelberg: Springer-Verlag, 2002. 107−201.

[9] Zhang J, Qiu RG. Fault injection-based test case generation for SOA-oriented software. In: Robin GQ, ed. Proc. of the IEEE Int'l Conf. on Service Operations and Logistics and Informatics. Los Alamitos: IEEE Computer Society Press, 2006. 1070−1078.

[10] Arlat J, Costes A, Crouzet Y, Laprie JC, Powell D. Fault injection and dependability evaluation of fault-tolerant systems. IEEE Trans. on Computers, 1993,42(8):913−923.

[11] Looker N, Xu J. Assessing the dependability of SOAP RPC-based Web services by fault injection. In: Proc. of the 9th IEEE Int'l Workshop on Object-Oriented Real-Time Dependable Systems. Los Alamitos: IEEE Computer Society Press, 2003. 163−170.

[12] Fan XQ, Jiang CJ, Wang JL. Random-QoS-Aware reliable Web service composition. Journal of Software, 2009,20(3):546−556 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3339.htm

[13] Fan GS, Liu DM, Chen LQ, Yu HQ. A coordination strategy for reliable service composition and its analysis. Chinese Journal of Computers, 2008,31(8):1445−1457 (in Chinese with English abstract).

[14] Bhiri S, Perrin O, Godart C. Ensuring required failure atomicity of composite Web services. In: Proc. of the 14th Int'l Conf. on World Wide Web. New York: ACM Press, 2005. 138−147.

[15] Schuldt H, Alonso G, Schek H. Concurrency control and recovery in transactional process management. In: Proc. of the 18th ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems. New York: ACM Press, 1999. 316−326.

[16] Yuan CY. The Theory and Application of Petri Nets. Beijing: Publishing House of Electronics Industry, 2005 (in Chinese).

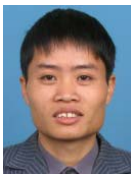
[17] He XD, Deng Y. A framework for developing and analyzing software architecture specifications in SAM. The Computer Journal, 2002,45(1):111−128.

附中文参考文献:

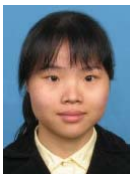
[12] 范小芹,蒋昌俊,王俊丽,庞善臣.随机 QoS 感知的可靠 Web 服务组合.软件学报,2009,20(3):546−556. http://www.jos.org.cn/1000-9825/3339.htm

[13] 范贵生,刘冬梅,陈丽琼,虞慧群.可靠服务组合的协调策略与分析.计算机学报,2008,31(8):1445−1457.

[16] 袁崇义.Petri 网原理与应用.北京:电子工业出版社,2005.



范贵生(1980－),男,福建莆田人,博士,助理研究员,CCF 学生会员,主要研究领域为软件工程,面向服务计算,形式化方法.



陈丽琼(1982－),女,博士,讲师,主要研究领域为分布式计算,嵌入式系统,形式化方法.

虞慧群(1967－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为软件工程,可信计算,形式化方法.

刘冬梅(1970－),女,博士,高级工程师,主要研究领域为软件工程,面向服务计算.